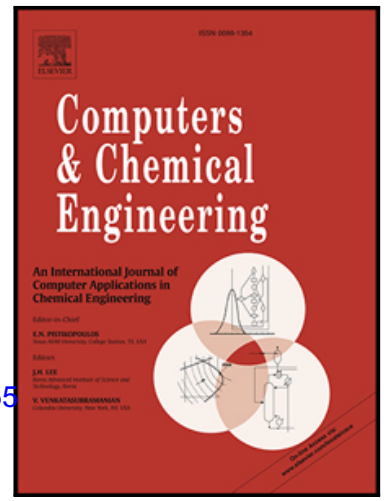

Simulation-Based Optimization of Distillation Processes Using an Extended Cutting Plane Algorithm

Juan Javaloyes-Antón , Jan Kronqvist , José A. Caballero

PII: S0098-1354(21)00432-4
DOI: https://doi.org/10.1016/j.compchemeng.2021.107655
Reference: CACE 107655

To appear in: *Computers and Chemical Engineering*

Received date: 7 September 2021
Revised date: 23 December 2021
Accepted date: 29 December 2021

Please cite this article as: Juan Javaloyes-Antón , Jan Kronqvist , José A. Caballero , Simulation-Based Optimization of Distillation Processes Using an Extended Cutting Plane Algorithm, *Computers and Chemical Engineering* (2021), doi: https://doi.org/10.1016/j.compchemeng.2021.107655

Simulation-Based Optimization of Distillation Processes Using an Extended Cutting Plane Algorithm

Juan Javaloyes-Antón[a*], Jan Kronqvist[b,c], José A. Caballero[a]

[a]*Institute of Chemical Process Engineering. University of Alicante, Alicante, Spain.*

[b]*Optimization and Systems Theory, Department of Mathematics. KTH Royal Institute of Technology, Stockholm, Sweden.*

[c]*Department of Computing. Imperial College London, London, United Kingdom.*

javaloyes.anton@ua.es, *jankr@kth.se*

Highlights

- A novel framework for optimizing nonconvex simulation-based MINLP problems
- The method combines the ECP algorithm with techniques tailored for black-box problems
- No-good cuts and backtracking techniques are implemented for simulation failures
- Heuristic strategies designed to improve practical performance for nonconvex problems
- Expansion of the search space and restarting techniques for nonconvex problems

Abstract

The use of commercial flowsheeting programs enables straight-forward use of rigorous, but user hidden, mathematical formulations of chemical processes. The optimization of such black-box models is a challenging task due to nonconvexity, absence of accurate derivatives, and simulation convergence failures which can prevent classical optimization procedures from continuing the search. Here, we present an optimization framework based on the extended cutting plane algorithm with additional heuristic techniques and strategies designed to improve its practical performance for solving nonconvex simulation-based MINLP problems. The new algorithmic features include two approaches for dealing with nonconvexities; the first technique expands the search space to restore feasibility of the MILP subproblems, and the second is a restarting technique to avoid premature termination to non-optimal solutions. We also propose two approaches for handle simulation failures, based on no-good cuts and backtracking. The proposed optimization framework is successfully applied to four case studies dealing with the economic optimization of distillation processes.

Keywords: black-box optimization; process simulation; MINLP; extended cutting plane; nonconvex heuristics; distillation processes.

1 Introduction

Chemical process simulators or flowsheeting programs have reached a level of maturity that they have become an everyday tool for chemical engineers in industry and academic institutions. These software packages include thermodynamic libraries and tailored numerical models for a variety of unit operations that lead to accurate predictions for a variety of simulated processes. While chemical process simulators are widely used in analysis, sizing, or cost estimation, they have traditionally only been used in the final stage of process synthesis and design as a validation tool. In recent years, researchers have studied the advantages of using chemical process simulators also in the first stage of chemical process synthesis and they have proposed different approaches to deal with the closed modular structure of the commercial process simulators.

Due to the demand of synthesis tools, process simulation companies have implemented some optimization routines in their products. For example, Aspen Plus incorporates an optimization tool for solving nonlinear programming (NLP) problems, including two optimization algorithms. The Complex algorithm, which is a feasible path black-box pattern search that does not require derivative information, and a sequential quadratic programming (SQP) method utilizing gradient information. The Complex algorithm only handles inequality constraints and cannot be used in problems with recycle loops, while the SQP method can handle both inequality and equality constraints in problems with recycle loops (AspenTech, 2003). Aspen HYSYS has several optimization solvers for NLP problems, including derivative free algorithms and an SQP solver among others (AspenTech, 2011). However, the optimization capabilities of process simulators are still limited for optimization problems involving integer variables.

Integer variables are especially important in the first stage of chemical process synthesis since they enable distinct decisions, such as on/off, number of units, and out of service/in service, to be included in optimization problems. Such optimization problems with both nonlinear functions and integer variables are commonly referred to as mixed-integer nonlinear programming (MINLP) problems. There are several MINLP solvers available in general purpose optimization tools and modelling frameworks (Kronqvist et al., 2019), such as AIMMS, GAMS, and Pyomo. However, in process simulators the use and availability of MINLP solvers is still limited. Aspen Plus does not include MINLP solvers, but Aspen HYSYS includes two MINLP solvers. The first is a stochastic approach based on the simulated annealing algorithm, and the second is based on a branch and bound (BB) algorithm. Still, these MINLP solvers offers limited flexibility. For example, to the authors best knowledge it is not possible to model the optimization of the number of trays and feed location of a distillation column.

In this paper, we study how the extended cutting plane (ECP) algorithm (Westerlund and Pettersson, 1995) can be applied to synthesis problems based on process simulators. We develop techniques to address the challenges of using the algorithm on problems where some variables or function values are provided by a process simulator, and techniques to improve the algorithms performance for these nonconvex MINLP problems. We focus on the synthesis and design of distillation columns, but the techniques presented within the paper are applicable to more general synthesis problems based on process simulators. There are known challenges related to the calculation of the derivatives with process simulators but is has been shown that derivative-based solvers can still efficiently be employed. A brief literature review of different approaches for solving optimization problems with process simulators embedded is given in Section 2. As far as the authors are aware, previous work has mainly focused on using the outer approximation (OA) (Duran and Grossmann, 1986) algorithm to solve the MINLP optimization problem. However,

for problems including black-box functions the ECP algorithm has two main advantages over algorithms such as OA and BB. First, the ECP algorithm does not need to evaluate the functions at fractional solutions. This gives an advantage since it may not be possible to evaluate the process simulator at non-integer solutions, e.g., using 8.3 trays in a distillation column. Secondly, it has been shown that the ECP algorithm can require significantly fewer function evaluations for problems where derivatives need to be calculated numerically by finite differences (Emet and Westerlund, 2004). For the problems considered in this paper, some function evaluations involve running a scenario with the process simulator, making it a timewise expensive operation. In general, the ECP algorithm has a slower convergence rate than method such as OA, Q-OA (Kronqvist et al., 2020), and center-cut (Kronqvist et al., 2017) , but the NLP subproblems utilized by these methods can result in significantly more function evaluations. Therefore, the ECP algorithm can be well suited for these optimization problems.

MINLP problems are in general difficult to solve, and additionally the simulation-based optimization problems considered here involve following challenges: nonconvex functions, nonsmooth functions, inaccurate derivatives, and convergence issues of the simulator. The nonsmooth functions can easily be handled with the ECP algorithm (Eronen et al., 2015), but the non-convexity and convergence issues are major challenges. Both ECP and OA are intended for convex problems, and they can even fail to find a feasible solution to non-convex problems. Here we use some simple, but efficient, techniques to improve the performance of the ECP algorithm for non-convex problems. The techniques improve the algorithm's capability of finding feasible solutions, result in better solutions, and help to mitigate effects of inaccurate derivatives. Convergence failure of the process simulator is here a major challenge that cannot be ignored, as it interrupts the optimization procedure. For example, the simulator might be given an infeasible variable combination, or it may fail to converge due to numerical difficulties. Here, we develop a framework for dealing with simulator convergence issues to enable the optimization procedure to continue; first we try to automatically converge the simulator and secondly, we derive cutting planes through backtracking and exclude variable combinations resulting in convergence failure. Together these techniques allow us to find high-quality solutions to our case studies on optimizing distillation processes.

The remainder of this work is organized as follows. Section 2 describes the main challenges when commercial sequential modular process simulators are used as calculation engines for optimization purposes, and gives a brief overview of different approaches presented in literature. Section 3 provides a background of the MINLP extended cutting plane algorithm. Section 4 describes in detail the heuristic techniques proposed to improve the performance of the ECP algorithm for nonconvex simulation-based optimization problems. Section 5 describes the implementation of the proposed framework. Section 6 shows the application of the proposed simulation-based approach to optimize four distillation-based separation processes. In Section 7 the results obtained with the proposed approach based on the ECP algorithm are compared with results obtained with a derivative-free particle swarm optimization algorithm. Finally, some concluding remarks are discussed in Section 8.

2 Using process simulators as calculation engines within optimization problems

In the following paragraphs we outline how a modular process simulator can be used as a calculation engine for solving chemical process synthesis problems.

In sequential modular process simulators, it is possible to define a superstructure using a State Task Network (STN) representation (Yeomans and Grossmann, 1999). The approach consists of assuming that a given task can be carried out by a single piece of equipment/unit operation. The connectivity between the different tasks can be done through

mixers and splitters and can be used as logic nodes. All the mixers and splitters do not necessarily exist in the final solution, but they are used to model different process configurations. The superstructure can be drawn in the actual process simulators.

The next step consists of forming an optimization problem and develop a methodology to solve it. Generalized Disjunctive Programming (GDP) (Raman and Grossmann, 1994) offers a natural framework for modeling such optimization problem. However, GDP problems are typically also transformed into MINLP problems to solve the problem by MINLP solvers (Trespalacios and Grossmann, 2014), and here we directly model the optimization tasks as MINLP problems. We now need to deal with the optimization of a black-box MINLP problem, which give rise to a set of known problems (Cozad et al., 2014) described in the following paragraphs.

Lack of derivative information and numerical errors: Due to the black-box problem architecture, gradient information can only be obtained by numerical differentiation, i.e., derivatives are calculated by perturbing the independent variables. This significantly increases the number of evaluations of the simulation model, and consequently, increases the computation time. More importantly, some unit operations inherently include numerical noise (Caballero and Grossmann, 2008). The noise can be negligible from the simulation or design point of view, but may result in poor estimates of the derivatives. Inaccurate derivatives can cause erratic solver behavior, as ascent directions may incorrectly be considered as descent directions and the KKT (Karush-Kuhn-Tucker) conditions might not even be satisfied at an optimal solution. The effect of the numerical noise in the calculation of derivatives is illustrated with the simulation experiment described in Supplementary Material.

The experiment shows that accurate derivatives can be estimated by selecting an appropriate perturbation parameter and Equilibrium/Energy error tolerances in the simulator. Evidently, the tighter the tolerances, the better the derivatives. However, very small tolerances often increase the simulation time and can even lead to convergence failure for the simulator. For a given value of the error tolerances, the quality of the derivatives gets worse (more noise) as the perturbation parameter (or step length) decreases. On the other hand, the perturbation size must be as small as possible to minimize the error in the approximation of the derivatives. Thus, to obtain accurate derivatives, the perturbation parameter must be as small as possible and at the same time, big enough "to guard against" numerical noise (Biegler and Hughes, 1982). Moreover, the numerical noise amplitude is not necessarily the same for different dependent variables. For example, in the simulations in Supplementary Material the product component mole fractions and temperature are affected by numerical noise much less than the condenser and reboiler duties. This implies that different perturbation sizes can be motivated for different dependent variables. Nevertheless, the use of different perturbation parameters increases the number of flowsheet evaluations by a factor equal to the number of different perturbation parameters used to estimate the derivatives.

Simulator convergence failures: Perhaps the most important challenge when using a derivative-based solver, is related with flowsheet convergence errors. Such errors are critical since it prevents us from determining the values of some dependent variables and/or estimate gradients. With a gradient-based algorithm such as ECP or OA, it is not possible to directly continue the optimization procedure under such circumstances as gradient information is needed. Therefore, it is necessary to implement strategies to deal with simulator convergence failures to continue the optimization procedure. Simulator convergence failures tend to become more frequent as the flowsheet (simulation model) becomes more complex. Some flowsheet components are also known to cause more numerical errors and convergence difficulties, e.g., recycle streams can make convergence more difficult and behave as noise amplifiers (Martín, 2014).

Despite the difficulties related to numerical errors and inaccurate gradients mention earlier, several gradient-based flowsheet optimization techniques have been proposed. For example, Diwekar et al. (1992) proposed a MINLP methodology using Aspen Plus and a variant of the OA algorithm for solving different process synthesis problems, including the hydrodealkylation of toluene process (HDA). Díaz and Bandoni (1996) developed a procedure using the OA algorithm and an existing ad-hoc simulator for the optimization of a real ethylene plant. Caballero et al. (2005) proposed a methodology to optimize distillation columns using a MINLP approach and Aspen HYSYS. The optimization of distillation columns requires special attention since these unit operations involve integer variables to determine the number of trays and feed tray location and, in addition, they inherently include some numerical noise. Caballero et al. adapted the OA algorithm to avoid solving the relaxed NLP subproblem at the initial point and modified the MILP master problem to deal with the special features of distillation columns in a modular environment. Later, Brunet et al. (2012) addresses the multi-objective flowsheet optimization (with cost and environmental objectives) of ammonia-water absorption cycles using Aspen Plus and a similar MINLP strategy as used by Caballero et al. Navarro et al. (2014) developed a modelling system that integrates Aspen HYSYS with the logic-based outer approximation algorithm (Türkay and Grossmann, 1996) for the structural flowsheet optimization, and they applied it to the synthesis of a methanol plant. Caballero et.al (2015) also solved problems involving sequences of complex distillation columns by considering that a distillation column is generated by selecting a rectifying and stripping section among a set of candidates that differ in the number of trays. The model is formulated as a Generalized Disjunctive Programming (GDP) problem (Balas, 1979; Raman and Grossmann, 1994) and solved using the logic based outer approximation algorithm and Aspen HYSYS.

Another well-established approach is to replace the critical unit operations, i.e. numerically noise and/or time-consuming unit operations, by surrogate models. Then, the resulting hybrid model involving explicit equations and unit operations at the level of the process simulator is optimized using NLP or MINLP solvers. Artificial neural networks and kriging interpolation techniques are two of the most popular methods to build surrogate models (Henao and Maravelias, 2011). Examples of this approach are the works of Caballero and Grossmann (2008) and Quirante et al. (2018). A detailed review of surrogate-based optimization is given by (Bhosekar and Ierapetritou, 2018).

Several approaches using derivative free optimization (DFO) algorithms coupled with sequential modular process simulators have also been presented. Remark that, DFO algorithms are particularly well suited for black-box optimization problems, given that they only require the objective function value to be evaluated and it is easy to handle simulation convergence failures. However, we cannot expect the same performance of the derivative-base methods (Biegler, 2010). General DFO algorithms can require a large number of function evaluations, exhibit poor performance in highly constrained systems, and typically they do not guarantee optimality in a finite number of iterations. Most of these algorithms handle the constraints by adding a penalty to the objective function to account for the violation of the constraint set (Mezura-Montes and Coello Coello, 2011). However, different DFO algorithms have been tested for optimization problems with process simulators. For instance, Gross and Roosen, 1998; Ibrahim et al., 2017; Leboreiro and Acevedo, 2004 and Vazquez–Castillo et al., 2009 have used genetic algorithms for the optimization of different chemical processes, including distillation columns. Dantus and High (1999) developed a methodology using a stochastic annealing algorithm, Javaloyes et al., (2013) used the particle swarm optimization algorithm, and Aspelund et al., (2010) developed a simulation-based optimization approach based on a tabu search and the Nelder-Mead Downhill Simplex method. Bayesian optimization is another popular approach to optimize expensive black box functions (Brochu et al., 2010). A black-box optimization framework that combines gradient boosted trees and uncertainty measures to efficiently optimize expensive black box functions was recently presented in (Thebelt et al.,

2020). For more details on DFO optimization, we refer to the review papers by (Boukouvala et al., 2016; Rios and Sahinidis, 2012)

3 Basics of the Extended Cutting Plane

The MINLP problems we consider can all be formulated as

$$\begin{aligned} &\min f(\mathbf{x}) \\ &\text{s.t. } g_j(\mathbf{x}) \leq 0 \quad \forall j = 1,\ldots,m, \\ &\quad \mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub}, \\ &\quad \mathbf{x} \in \mathbf{R}^N,\ x_i \in \mathbf{Z}\ \forall i \in I_{\mathbf{Z}}, \end{aligned} \tag{1}$$

where $\mathbf{x}_{lb}$ and $\mathbf{x}_{ub}$ are lower and upper bounds on the variables, and $I_{¢}$ is a set containing the indices of all integer variables . Here, some of the constraint functions g_j and/or the objective function f will be black box functions, where the functions values are obtained by running the process simulator. Note that, an equality constraint is easily represented by two inequality constraints. In general, MINLP problem (1) is regarded as convex if both the objective and all constraints are given by convex functions. The MINLP problems dealt within this paper are nonconvex, but we begin by presenting the basic ECP algorithm within a convex setting to better explain the difficulties of applying the algorithm to nonconvex problems.

The ECP algorithm was originally intended for solving convex MINLP problems (Westerlund and Pettersson, 1995), and with minor modifications it has been proven that the algorithm finds the optimal solution to nonsmooth problems with pseudo convex functions (Eronen et al., 2015). The main idea behind the ECP method is to iteratively construct a linear approximation of the MINLP problem (1). At iteration k, a linear approximation of the MINLP problem is given by

$$\begin{aligned} &\left(\mathbf{x}^k, m^k\right) \in \arg\min m \\ &\text{s.t. } f(\mathbf{x}^i) + \nabla f(\mathbf{x}^i)^T \left(\mathbf{x} - \mathbf{x}^i\right) \leq m \quad \forall i = 1,\ldots k-1, \\ &\quad g_j(\mathbf{x}^i) + \nabla g_j(\mathbf{x}^i)^T \left(\mathbf{x} - \mathbf{x}^i\right) \leq 0 \quad \forall j \in A_i, \forall i = 1,\ldots k-1, \\ &\quad \mathbf{x}, m \in \mathbf{R}^{N+1},\ x_i \in \mathbf{Z}\ \forall i \in I_{\mathbf{Z}}, \end{aligned} \tag{MILP-k}$$

where $\mathbf{x}^i$ are the trial solutions obtained in previous iterations and A_i contains the indexes of the violated constraints at iteration i. Problem (MILP-k) is sometimes referred to as the *MILP-master problem*, and we use this terminology throughout the paper. New trial solutions are obtained by repeatedly solving the MILP-master problem and the linear approximation is improved by accumulating linearizations of the nonlinear functions, or so-called cutting planes. We use a strategy of always adding a cutting plane for the objective function and for all violated constraints, which tends to reduce the number of iterations needed to solve the problem. The problem is usually considered as solved if the tolerance criterions

$$\begin{aligned} &f(\mathbf{x}^k) - m^k \leq \varepsilon, \\ &g_j(\mathbf{x}^k) \leq \varepsilon \quad \forall j = 1,\ldots,m, \end{aligned} \tag{2,3}$$

are satisfied. The tolerance ε is chosen as a small value, which is considered as an acceptable constraint violation. In case the criterions are not met, the procedure continues, and the next MILP-master problem will then be more accurate

due to the accumulated cutting planes. For convex problems the linearizations will underestimate the nonlinear functions, which guarantee that no feasible solutions are excluded from the search space by the cutting planes, and after a finite number of iterations the procedure is guaranteed to obtain an optimal solution within the given tolerance (Westerlund and Pettersson, 1995).

As previously mentioned, the problems considered here are not convex and includes further difficulties, such as noisy gradients and convergence issues of the simulator. Due to the nonconvexity, the linearizations in the MILP-master problem will not underestimate the functions and feasible solutions will be excluded from the search space. Therefore, the ECP algorithm will not in general converge to a local optimal solution for nonconvex problems and might even fail to find any feasible solutions. The same convergence issues will also appear with the OA algorithm, and the noisy gradients can also cause issues when solving in the NLP subproblems. For example, due to noise in the gradients the KKT system might not be satisfied even for the optimal set of primal and dual variables.

In the next section, we will present some heuristic techniques to improve the practical performance of the ECP algorithm for non-convex MINLP problems where some functions are evaluated by running a process simulator.

4 Using the extended cutting plane method for non-convex black box problems

We have already mentioned several challenges in directly applying a linearization-based algorithm, such as ECP or OA, to nonconvex MINLP problems with black-box functions. For these problems, we will not be able to guarantee global optimality and our main goal is to obtain a good feasible solution with relatively few evaluations of the black-box functions. In the next paragraphs, we will cover some of the difficulties faced when applying a linearization-based algorithm to these nonconvex problems and present some techniques to mitigate these issues.

The main difficulty in applying a linearization-based method, such as ECP, to a nonconvex problem is that linearizations do not necessarily underestimate the real functions. A cutting plane of a nonlinear constraint can, therefore, exclude some, or even all, feasible solutions from the search space. In such a situation, there are might not be any solutions satisfying the MILP-master problem resulting in an empty search space. Noisy gradients can also amplify this issue, by causing linearization to have a noncorrect slopes. Ending up with an empty search space before finding a feasible solution is not only a theoretical possibility, but a difficulty we immediately encountered in the case studies. It was, therefore, a necessity to have an approach for dealing with situations when the MILP-master problem is infeasible.

If the MILP-master problem becomes infeasible, it is not possible to directly continue with the ECP algorithms as the search space is empty. To continue the search procedure, we use the approach presented by (Javaloyes-Antón et al., 2018) which is also used as a nonconvex heuristic in the SHOT solver (Lundell and Kronqvist, 2020). If the MILP-master problem becomes infeasible in iteration k, we solve a feasibility problem defined as

$$
\begin{aligned}
&\min \sum_{j} \sum_{i} w_i r_{j,i} \\
&\text{s.t. } f(\mathbf{x}^i) + \tilde{\nabla} f(\mathbf{x}^i)^T \left(\mathbf{x} - \mathbf{x}^i\right) \leq m \quad \forall i = 1, \dots k-1, \\
&\quad g_j(\mathbf{x}^i) + \tilde{\nabla} g_j(\mathbf{x}^i)^T \left(\mathbf{x} - \mathbf{x}^i\right) - r_{j,i} \leq 0 \quad \forall j \in A_i, \forall i = 1, \dots k-1, \\
&\quad \mathbf{x}, m \in \mathbf{R}^{N+1}, r_{j,i} \in \mathbf{R}_+, x_i \in \mathbf{Z}. \quad \forall i \in I_{\mathbf{Z}},
\end{aligned}
\tag{MILP-F}
$$

where $r_{j,i}$ is a new "residual" variable allowing violations of the original cutting planes. By solving problem (MILP-F) we can determine the smallest relaxation of the cutting planes needed to restore feasibility of the MILP-master

problem. The weights w_i in the objective are intended to reduce the risk of cycling, and we use a simple strategy of selecting the weights as

$$w_i = i^2, \tag{4}$$

where i is the iteration index. This strategy will give a larger cost of relaxing the cutting planes in the last iterations, and favor larger changes to the cutting planes obtained in early iterations. This strategy is intended to avoid situations where a cutting plane is first added in one iteration and effectively removed in the following iteration.

By solving problem (MILP-F) we obtain the minimum residuals $r^*_{j,i}$, and we modify the cutting planes in the MILP-master problem according to

$$g_j(\mathbf{x}^i) + \tilde{\mathrm{N}} g_j(\mathbf{x}^i)^T \left(\mathbf{x} - \mathbf{x}^i\right) \pounds f \times r^*_{j,i} \quad \forall j \hat{\mathrm{I}} A_i, \forall i = 1,\mathrm{K}\, k. \tag{5}$$

The cutting planes are further relaxed by a factor of $f \geq 1$ times the minimum residuals to further extend the search space. Numerical tests have shown that the feasibility restoration technique can greatly improve the ability to find feasible solutions to nonconvex MINLP problems (Lundell and Kronqvist, 2020). Another technique to relax equality constraints and avoid infeasible MILP-master problems was presented by Viswanathan and Grossmann (1990), which is used by the solver DICOPT. However, neither of the techniques can guarantee that a feasible solution can be found for nonconvex problems.

We must also be able to handle situations where the process simulator fails to converge. Convergence failures can be a consequence of numerical difficulties, and in Section 5 we describe some automatic techniques to attempt achieve convergence. However, the process simulator can also fail due to an infeasible configuration of the variables. Tight upper- and lower bounds help to mitigate this issue, but there might still be infeasible variable combinations in between the bounds. Such infeasible variable combinations can be regarded as hidden-constraints, as they are not known by the end-user but hidden within the process simulator. In case the current trial solution $\mathbf{x}^k$ results in simulation failure, we will not be able to evaluate the constraint and objective functions and we will not be able to directly continue the search with the ECP algorithm.

If we are not able to get the process simulator to converge for a trial solution $\mathbf{x}^k$, we will consider it as an infeasible variable assignment. We will not be able to exclude such infeasible trial solutions with a normal cutting plane, and instead we will exclude a small neighborhood around the infeasible solution by a so-called no-good cut (D'Ambrosio et al., 2010; Nannicini and Belotti, 2012)

$$\left\|\mathbf{x}^k - \mathbf{x}\right\|_1 \geq d, \tag{6}$$

where d is the radius of the excluded neighborhood. The nonconvex constraint (6) is representable in mixed-integer linear form by introducing auxiliary variables and the constraints

$$
\begin{aligned}
&\sum_{i=1}^{N} a_i \geq d, \\
&a_i \leq x_i - x_i^k + M s_i && \forall i = 1,\ldots,N, \\
&a_i \leq x_i^k - x_i + M(1 - s_i) && \forall i = 1,\ldots,N, \\
&-a_i \leq x_i - x_i^k \leq a_i && \forall i = 1,\ldots,N, \\
&a_i \in \mathbf{R}_+ && \forall i = 1,\ldots,N, \\
&s_i \in \{0,1\} && \forall i = 1,\ldots,N.
\end{aligned}
\tag{7a, 7b, 7c, 7d, 7e, 7f}
$$

In the set of constraints x_i refers to the i-th component of the variable vector $\mathbf{x}$ and similarly x_i^k is the i-th component of $\mathbf{x}^k$. A detailed derivation of the mixed-integer linear formulation of constraint (6) is given in Bernal et al. (2020). By adding the constraints (7a-f) to the MILP-master problem we can exclude a d neighborhood around the infeasible a trial solution $\mathbf{x}^k$, which enables us to continue the search with the ECP algorithm. The technique of adding constraints (7a-f) enables us to handle convergence failures of the process simulator, but the extra constraints and variables do increase the complexity of the MILP-master problems. In practice, d should be chosen small to avoid cutting of optimal solutions. The no-good cut technique can result in little progress for problems with large hidden infeasible regions, therefore, we also propose a second heuristic approach.

The second technique for dealing with convergence failure of the process simulator uses *feasibility backtracking*. The last solution that was successfully evaluated by the process simulator is stored as $\bar{\mathbf{x}}$. If the process simulator does not converge in iteration k, we obtain an alternative trial solution $\hat{\mathbf{x}}$ by backtracking according to:

$$\hat{\mathbf{x}} = \alpha \bar{\mathbf{x}} + (1 - \alpha)\mathbf{x}^k, \tag{8a}$$

$$x_i^{\mu} = round\left(x_i^{\mu}\right) \quad \forall i \in I_Z, \tag{8b}$$

where $\alpha \in [0,1)$ is a backtracking parameter. If the process simulator can successfully be evaluated at $\hat{\mathbf{x}}$, then it serves as the trial solution in the iteration and is used for generating linearizations to MILP-master problem. If the feasibility backtracking succeeds in finding a solution for which the simulator converges, then it is in general more efficient than the no-good cut technique as the resulting linearizations are more informative than the no-good cut.

Due to the nonconvexity, a linearization-based algorithm, such as ECP or OA, can converge to a locally non-optimal solution. The termination conditions will then be satisfied, even if the current solution is not even locally optimal. To avoid terminating the search at such a non-optimal solution, we propose a *restating technique*. We could perform a complete restart of the search, by simply using the best-found solution as a starting point. However, since running the simulator is time-wise expensive, we want use as much information as possible from the simulations already performed. Instead of throwing away all the cutting planes to restart the search, we propose a heuristic technique of relaxing the cutting planes and performing a warm start.

During the search with the ECP algorithm, we obtain a sequence of trial solutions $\{\mathbf{x}^i\}_{i=1}^k$ at which all the constraints and objective are evaluated. At iteration k the ECP algorithm generates the following cutting planes

$$\begin{aligned} c_{f,k}(\mathbf{x}) &= f(\mathbf{x}^k) + \tilde{\nabla} f(\mathbf{x}^k)^T \left(\mathbf{x} - \mathbf{x}^k\right) \leq m, \\ c_{g_{j,k}}(\mathbf{x}) &= g_j(\mathbf{x}^k) + \tilde{\nabla} g_j(\mathbf{x}^k)^T \left(\mathbf{x} - \mathbf{x}^k\right) \leq 0 \quad \forall j \in A_k. \end{aligned} \tag{9a, 9b}$$

If all the functions were convex, then the cutting planes would satisfy the conditions

$$\begin{aligned} c_{f,k}(\mathbf{x}^i) &\leq f(\mathbf{x}^i) \quad \forall i = 1,\ldots,k, \\ c_{g_{j,k}}(\mathbf{x}^i) &\leq g_j(\mathbf{x}^i) \quad \forall j \in A_k, \forall i = 1,\ldots,k, \end{aligned} \tag{10a, 10b}$$

which follows directly from convexity (Boyd and Vandenberghe, 2004). Since the functions are not convex, some of the cutting planes do not satisfy the conditions (10a) and (10b). When conditions (10a) and (10b) are violated, it indicates that some feasible solutions might be excluded from the search space. To expand the search space and the restart the search, we relax each cutting plane by reducing the left-hand side such that the conditions (10a) and (10b) are satisfied. This enables us to utilize some information accumulated by the cutting planes, and results in what can be considered a warm start. The relaxation of the cutting planes and expansion of the search space can enable the ECP algorithm to find better solutions, but there are no guarantees that the restarting technique will result in a better solution. The relaxation can also result in cycling. Therefore, we only perform the relaxation after the termination conditions are satisfied with the intention of restarting the search and try to find a better solution.

In this section we have described four techniques to improve the practical performance of the ECP algorithm for nonconvex problems with expensive black-box functions. To better illustrate how the techniques are combined with the ECP algorithm, we have included a flowsheet in that shows how the techniques are integrated.

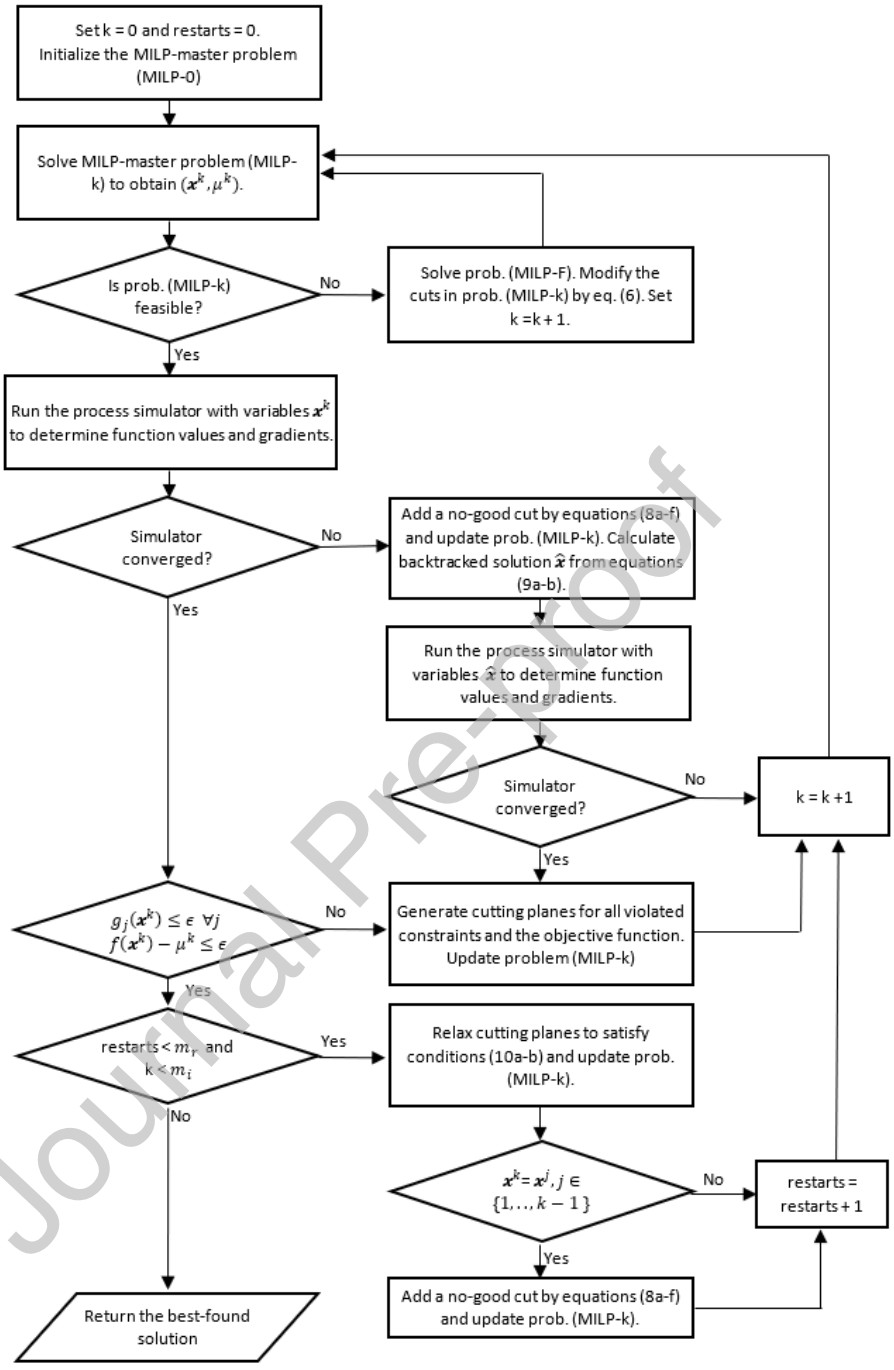

Figure 1. *Flowsheet of the ECP algorithm with nonconvex heuristics and functionality for dealing with function evaluation failure. In the flowsheet* m_r, m_i *denote the maximum restarts and maximum number of iterations.*

5 Simulation-based optimization framework

The following sections describe the use of a modular process simulator as a calculation engine to enable the implementation of the proposed simulation-based optimization tool.

5.1 MINLP formulation and process simulator integration

When a process simulator, or a general black-box, is used as a calculation engine in an MINLP optimization tool, the dimensionality of the problem for the solver is reduced since the solver only needs to consider the super-basic variables (degrees of freedom). Thus, to handle a process simulator embedded in a MINLP optimization tool, it is appropriate to partition the continuous and integer variables into dependent $\mathbf{x}^D$, and independent $\mathbf{x}^I$ variables, i.e., $\mathbf{x} = \left[\mathbf{x^D}, \mathbf{x^I}\right] \in \mathbb{R}^{n_D + n_I}$. The latter ones are the design variables, and the number of such variables is equal to the degrees of freedom of the problem. Here, the dependent variables $\mathbf{x}^D$ will be determined by the process simulator through the implicit equality constraints $\mathbf{x}^D = h^{Implicit}\left(\mathbf{x}^I\right)$. The implicit constraints, evaluated at the level of the process simulator, are the equations that describe the behavior of the process (e.g., material and energy balances, vapor-liquid equilibrium laws). Furthermore, the MINLP problem also contains some explicit constraints of the form $g_k\left(\mathbf{x^D}, \mathbf{x^I}\right) \le 0$. The explicit constraints, implemented within the MATLAB environment, can define process specifications (e.g., product specifications, temperature limits or pressure limits). The MINLP problem we consider here can , thus, be written as

$$\begin{aligned} &\min_{x^I, x^D} f\left(\mathbf{x}^D, \mathbf{x}^I\right) \\ &s.t.\ \ \mathbf{x}^D = h^{Implicit}\left(\mathbf{x}^I\right), \\ &\quad g_k\left(\mathbf{x}^D, \mathbf{x}^I\right) \le 0 \quad \forall k = 1,\ldots,q, \\ &\quad \mathbf{x}^D \in \mathbb{R}^{n_D}, \mathbf{x}^I \in \mathbb{R}^{n_I}, some\ x_i^I \in \mathbb{Z}, \end{aligned} \tag{11}$$

However, the MINLP solver will only be used to determine the independent variables and all the dependent variables will be determined by the process simulator. The MINLP solver will, therefore, only consider a reduced search space, and the number of variables considered by the MINLP solver can be reduced significantly. From the MINLP solver perspective, the optimization problem is defined as

$$\begin{aligned} &\min_{x^I} \tilde{f}\left(\mathbf{x}^I\right) \\ &s.t.\ \ \tilde{g}_k\left(\mathbf{x}^I\right) \le 0 \quad \forall k = 1,\ldots,q, \\ &\quad \mathbf{x}^I \in \mathbb{R}^{n_I}, some\ x_i^I \in \mathbb{Z}, \end{aligned}$$

where $\tilde{f}\left(\mathbf{x}^I\right) := f\left(h^{Implicit}\left(\mathbf{x}^I\right), \mathbf{x}^I\right)$ and $\tilde{g}_k\left(\mathbf{x}^I\right) := g_k\left(h^{Implicit}\left(\mathbf{x}^I\right), \mathbf{x}^I\right)$. Remark that this formulation requires sequential function evaluations. First, the implicit constraints at the level of the process simulator must be solved to obtain values of the dependent variables. Then, the objective function and explicit constraints can be evaluated.

5.2 MATLAB and Aspen HYSYS interface

Aspen HYSYS can be accessed from external programs via Automation. Automation technology makes it possible for one application to drive objects implemented in another application, or to expose objects so they can be

manipulated. Hence, Aspen HYSYS behaves as an Automation server, and its functionalities are exposed through the binary-interface standard Component Object Model (COM) to other applications, called Automation clients. We utilize MATLAB as an Automation client to access the objects exposed by the developers of Aspen HYSYS. Thus, by writing MATLAB code, it is possible to send and receive information to and from the process simulator. The exposed objects make it possible to perform nearly any action that is accomplished through the Aspen HYSYS graphical user interface, allowing us to use the process simulator as a calculation engine.

In this work we have used Aspen HYSYS, but any other process simulator that can act as a COM server, such as Aspen Plus, CHEMCAD or ProMax, could also be used as calculation engines for the proposed methodology.

5.3 Methodology Description

Figure 2 shows a scheme of the tool implemented in MATLAB for optimizing distillation processes using Aspen HYSYS and the ECP-based solver.

First, a superstructure that includes all the alternatives of interest of the process must be implemented at the level of the process simulator. Then, within MATLAB environment, the bounds on the design variables, their nature (continuous or integer), explicit constraint, constant terms and stopping criteria are defined. The next step in the algorithm is to initialize the MATLAB – Aspen HYSYS interface. We create a local Automation server with Aspen HYSYS application through the *actxserver* MATLAB function. Once this server is created, we can access to the objects, such as the material and energy streams, and unit operations in the active Aspen HYSYS flowsheet. In turn, by accessing these objects, values of the material and energy stream properties, and equipment specifications can be directly specified or modified from MATLAB.

Then, the *Process Simulator Controller Module* modifies the Aspen HYSYS superstructure by setting the new (or initial) values of the design variables. The design variables are related with the process topology (number of trays of each column section), unit operation specifications (such as reflux and boilup ratios) and/or operating conditions (column top pressure for instance). Once the superstructure is modified and the Aspen HYSYS flowsheet is executed, the simulation convergence is checked. If the flowsheet converges, the *Objective Function Module* is responsible for reading all the values of the dependent variables required to compute the objective function and constraints. If the stopping criteria are satisfied, then the algorithm is interrupted. If the stopping criteria are not satisfied, then the MINLP algorithm will determine a new trial solution. First, derivatives of the objective and constraint functions are estimated at the current solution using finite difference approximations. The derivatives and function values are then used to form new cutting planes and the master problem (MILP-k) is updated by including the new cuts. The master problem is then resolved, using the Gurobi solver, to obtain a new trial solution. As mentioned earlier, the master problem might be infeasible. In that case we apply the strategies to restore feasibility presented in Section 4, and the modified master problem is resolved to obtain a new trial solution. The procedure is then repeated and the Aspen HYSYS superstructure is modified according to the new design variables.

One of the main issue in the algorithm, and in any derivative-based approach, is when the flowsheet fails to convergence. If this happens, the *Convergence Block* tries to converge the simulation using the current values of the design variables but from a different feasible initial point. In practice, this convergence recovery strategy seems to have a low efficiency. For the case studies presented in this work, the convergence recovery strategy only managed to converge the flowsheet for approximately 10% of the convergence failures. However, if convergence can be restored, then we obtain significantly more information, i.e., function values and derivatives. Therefore, the *Convergence Block* is a valuable component even with the relatively low success rate.

If the *Convergence Block* fails to converge the flowsheet, then the *Convergence Failure Block* is activated. There are two main procedures in *Convergence Failure Block.* First, we calculate a backtracked solution by eqs. (8a-8b) which gives a solution in between the current solution and a known feasible point. If the flowsheet converges at the backtracked solution, then we use it as the current trial solution to generate cutting planes and refine the master problem. We also exclude a small neighborhood around the point where we could not converge the flowsheet by a so-called no-good cut, which is included in the master problem through equations (7a-7f). The no-good cut excludes the point where we could not converge the flowsheet from the search space and ensures that we do not try to again run the process simulator with these values for the independent design variables. Note that the no-good cuts only exclude a small neighborhood from the search space and increase the complexity of the master problem. Cuts obtained from a backtracked solution can, therefore, in practice be more efficient. But the no-good cuts are needed to guarantee that the algorithm continues, and that the same solution is not obtained in consecutive iterations by the master problem. By these simple techniques, the *Convergence Failure Block* enables us to continue the search without obtaining the objective/constraint function values or derivatives. More details on techniques are given in Section 4.

The derivatives are estimated through finite difference approximations by rerunning the process simulator and perturbing one design variable at a time. If possible, forward difference are used with a relative perturbation parameter with the perturbation defined as 1% of the current variable value. The reason for this relative perturbation parameter is to account for the different magnitudes of the variables, e.g., a fixed perturbation of 0.1 can be suitable for the distillate temperature but can be an order of magnitude too large for the component mole fraction. If one of the design variables is at the upper bound, then the derivative is estimated by a backward difference. The integer design variables are perturbed by a unit step.

For all the case studies we have used the following parameters with the ECP algorithm: maximum number of iterations $m_i = 100$, maximum number of restarts $m_r = 10$, backtracking parameter $a = 0.05$, constraint tolerance $e = 10^{-4}$, cut relaxation factor $f = 1.1$, and no-good cut radius $d = 0.1$. The parameters were chosen by trial and error and worked well for the test cases. Out of these parameters, the cut relaxation factor seems to have the largest impact. But overall, the algorithm seems robust with regards to the parameter values. All case studies were solved on a PC with an Intel i7-6700HQ at 2.6GHz processor and 16GB RAM.

Several restarts were performed for all the case studies and resulted in better solutions for two of the four case studies. As mentioned earlier, the feasibility restoration of the master problem is an essential component, and without it we would have failed to even find a feasible solution for two of the case studies.

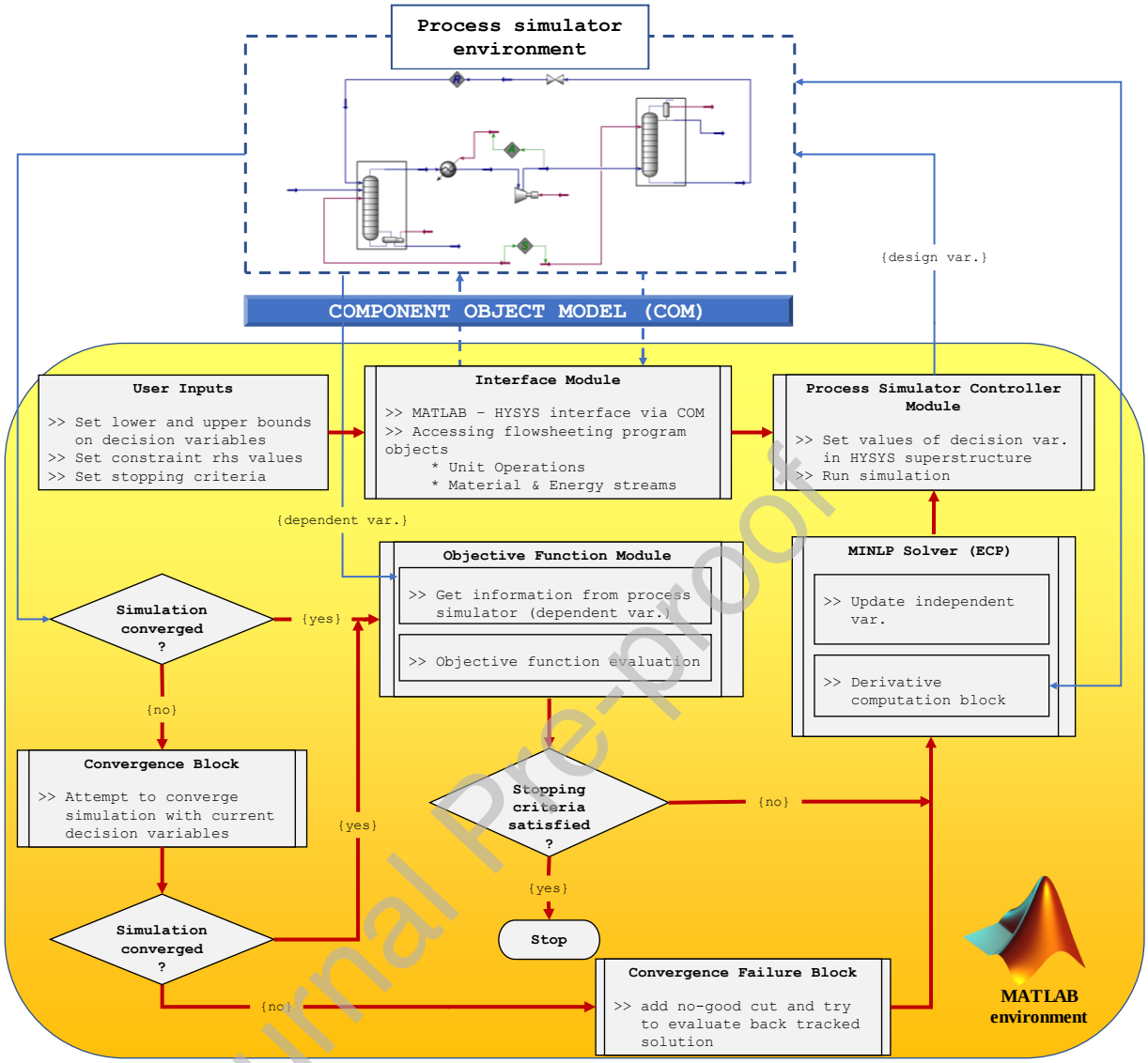

Figure 2. *Algorithm flowchart of the simulation-based optimization modeling framework.*

6 Case studies

Four examples involving different distillation-based separation processes are presented to analyze the performance and capabilities of the proposed approach. In all case studies, the objective is to minimize the total annual cost (TAC) of the distillation system. The TAC accounts for the annualized capital cost of the main equipment and the most relevant operating costs:

$$TAC = f\, C_{cap} + C_{op}, \tag{12}$$

where C_{cap} is the total cost of installed equipment ($MM), C_{op} refers to the total operating cost ($MM/year), and f is the annualization factor for the capital cost, and takes into account the fixed interest rate per year i, and the years over which the capital is to be annualized, n (Smith, 2005):

$$f = \frac{i(1+i)^n}{(1+i)^n - 1}. \tag{13}$$

For the estimation of the capital cost, only the main equipment is considered, i.e., column shells, trays, heat exchangers and compressors. The installed cost estimates are based on the correlations given by Douglas (1988). These costs are updated from the base year (1968) to 2018 using the Chemical Engineering Plant Cost Index (CEPCI_2018 = 603.1). For estimating the yearly operating cost, we only consider the contribution of the hot, cold and electricity utilities. Prices for these utilities are taken from Turton et al. (2008). Each utility cost is estimated based on 8000 operating hours per year. The equipment cost correlations and utility prices used in the TAC calculations are given in Appendix A.

6.1 Case study 1: conventional distillation column

The first case study aims at optimizing a single conventional distillation column. The problem can be stated as follow. Given a fixed feed to be separated into two products with known composition and required purity of the products (distillate and bottom). The optimization task is to determine the optimal column topology (number of trays and feed tray location), as well as the column operating conditions (reflux ratio, boilup ratio and column top pressure) that minimize the total annual cost.

To this end, we consider the superstructure of the distillation column shown in Figure 3. This superstructure is inspired in the one proposed by Yeomans and Grossmann (2000). The basic idea behind the work of Yeomans and Grossman was to consider a distillation column as a set of permanent (or fixed) trays consisting of the feed, reflux and boilup trays, and additionally, a set of conditional trays, placed between pairs of permanent trays. The latter set of trays then either behave as equilibrium stages, in which mass transfer occurs according to the vapor-liquid equilibrium laws, or as a bypass without any exchange of mass.

In our approach, we use a process simulator as a calculation engine that allows to modify the number of trays and feed tray location directly. Therefore, we do not need to model the stages as equilibrium stages or bypasses. The column topology can be simply determined from two integer variables, N_1 and N_2, which represent the number of trays in the rectifying and striping section of the column respectively, as illustrated in Figure 3.

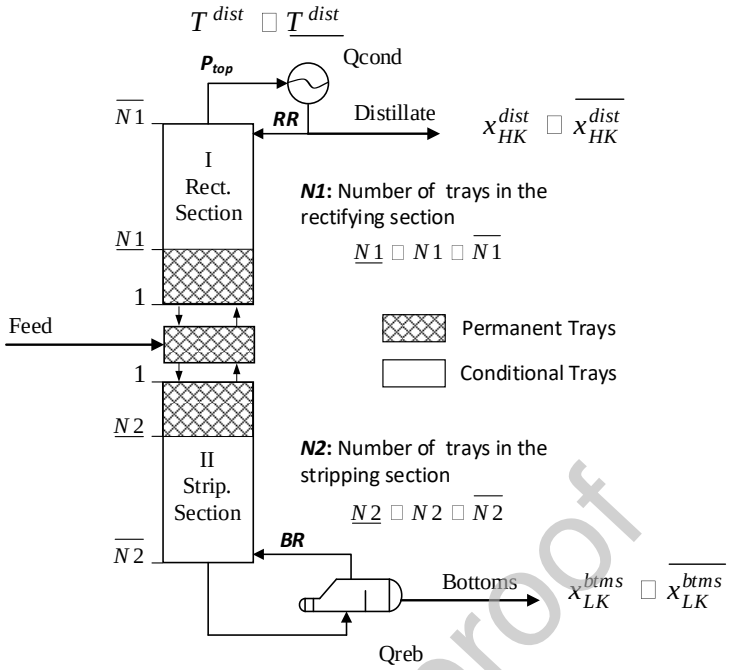

***Figure 3.** Conventional distillation column superstructure.*

Conceptually, this MINLP optimization problem can be formulated as shown in (P1)

$$\min_{\mathbf{x}^{I,cont},\mathbf{x}^{I,\text{int}},\mathbf{x}^{D}} TAC = fC_{cap}\left(N_1,N_2,A_{COL},A_{cond},A_{reb}\right)+C_{op}\left(Q_{cond},Q_{reb}\right)$$

$$s.t.\ \mathbf{x}^{D} = h^{sim}\left(\mathbf{x}^{I,cont},\mathbf{x}^{I,\text{int}}\right),$$

$$T^{dist} \geq \underline{T^{dist}},$$

$$x_{HK}^{dist} \leq \overline{x_{HK}^{dist}},$$

$$x_{LK}^{btms} \leq \overline{x_{LK}^{btms}}, \qquad \text{(P1)}$$

$$\underline{\mathbf{x}^{I,cont}} \leq \mathbf{x}^{I,cont} \leq \overline{\mathbf{x}^{I,cont}},$$

$$\underline{\mathbf{x}^{I,\text{int}}} \leq \mathbf{x}^{I,\text{int}} \leq \overline{x^{I,\text{int}}},$$

$$\mathbf{x}^{I,cont} = \{RR,BR,P_{top}\} \in \mathbb{R}^3, \quad \mathbf{x}^{I,\text{int}} = \{N_1,N_2\} \in \mathbb{Z}^2,$$

$$\mathbf{x}^{D} = \{T^{dist},x_{HK}^{dist},x_{LK}^{btms},Q_{reb},Q_{cond},A_{COL},\} \in \mathbb{R}^6.$$

The objective is to minimize the conventional distillation column TAC, which comprises the capital cost of the column shell, column trays, condenser and reboiler (all constructed in carbon steel), and the operating cost associated to the steam and cooling water required by the reboiler and condenser, respectively. The shell and tray costs are functions of the number of trays N_1 and N_2, and column area A_{COL}. The costs of the condenser and reboiler depend on their heat transfer areas A_{cond} and A_{reb} respectively. On the other hand, the operating cost is given by Q_{reb} and Q_{cond}, that correspond to the heat duties of the reboiler and condenser. Details of how TAC is computed are given in Appendix A.

The set of constraints are divided in two groups. First, the function h^{sim} refers to the set of implicit constraints which are evaluated by the process simulator. These equality constraints describes the behavior of the distillation column using rigorous tray-by-tray models. Furthermore, there are three explicit constraints implemented within MATLAB environment. These constraints are included in the model to: (i) set a minimum temperature for the distillate product, $\underline{T^{dist}}$, to allow cooling water to be used in the condenser as cold utility, (ii) the maximum molar fraction of heavy-key impurity in the distillate $\overline{x_{HK}^{dist}}$, and (iii) maximum molar fraction of light-key impurity in the bottoms $\overline{x_{LK}^{btms}}$. The independent (or design) variables of the problem are the two integer variables aforementioned (N_1 and N_2), and the continuous variables reflux ratio (RR), boilup ratio (BR), and column's top operating pressure P_{top}. We consider a standard pressure drop of 0.0069 bar (0.1 psi) per tray (Luyben, 2006).

The model implemented for the simulation-based optimization of a single conventional distillation column is tested with a multicomponent mixture of hydrocarbons ranging from C4 to C6. The objective is to separate C4's from C5's with 0.5 mol% as the maximum allowed impurity of the key components in the product streams. The molar flowrate and composition of the feed stream, thermodynamic package and other data for the example are shown in Table 1. Furthermore, we have used lower bounds to the integer variables N_1 and N_2 greater than one. Specifically, we set the bounds on $\underline{N_1}$ and $\underline{N_2}$ to 15 and 20 respectively. In this way, a distillation column with a minimum of 36 stages always exist (35 stages plus the feed tray). The reason for setting a minimum number of trays is two-folded: to avoid column configurations that cannot achieve the desired separation task, and at the same time, reduce the search space.

The best found values of the design variables and the total annual cost for the case study 1 are shown in Table 2 and Table 3 respectively. Additional information about the results can be found in the Supplementary Material.

In total the ECP algorithm performed 100 iterations, and the best solution was found after only 49 iterations. The total execution time was 461.1 seconds, of which only 10 seconds was spent on solving the MILP subproblems. The feasibility restoration technique was an essential component, and without it we would have failed to even obtain a feasible solution. Furthermore, the restarting technique allowed us to find a better solution. Compared to the other case studies the flowsheet convergence was more robust, and we only encountered a single convergence issue that was successfully handled by the Convergence Block.

Table 1. *Problem data for Case Study 1*

Feed molar flow (kmol/h)	1000
Composition (mol fraction)	
i-butane	0.17
n-butane	0.12
Cyclobutene (LK)	0.06
i-pentane (HK)	0.13
n-pentane	0.09
cyclopentane	0.07
2-methylpentane (isohexane)	0.09
n-hexane	0.12
cyclohexane	0.15
Thermodynamic fluid package	SRK
Pressure drop per tray (bar)	0.0069
Explicit constraints	
i-pentane (HK) impurity in distillate (mf)	$x_{HK}^{dist} \leq 0.005$

Cyclobutene (LK) impurity in bottoms (mf)	$x_{LK}^{btms} \leq 0.005$
distillate temperature (°C)	$T^{dist} \geq 70$
Bounds on independent variables	
N_1 *(number of trays in rectifying section)*	$15 \leq N_1 \leq 30$
N_2 *(number of trays in stripping section)*	$20 \leq N_2 \leq 35$
RR *(reflux ratio)*	$1.8 \leq RR \leq 3.5$
BR *(boilup ratio)*	$1.1 \leq BR \leq 2.5$
P_{top} *(column top pressure, bar)*	$9 \leq P_{top} \leq 13$

Table 2. *Best found design variables – Case study 1*

Design Variable	**N1**	**N2**	**RR**	**BR**	**P (bar)**
Best solution	25	34	2.5403	1.7097	9.2296

Table 3. *Objective function breakdown – Case study 1*

FIXED INVESTEMTN COST [$MM]	2.2572
Shell	1.3404
Trays	0.1071
Reboiler	0.5345
Condenser	0.2770
OPERATING COST [$MM/y]	2.8340
Low Pressure Steam [5 barg, 160 ºC]	2.7721
Cooling Water [30 to 40-45 ºC]	0.0619
TOTAL ANNUAL COST [$MM/y]	**3.2571**

6.2 Case study 2: divided wall column (modify for case with explicit constraints)

We extend the proposed methodology for the separation of a non-azeotropic three-fraction mixture using a fully thermally coupled distillation arrangement. This arrangement arises when the condenser and reboiler of the configuration for three-product separation shown in Figure 4a (prefractionator configuration) are replaced by vapor-liquid interconnections (Figure 4b). The distillation system in Figure 4b, also known as the Petlyuk configuration (Petlyuk et al., 1965), is the separation arrangement of lowest energy consumption when a three-fraction mixture has to be split into three relative pure products (Halvorsen and Skogestad, 2003a, 2003b, 2003c). In addition, this distillation system only employs two heat exchangers (one condenser and one reboiler) in contrast to the four heat exchangers needed when the separation is carried out using conventional distillation columns. However, a price to pay for this configuration is that the energy must be supplied at the highest temperature in the reboiler, and removed at the lowest temperature in the condenser, preventing, in some cases the use of more economic hot and cold utilities. A thermodynamic equivalent configuration to the Petlyuk column that uses a single column shell is shown in Figure 4c. This arrangement is known as a divided wall column (DWC) (Caballero and Grossmann, 2003).

A Petlyuk arrangement can be simulated using two columns and two heat exchangers (condenser and reboiler). However, when we simulate a thermally coupled sequence by using a process simulator it is necessary to introduce a recycle structure due to the liquid and vapor streams connecting the columns. This recycle structure give rise to two primary drawbacks a) the computation time for a single simulation increases considerably, and the most importantly b) the flowsheet becomes prone to convergence errors.

Navarro et al. (2012) show that it is possible to replace the two material streams of each thermal couple by a combination of a material and an energy stream, leading to accurate simulations with maximum errors in internal flows and energy consumption lower than 5% compared with the closed system. Specifically, in the rectifying section, the material stream is vapor at its dew point and the energy stream is equivalent to the energy removed if we include a partial condenser. And, in the stripping section, the material stream is liquid at is bubble point and the energy stream is equivalent to the energy added if we include a reboiler to provide vapor to the first column. In this way, the recycle structure is avoided resulting in a more numerically robust flowsheets (in terms of convergence).

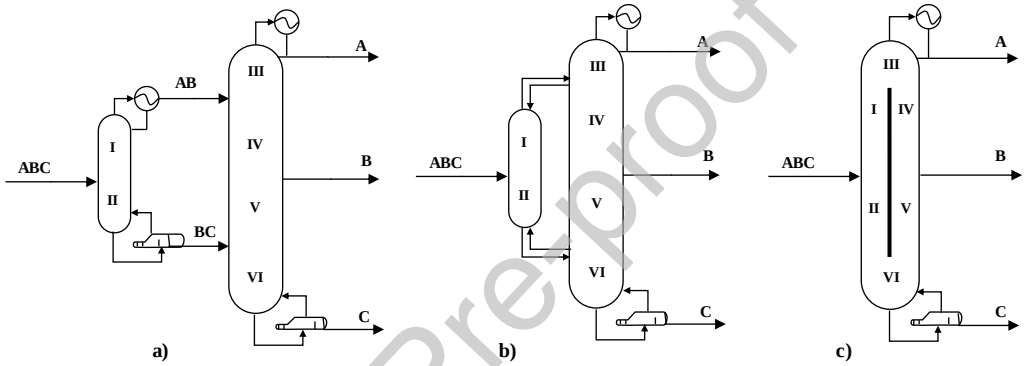

Figure 4. *a) Prefractionator configuration for three product separation, b) fully thermally coupled arrangement, or Petlyuk configuration, and c) the thermodynamically equivalent divided wall column (DWC). Strictly, the configurations in a and b are thermodynamically equivalent if there is no heat transfer across the wall.*

The superstructure representation for this problem considering both, the closed system (fully thermally coupled arrangement with the two material streams involved in a thermal couple), and the open system (fully thermally coupled arrangement with a combination of a material and energy stream) are shown in Figure 5a and b respectively. To facilitate the convergence, the open system superstructure is use in this work.

Given the feed with known composition, the required purity of each product and the column operating pressure, the objective is to determine the number of trays of each column section, as well as the column operating conditions that minimize the TAC.

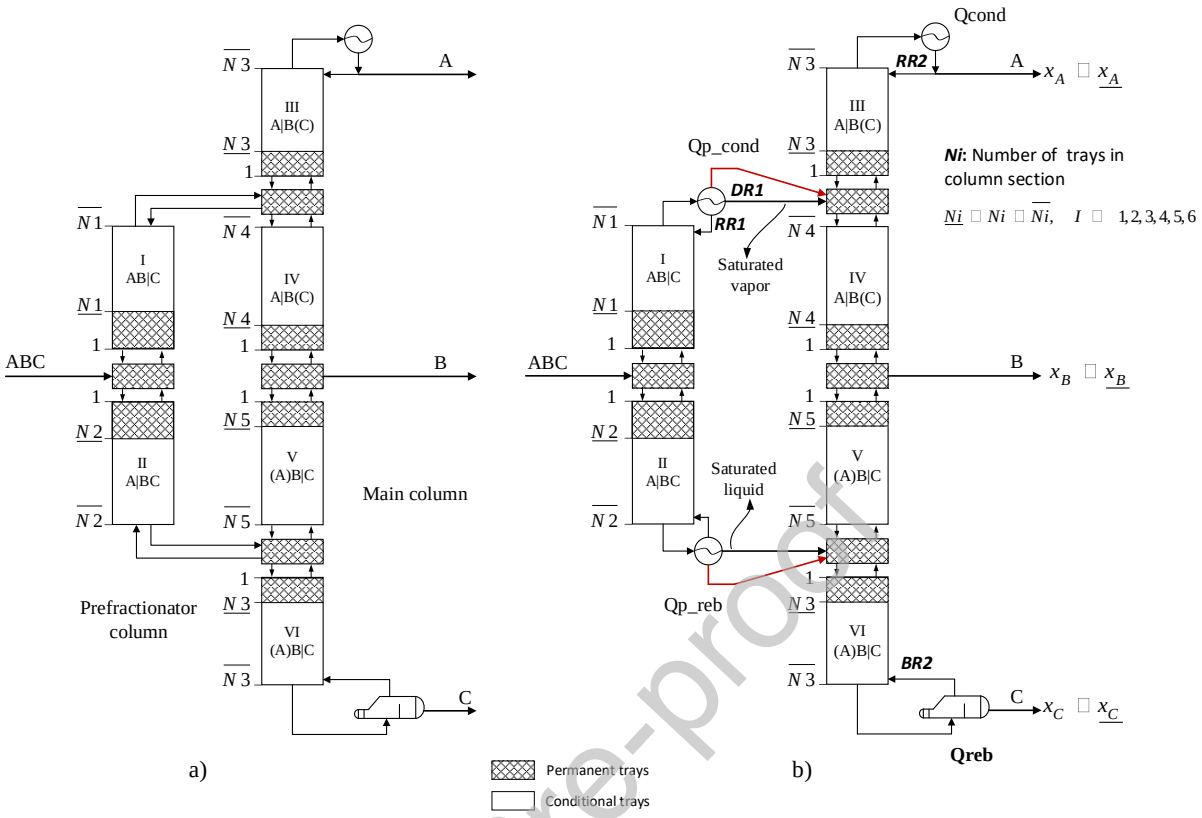

Figure 5. *Fully thermally coupled arrangement a) closed system (with vapor and liquid material recycle streams) b) open system (with one material and one energy stream).*

The MINLP optimization problem for this case study can be formulated

$$\min_{\mathbf{x}^{I,cont},\mathbf{x}^{I,int},\mathbf{x}^{D}} TAC = fC_{cap}\left(N_1,\mathrm{K},N_6,A_1,\mathrm{K},A_6,A_{cond},A_{reb}\right) + C_{op}\left(Q_{cond},Q_{reb}\right)$$

$$s.t.\ \mathbf{x}^{D} = h^{sim}\left(\mathbf{x}^{I,cont},\mathbf{x}^{I,\mathrm{int}}\right),$$

$$x_A^{dist} \geq \underline{x_A^{dist}},$$

$$x_B^{sdraw} \geq \underline{x_B^{sdraw}},$$

$$x_C^{btms} \geq \underline{x_C^{btms}},$$

$$\underline{\mathbf{x}^{I,cont}} \leq \mathbf{x}^{I,cont} \leq \overline{\mathbf{x}^{I,cont}},$$

$$\underline{\mathbf{x}^{I,\mathrm{int}}} \leq \mathbf{x}^{I,\mathrm{int}} \leq \overline{\mathbf{x}^{I,\mathrm{int}}},$$

$$\mathbf{x}^{I,cont} = \left\{RR1, DR1, RR2, BR2, Q_{reb}\right\} \in \mathbb{R}^{5}, \quad \mathbf{x}^{I,\mathrm{int}} = \left\{N_1,\mathrm{K},N_6\right\} \in \mathbb{Z}^{6},$$

$$\mathbf{x}^{D} = \left\{x_A^{dist}, x_B^{sdraw}, x_C^{btms}, Q_{cond}, A_1,\mathrm{K},A_6\right\} \in \mathbb{R}^{10}. \qquad \text{(P2)}$$

The TAC consists of the capital cost of the column (shell and trays), condenser and reboiler, as well as the operating cost related to steam and cooling water required by the reboiler and condenser. To estimate the capital cost of the column shell, we consider that the fully thermally coupled arrangement is built using a single shell with a vertical dividing wall as is depicted in Figure 4c.

The tower diameter, A_{DWC}, is estimated from the area of each of the six tray sections of the divided wall column as the $\max\left(A_3,\left(A_{1/2}+A_{4/5}\right),A_6\right)$, where $A_{1/2}=\max\left(A_1,A_2\right)$ and $A_{4/5}=\max\left(A_4,A_5\right)$. Now, if the areas A1 and A2, and A4 and A5 are all different from each other, we always overestimate the central tower diameter. To estimate the tray costs, we consider the areas A_3, $A_{1/2}$, $A_{4/5}$ and A_6, with their corresponding number of trays $\left(N_3+1\right)$, $\left(N_1+N_2+1\right)$, $\left(N_4+N_5+1\right)$ and $\left(N_6+1\right)$ respectively (we add one stage to take into account the feed/product stages) . The investment cost of the condenser and reboiler are obtained from the heat transfer areas A_{con} and A_{reb}. The operating costs, which is dominated by the steam required in the reboiler and cooling water in the condenser, are calculated from the heat loads Q_{reb} and Q_{cond} respectively.

The set of implicit functions, h^{sim}, that embodies the equations that describe the behavior of the separation arrangement are implemented in the process simulator. Three explicit constraints are written within the MATLAB environment to specify the three product purities requirements. Namely, (i) the light component mole fraction in the top column product stream x_A^{dist}, (ii) middle volatility component mole fraction in the side draw product stream x_B^{sdraw}, and (iii) heavy component mole fraction in bottoms product stream x_C^{btms}. For the mole fractions of these streams, we have certain requirements on the purity.

The design variables of the optimization problem are the six integer variables corresponding to the number of active trays in each section of the DWC, and the continuous variables corresponding to the degrees of freedom of the prefractionator and main distillation columns. Namely, the reflux ratio and distillate rate $RR1$ and $DR1$ of the prefractionator column, and the reflux ratio, boilup ratio and reboiler heat duty $RR2$, $BR2$ and Q_{reb} of the main tower.

A mixture of benzene, toluene and o-xylene is used to illustrate the performance of the separation of a three-component mixture using the fully thermally coupled model described above. Data for this example is presented in Table 4. The results are summarized in Table 5 and Table 6.

The best solution was found after only 9 iterations, but in total the ECP algorithm performed 100 iterations (due to multiple restarts). The total execution time was 1615.8 seconds, of which only 19.7 seconds was spent on solving the MILP subproblems.

Table 4. *Problem data for Case Study 2*

Feed molar flow (kmol/h)	500
Composition (mol fraction)	
Benzene (A)	0.17
Toluene (B)	0.39
o-Xylene (C)	0.44
Thermodynamic fluid package	SRK
Implicit constraints (as design spec.)	
Benzene purity in distillate (mf)	$x_A \geq 0.995$
Toluene purity in side draw (mf)	$x_B \geq 0.995$
o-Xylene purity in bottoms (mf)	$x_C \geq 0.995$
Bounds on independent variables	
N_1 *(number of trays in section I)*	$10 \leq N_1 \leq 30$

N_2 *(number of trays in section II)*	$10 \le N_2 \le 30$
N_3 *(number of trays in section III)*	$3 \le N_3 \le 20$
N_4 *(number of trays in section IV)*	$10 \le N_4 \le 25$
N_5 *(number of trays in section V)*	$15 \le N_5 \le 35$
N_6 *(number of trays in section VI)*	$10 \le N_6 \le 30$
$RR1$ *(reflux ratio column 1)*	$0.4 \le RR1 \le 1.7$
DR_1 *(distillate rate column 1, kmol/h)*	$170 \le DR1 \le 220$
$RR2$ *(reflux ratio column 2)*	$4.5 \le RR2 \le 5.5$
$BR2$ *(boilup ratio column 2)*	$2.2 \le BR2 \le 3.2$
Q_{reb} *(reboiler heat duty in MW)*	$4.0 \le Q_{reb} \le 6.5$

Table 5. *Best found design variables – Case study 2*

Design Variable	**N1**	**N2**	**N3**	**N4**	**N5**	**N6**	**RR1**	**DR1 [kmol/h]**	**RR2**	**BR2**	**Qreb [MW]**
Best values	16	19	8	15	30	25	0.7524	189.6814	5.1506	2.8085	5.6806

Table 6. *Objective function breakdown – Case study 2*

FIXED INVESTEMTN COST [$MM]	1.5572
Shell	0.9555
Trays	0.1056
Reboiler	0.2767
Condenser	0.2194
OPERATING COST [$MM/y]	2.4821
Low Pressure Steam [5 barg, 160 ºC]	2.4262
Cooling Water [30 to 40-45 ºC]	0.0559
TOTAL ANNUAL COST [$MM/y]	**2.7740**

6.3 Case study 3: extractive distillation process

The third case study is based on the economic optimization of the extractive distillation arrangement shown in Figure 6. This sort of enhanced distillation is widely applied to separate minimum-boiling homogeneous azeotropes and other mixtures that have key components with close relative volatilities (Seader and J. Henley, 2006).

We consider the industrially relevant separation of ethanol (nbp = 78.31ºC) and water (nbp =100.02 ºC) using ethylene glycol (nbp = 197.08ºC) as the solvent to illustrate and test the simulation-based optimization of this two-column sequence distillation process. Ethanol and water form a minimum-boiling homogeneous azeotrope at 78.15ºC and 1 atm with composition 0.895%mol ethanol.

The MINLP optimization problem for this case study can be conceptually described by

$$\min_{\mathbf{x}^{I,cont},\mathbf{x}^{I,\text{int}},\mathbf{x}^{D}} TAC = fC_{cap}\left(N_1,\text{K},N_5,A_{C1},A_{C2},A_{cond1},A_{reb1},A_{cond2},A_{reb2},M_{solv}\right) + C_{op}\left(Q_{CondC1},Q_{RebC1},Q_{CondC2},Q_{RebC2}\right)$$

$$s.t.\ \mathbf{x}^{D} = h^{simulator}\left(\mathbf{x}^{I,cont},\mathbf{x}^{I,\text{int}}\right),$$

$$rec_{LK1}^{distC1} \geq \underline{rec}_{LK1}^{distC1},$$

$$x_{LK1}^{distC1} \geq \underline{x}_{LK1}^{distC1},$$

$$x_{HK2}^{btmsC2} \geq \underline{x}_{HK2}^{btmsC2},$$

$$rec_{HK2}^{btmsC2} \geq \underline{rec}_{HK2}^{btmsC2},$$

$$\underline{\mathbf{x}}^{I,cont} \leq \mathbf{x}^{I,cont} \leq \overline{\mathbf{x}^{I,cont}},$$

$$\underline{\mathbf{x}}^{I,\text{int}} \leq \mathbf{x}^{I,\text{int}} \leq \overline{\mathbf{x}^{I,\text{int}}},$$

$$\mathbf{x}^{I,cont} = \left\{RR1,BR1,RR2,BR2,M_{solv}\right\} \in \mathbb{R}^{5},\quad \mathbf{x}^{I,\text{int}} = \left\{N_1,\text{K},N_5\right\} \in \mathbb{Z}^{5},$$

$$\mathbf{x}^{D} = \left\{\begin{array}{l} x_{LK1}^{dist1},\ rec_{LK1}^{dist1},x_{HK2}^{btms2},rec_{HK2}^{btms2},Q_{cond1},Q_{reb1}, \\ Q_{cond2},Q_{reb2},A_{C1},A_{C2},A_{cond1},A_{reb1},A_{cond2},A_{reb2} \end{array}\right\} \in \mathbb{R}^{14}. \qquad \text{(P3)}$$

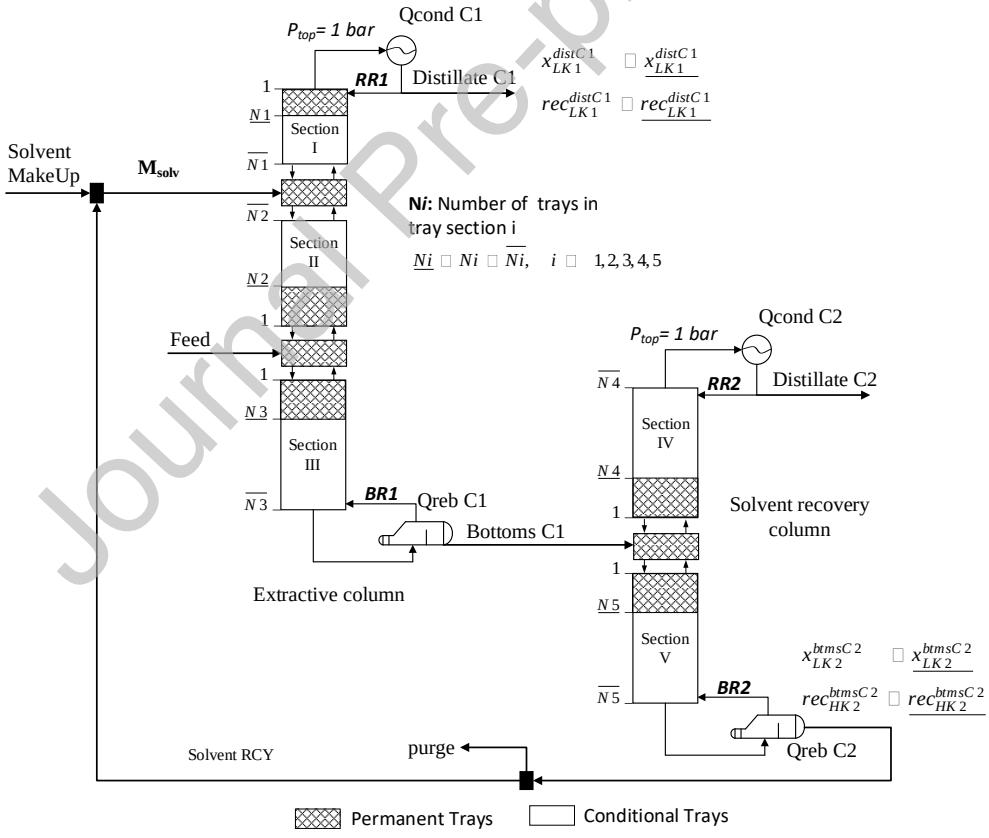

Figure 6. *Extractive distillation with conventional solvent superstructure.*

The goal is to find the extractive and solvent column topologies (number of trays and feed tray locations), operating conditions (reflux and boilup ratios), as well as the molar flowrate of solvent, that minimize total annual cost for a

given feed and product stream specifications. The TAC comprises the investment and operating cost of the process. The investment cost contains: 1) cost of the extractive column (shell and trays) given by the number of trays N_1, N_2 and N_3, and column area A_{C1}, 2) cost of the solvent recovery column (shell and trays) given by the number of trays N_4 and N_5, and column area A_{C2}, 3) the cost of the condenser and reboiler of each distillation column given by their heat transfer areas A_{cond1}, A_{reb1} and A_{cond2}, A_{reb2}. 4) the cost of the solvent (ethylene glycol) is considered as an investment cost. The cost of the solvent is obtained from the total molar flowrate of solvent fed to the extractive column M_{solv}. The operating cost of the process is estimated from the heat duties supplied in both reboilers Q_{reb1} and Q_{reb2}, and heat duties removed in the condensers of each column Q_{cond1} and Q_{cond2}.

Similarly to the other examples, the behavior of the extractive distillation process is described by the equality constraints containing the implicit function h^{sim}. This function represents the process simulator and is evaluated before the other explicit constraints. The inputs required by h^{sim} are the set of independent variables of the problem $\mathbf{x}^{I,\text{int}}$ and $\mathbf{x}^{I,cont}$, namely, the number of trays of each column section N_i with $i = \{1,\text{K},5\}$, the reflux ratio of both columns $RR1$ and $RR2$, the boilup ratios $BR1$ and $BR2$, and the solvent molar flowrate M_{solv}. The explicit constraints define (i) a minimum mole fraction $\underline{x_{LK1}^{dist1}}$, (ii) minimum recovery $\underline{rec_{LK1}^{dist1}}$ for the light-key component at the top of the extractive column (ethanol), (iii) minimum mole purity $\underline{x_{HK2}^{btms2}}$, and (iv) minimum recovery $\underline{rec_{HK2}^{btms2}}$ for the solvent at the bottoms of the solvent recovery column. The data for this example is shown in Table 7, and the results in Table 8 and Table 9.

In total the ECP algorithm performed 100 iterations (iteration limit), and the best solution was found after only 11 iterations. The total execution time was 1628.7 second, of which only 23.5 seconds was spent on solving the MILP subproblems. Out of all the case studies, we encountered most flowsheet convergence failures here. The no-good cut alone is enough to handle convergence failures, but the backtracking technique was in practice significantly more efficient. The backtracked solutions allowed the algorithm to find points where the flowsheet converge, and the derived cuts successfully directed the search away from areas with convergence failures.

Table 7. *Problem data for Case Study 3.*

Feed molar flow (kmol/h)	300
Columns top pressure (bar.)	1
Composition (mol fraction)	
Ethanol	0.855
Water	0.115
Thermodynamic fluid package	NRTL
Pressure drop per tray (bar)	0.0069
Explicit constraints	
Ethanol mol frac. in Ext. col. ovhd. stream	$x_{LK1}^{distC1} \geq 0.996$
Ethanol % recovery in Extr. col. ovhd. stream	$rec_{LK1}^{distC1} \geq 99.80$
EG mol % in Solv. Rec. col btms. stream	$x_{HK2}^{btmsC2} \geq 0.998$
EG % recovery in Solv. rec. col. btms. stream	$rec_{HK2}^{btmsC2} \geq 99.95$
Bounds on independent variables	

N_1 *(number of trays section I – Ext. co.l)*	$2 \le N_1 \le 10$
N_2 *(number of trays section II – Ext. col.)*	$10 \le N_2 \le 40$
N_3 *(number of trays section III – Ext. col)*	$5 \le N_3 \le 20$
N_4 *(number of trays section IV– Solv. rec. col)*	$3 \le N_4 \le 15$
N_5 *(number of trays section V – Solv. rec. col)*	$3 \le N5 \le 15$
$RR1$ *(extractive column reflux ratio)*	$0.5 \le RR1 \le 1.1$
$BR1$ *(extractive column boilup ratio)*	$1.2 \le BR1 \le 1.9$
$RR2$ *(solvent rec. column reflux ratio)*	$0.3 \le RR2 \le 1.5$
$BR2$ *(solvent rec. column boilup ratio)*	$0.3 \le BR2 \le 0.9$
M_{SOLV} *(solvent molar flowrate in kmol/h)*	$170 \le M_{SOLV} \le 200$

Table 8. *Best found design variables – Case study 3.*

Design Variable	**N1**	**N2**	**N3**	**N4**	**N5**	**RR1**	**BR1**	**RR2**	**BR2**	**SOLV (kmol/h)**
Best value	2	36	8	4	6	0.7164	1.6453	0.4535	0.3485	165.255

Table 9. *Objective function breakdown – Case study 3.*

	Extractive col.	Solvent rec. col.
FIXED INVESTEMTN COST [$MM]	1.1733	0.1845
Shell	0.5797	0.0734
Trays	0.0335	0.0017
Reboiler	0.3396	0.0698
Condenser	0.2128	0.0396
Solvent	0.0077	---
OPERATING COST [$MM/y]	2.0095	0.4507
High Pressure Stream [40 barg, 254 ºC]	---	0.4450
Medium Pressure Steam [10 barg, 184 ºC]	1.9587	---
Cooling Water [30 to 40-45 ºC]	0.0508	0.0057
TOTAL ANNUAL COST [$MM/y]	**2.7147**	

6.4 Case study 4: vapor recompression distillation process

Our final case study deals with the economic optimization of a propylene-propane splitter using the vapor recompression heat pump assisted distillation superstructure shown in Figure 7. The separation of this mixture by distillation is energy intensive and requires large distillation columns due to their close volatilities. In fact, the purification of propylene-propane (and ethylene-ethane) alone accounts for 0.3% of global energy consumption (Sholl and Lively, 2016). Typically, these separations are carried out using vapor recompression systems as the one proposed in Figure 7. Based on this superstructure, the objective of this case study is to determine the distillation column topology (number of trays and feed location), column operating conditions (reflux and boilup ratios), column top pressure, and vapor recompression (VRC) operating conditions (compressor outlet pressure and VRC heat exchanger

outlet temperature) that minimizes the total annualized cost of equipment and utilities for a given feed and overhead product stream specifications (propylene purity and recovery).

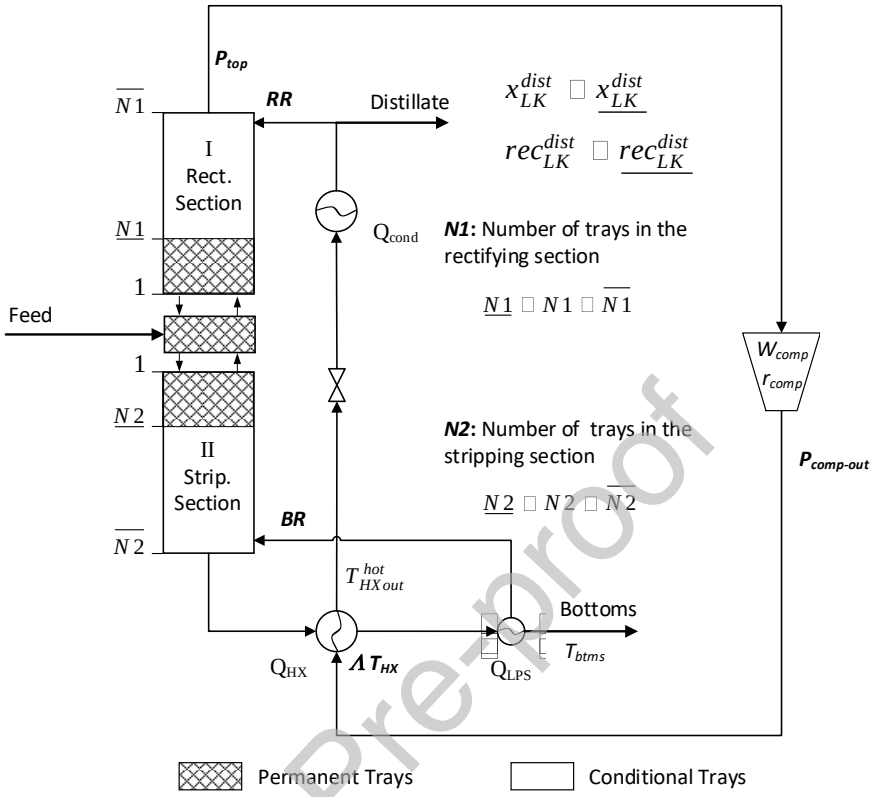

Figure 7. *Vapor recompression (VRC) heat pump assisted distillation superstructure.*

The optimization problem can be conceptually written as

$$\min_{x^{I,cont},x^{I,int},x^{D}} TAC = fC_{cap}\left(N_{COL},A_{COL},A_{cond},A_{HX},A_{Ext\ reb},W_{comp}\right)+C_{op}\left(Q_{cond},Q_{LPS},W_{comp}\right)$$

$$s.t.\ x^{D} = h^{sim}\left(x^{I,cont},x^{I,int}\right),$$

$$x_{dist,LK} \geq \underline{x}_{dist,LK},$$

$$rec_{dist,LK} \geq \underline{rec}_{dist,LK},$$

$$T_{dist} \geq \underline{T}_{dist},$$

$$\left(T_{btms} - T_{HXout}^{hot}\right) \geq \underline{\Delta T}_{\min},$$

$$\underline{x}^{I,cont} \leq x^{I,cont} \leq \overline{x^{I,cont}},$$

$$\underline{x}^{I,int} \leq x^{I,int} \leq \overline{x^{I,int}},$$

$$x^{I,cont} = \left\{RR,BR,P_{top},P_{comp\text{-}out},LT_{HX}\right\} \in \mathbb{R}^{5},\quad x^{I,int} = \left\{N1,N2\right\} \in \mathbb{Z}^{2},$$

$$x^{D} = \left\{T_{dist},T_{btms},T_{HX\text{-}out},x_{dist,LK},rec_{dist,LK},Q_{cond},Q_{LPS},W_{comp},A_{COL},A_{cond},A_{HX},A_{Ext\ reb}\right\} \in \mathbb{R}^{12}.$$

The objective is to minimize the TAC, which comprises the annualized investment cost of the distillation column, heat exchangers and compressor, as well as the operating cost due to the compressor power, and cold and hot (if required) utilities. Specifically, the capital cost of the distillation column depends on the number of trays N_{COL} $\left(N_{COL} = N_1 + 1 + N_2\right)$, and column area A_{COL}. The cost of each heat exchanger depends on their heat transfer areas: A_{cond} for the column condenser; A_{HX} for the column reboiler (which acts as a condenser from the point of view of the VRC); and $A_{Ext-reb}$ for an additional reboiler that may be employed if the heat supplied by the vapor compression system is insufficient to provide the required column boilup at its corresponding temperature. In addition, the compressor investment cost depends on its brake horsepower, W_{comp}. The main operating cost is given by the electricity required by the compressor, W_{comp}, and to a lesser extent to the cooling utility Q_{cond} (cooling water or refrigerated water, according to the column operating pressure) and extra low pressure steam, Q_{LPS}, if required.

As in previous examples, the function h^{sim} refers to the set of implicit constraints which are evaluated firstly at the level of the process simulator using the values of the independent variables $\mathbf{x}^I$, which in this case are the column reflux and boilup ratio, RR and BR respectively; the column top operating pressure, P_{top}; compressor outlet pressure, $P_{comp-out}$; degree of subcooling in column reboiler $\mathrm{L}T_{HX}$; and the number of active trays in the rectifying and stripping sections of the column, $N1$ and $N2$. Additionally, four explicit constraints are defined within MATLAB. These constraints are evaluated after the implicit constraints, and include the (i) top product light key mole fraction $x_{dist,LK}$, (ii) recovery $rec_{dist,LK}$, and (iii) top product temperature T_{dist} requirements, as well as the (iv) minimum temperature difference condition between the bottoms product stream T_{btms} and hot outlet stream of the column reboiler T_{HX-out}. The molar flowrate and composition of the feed stream, thermodynamic package, and other main data for this case study are shown in Table 10.

The best found solution is presented in Table 11, along with more details of the corresponding costs in Table 12. Once more the ECP algorithm performed 100 iterations, and the best solution was found after 42 iterations. The total execution time was 1214.2 seconds, of which only 4.0 seconds was spent on solving the MILP subproblems. The feasibility restoration technique was an essential component, and without it we would have failed to even obtain a feasible solution. The restarting technique allowed us to find significantly better solutions, and two of the restarts resulted in improved solutions.

Table 10. *Problem data for Case Study 4*

Feed molar flow (kmol/h)	300
Composition (mol fraction)	
Propylene	0.52
Propane	0.47
i-Butane	0.01
Thermodynamic fluid package	Peng-Robinson
Explicit constraints	

distillate temperature (°C)	T_{dit} = 22
Propylene mol frac. in ovhd product stream	x_{C3H6}^{dist} = 0.996
Propylene recovery in ovhd product stream (%)	rec_{C3H6}^{dist} = 99.5
Column reboiler minimum temp. diff. (°C)	ΔT_{min} = 10
Bounds on independent variables	
N_1 *(number of trays in section I)*	100 ≤ N_1 ≤ 145
N_2 *(number of trays in section II)*	60 ≤ $N2$ ≤ 110
RR *(reflux ratio)*	14 ≤ RR ≤ 20
BR *(boilup ratio)*	16 ≤ BR ≤ 24
P_{top} *(column top pressure, bar)*	10.5 ≤ P_{top} ≤ 15.0
$P_{comp\text{-}out}$ *(compressor outlet pressure, bar)*	25 ≤ $P_{comp\text{-}out}$ ≤ 35
ΔT_{HX} *(degree of subcooling)*	10 ≤ ΔT_{HX} ≤ 35

Table 11. *Best found design variables – Case study 4.*

Design Variable	**N1**	**N2**	**RR**	**BR**	**P_{top}(bar)**	**$P_{comp\text{-}out}$ (bar)**	**ΔT_{HX} (ºC)**
Best value	140	70	15.4751	18.2952	12.1381	26.2744	12.5608

Table 12. *Objective function breakdown – Case study 4.*

FIXED INVESTEMTN COST [$MM]	8.4203
Shell	2.1407
Trays	0.8824
Reboiler	0.5412
Condenser	0.2523
Extra reboiler	0.0227
Compressor	4.5810
OPERATING COST [$MM/y]	1.1220
Electricity	0.6992
Refrigerated Water [5 to 15 ºC]	0.2447
Extra Low Pressure Steam [5 barg, 160 ºC]	0.1781
TOTAL ANNUAL COST [$MM/y]	**2.7004**

7 Comparison of solutions obtained using the ECP solver and two stochastic global search methods.

To evaluate the performance of the proposed simulation-based optimization approach, for the case studies, we present a brief comparison with two popular population-based stochastic global search algorithms. The main purpose of this comparison is to obtain an alternative solution with a derivative free optimization method to evaluate the quality of the solutions obtained by our approach.

From the board range of derivative free optimization techniques, a proprietary global best integer version of the particle swarm optimization (PSO) algorithm (Kennedy and Eberhart, 1997) and the genetic algorithm (GA) available in MATLAB R2021b (The MathWorks, Inc. A) are used in this paper. We choose the PSO algorithm because we have previous experience using it, and it performed well for similar problems (Javaloyes et al., 2013). We also included the GA because it is one of the most popular stochastic global search methods, and it has been used for simulation-based optimization problems in the literature, for instance in Chia et al., 2021, Ibrahim et al., 2017, Vazquez–Castillo et al., 2009, Leboreiro and Acevedo, 2004 and Gross and Roosen, 1998.

As mentioned in section **Error! Reference source not found.**, DFO algorithms are particularly appropriate for sequential modular process simulators as the issue with obtaining accurate derivatives is avoided directly. In addition, if the flowsheet convergence fails for a given set of design variables during the optimization procedure, then, the approach to continue with the optimization it is more straightforward than in the case of derivative-based solvers. In essence, we only need to restore the simulation convergence using a feasible set of design variables (for instance, the best feasible point found so far) and penalize the objective function. Naturally, not all are advantages and one of the main drawbacks is related with solving optimization problems with constrained search spaces. In this work, for the PSO algorithm we handle the constrains by means of one of the most popular techniques, which is based on the fitness penalization of a solution as follows (Mezura-Montes and Flores-Mendoza, 2009):

$$f(\mathbf{x}) = f(\mathbf{x}) + p(\mathbf{x})$$

where $f(\mathbf{x})$ is the expanded objective function, $f(\mathbf{x})$ is the original objective function, and $p(\mathbf{x})$ a penalty function. A common penalty function that can be used is the following:

$$p(\mathbf{x}) = \sum_{j \in J} w_j \max\left[0, g_j(\mathbf{x})\right]$$

where w_j is a positive penalty factor. The constraint-handling technique implemented in the genetic algorithm coded in MATLAB is also based on penalty functions (The MathWorks, Inc. B).

Table 13 compares the values of the objective function corresponding to ECP, PSO and MATLAB R2021b GA algorithms. As shown, the results obtained with the proposed methodology are slightly better than the ones found by the PSO and GA in all cases.

Table 13. *Objective function comparison between ECP, PSO and GA algorithms.*

Case Study	I	II	III	IV
ECP	3.2571	2.7740	2.7147	2.7004
PSO	3.3977	2.7994	2.7276	2.8240
GA	3.2832	2.7985	2.7324	2.7857

It is worth mentioning that since the PSO and GA belongs to a class of stochastic global search algorithms, in some points of the procedure random numbers are used to update the values for the variables. Thus, convergence to the same solution is not guaranteed and the values reported in the above table corresponds to the best ones found after 30 consecutive executions. Both stochastic algorithms were tested under the same conditions (same population size and maximum number of iterations/generations). The total execution time of the ECP solver is somewhat longer than a single run of the PSO or GA solver (twice as long for two of the case studies). The execution times with the ECP solver could have been reduced significantly by reducing the number of restarts as the best solutions were all found early, but we wanted to keep searching for better solutions. Also, keep in mind that we repeated the runs 30 times with the PSO and GA solver to obtain these results. Therefore, the total run time for each case study was significantly longer with the derivative free optimization algorithms. More details for the results are provided in Supplementary Material – section 2.

8 Conclusions

With the techniques presented in this paper, we were able to successfully apply a cutting plane algorithm to several simulation based mixed-integer nonlinear optimization problems and overcome some well know challenges in using a derivative-based algorithm. We presented two main approaches for dealing with simulation failures, based on no-good cuts and backtracking. These approaches allowed us to successfully deal with simulation failures, and are not unique to our algorithm but applicable to other derivative-based algorithms. We also propose a restarting technique, and several restarts were performed for all the case studies and resulted in significantly better solutions for two of the four case studies. The restarting technique utilizes information accumulated from previous iterations to reduce the number of function evaluation while expanding the search space to continue searching for better solutions. As mentioned earlier, the feasibility restoration of the master problem is an essential component, and without it we would have failed to find a feasible solution to two of the case studies. Infeasible master problems are, at least partially, a consequence of inaccurate derivatives, and this is effectively handled by the feasibility restoration. For all the case studies, the total time spent on solving the MILP subproblems is insignificant in comparison to the time spent on running the process simulator, even with the added complexity of the no-good cuts. The comparison to a PSO and GA algorithm shows that the proposed algorithm finds good solutions to the case studies, by finding a better solution to all the test problems than the two popular stochastic global search algorithms tested.

Finally, we would like to remark that the proposed ECP can be considered as a competitive and viable method to solve simulation-based optimization MINLP problems with the general structure of problem given in equation (1).

Acknowledgments

The authors JJ and JC acknowledge financial support from the “Generalitat Valenciana” under project PROMETEO 2020/064. The author JK acknowledges financial support by a Newton International Fellowship from the Royal Society (NIF\R1\182194) and a grant by the Swedish Cultural Foundation in Finland.

Appendix A. Total annual cost correlations

The equipment cost of each distillation system is estimated based on the cost correlations proposed by Guthrie (1974), and gather by Douglas in his book Conceptual Design of Chemical Processes (1988) .

The correlations given below have been adapted from the original ones for directly using the international system of units instead of the English ones. Besides, as carbon steel is the construction material selected for all equipment, the corresponding material correction factor has been applied.

A1. Column shell

$$Installed\ Cost\ (\$) = \left(\frac{M\ \&\ S}{280}\right) 937.64\, D^{1.066}\, H^{0.802}\, (3.18 + F_p),$$

where D is the column diameter (m), H the column height (m), and F_p the pressure factor given in Table A14.

The distillation column diameters have been estimated with the Tray Sizing Analysis tool of Aspen HYSYS using the Fair's tray flood method considering sieve trays and the default values for the tray properties.

Table A14. *Pressure factor for column shell correlation*

Pressure, bar.	**Up to 3.45**	**6.89**	**13.79**	**20.68**	**27.58**	**34.47**	**41.37**
F_p	1.00	1.05	1.15	1.20	1.35	1.45	1.60

A2. Distillation column trays and tower internals

We consider that all distillation columns are built using sieve trays. In that case, the installed cost is given by:

$$Installed\ Cost\ (\$) = \left(\frac{M\ \&\ S}{280}\right) 59.28\, D^{1.55}\, N\, F_s,$$

were D is the column diameter (m), N total number of trays, and F_s the tray spacing correction factor, which is given in Table A15.

Table A15. *Tray spacing factor for column trays and tower internals correlations*

Tray spacing, in.	**24**	**18**	**12**
F_s	1.0	1.4	2.2

A3. Heat exchangers

The installed cost for the heat exchangers is estimated as a function of the heat exchange area, which is obtained from $A = Q/UDT$. Typical values for the overall heat transfer coefficient U are used (reboiler 820 W/m^2 K, condenser 800 W/m^2 K)

- Process stream heat exchangers and column condensers (fixed tube heat exchanger)

  $$Installed\ Cost\ (\$) = \left(\frac{M\ \&\ S}{280}\right) 474.69\, A^{0.65}\, (3.09 + F_p)$$

- Column reboiler (kettle reboiler)

  $$Installed\ Cost\ (\$) = \left(\frac{M\ \&\ S}{280}\right) 474.69\, A^{0.65}\, (3.64 + F_p)$$

where A is the heat exchanger area (m^2), and F_p the pressure correction factor (Table A16).

Table A16. *Pressure factor for heat exchanger correlations*

Pressure, bar.	**Up to 10.34**	**20.68**	**27.58**	**55.16**
$\mathbf{F_p}$	1.00	1.05	1.15	1.20

A4. Gas compressors

A centrifugal motor compressor is used in the vapor recompression system included in case study 4. The cost of this piece of equipment is given by

$$Installed\ Cost\ (\$) = \left(\frac{M\ \&\ S}{280}\right) 2047.24\, bhp^{0.65}\ ,$$

where *bhp* is the compressor brake horsepower (kW).

A5. Update cost factor

The original cost correlations given by Douglas (1988) are updated to the current year through the following update cost factor

$$updateFactor\ = \left(\frac{M\ \&\ S_{current}}{280}\right).$$

However, as the Marshall and Swift Equipment Cost Index is not available in the Chemical Engineering magazine from June 2012 (Chemical plant cost indexes), we modified the update factor as follows

$$updateFactor\ = \left(\frac{M\ \&\ S_{2010}}{280}\right)\left(\frac{CPCI_{current}}{CPCI_{2010}}\right),$$

where $M\&S_{2010}$ is the last annual average value of the Marshall and Swift Equipment Cost Index published in Chemical Engineering magazine on April 2012 ($M\&S_{2010}$ = 1457.4), $CPCI_{2010}$ is the annual Chemical Engineering Plant Cost Index corresponding to 2010, and $CPCI_{current}$ is the annual average value of CEPCI for 2018 ($CPCI_{current}$ = 603.1), published in the December 2019 issue of Chemical Engineering magazine.

A6. Utility costs

Prices for the main utilities required by the distillation systems considered in the case studies are given in Table A17

Table A17. *Utility costs* (Turton et al., 2008)

Utilities	**Price ($/MW h)**
Hot Utilities	
High pressure steam HPS (41 barg, 254ºC)	63.7
Medium pressure steam MPS (10 barg, 184ºC)	53.4
Low pressure steam LPS (5 barg, 160ºC)	50.6
Cold Utilities	
Cooling water CW (30 ºC to ~40 ºC)	1.3
Refrigerated water RW (5 ºC to 15 ºC)	15.9
Power	60.0

Declaration of interests

☒ 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.

Author Contributions

J. K. and J. J. A. were the main responsible authors of the paper. They conceived the research project, designed the case studies, implemented and developed the methods, and wrote the paper. In particular, J. K. was focusing more on the development of the ECP method and J. J. A. and J. A. C. on the development of the interface between the process simulators and solvers, design of case studies, and comparison with the PSO and GA algorithms.

References

Aspelund, A., Gundersen, T., Myklebust, J., Nowak, M.P., Tomasgard, A., 2010. An optimization-simulation model for a simple LNG process. Comput. Chem. Eng. 34, 1606–1617. doi:10.1016/j.compchemeng.2009.10.018

Balas, E., 1979. Disjunctive Programming. Ann. Discret. Math. 5, 3–51.

Bernal, D.E., Vigerske, S., Trespalacios, F., Grossmann, I.E., 2020. Improving the performance of DICOPT in convex MINLP problems using a feasibility pump. Optim. Methods Softw. 35, 171–190. doi:10.1080/10556788.2019.1641498

Bhosekar, A., Ierapetritou, M., 2018. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. Comput. Chem. Eng. 108, 250–267. doi:10.1016/j.compchemeng.2017.09.017

Biegler, L.T., 2010. Nonlinear Programing: Concepts, Algorithms, and Applications to Chemical Processes. Society for Industrial and Applied Mathematics.

Biegler, L.T., Hughes, R.R., 1982. Infeasible path optimization with sequential modular simulators. AIChE J. 28, 994–1002. doi:10.1002/aic.690280615

Boukouvala, F., Misener, R., Floudas, C.A., 2016. Global optimization advances in Mixed-Integer Nonlinear Programming, MINLP, and Constrained Derivative-Free Optimization, CDFO. Eur. J. Oper. Res. 252, 701–727. doi:10.1016/j.ejor.2015.12.018

Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

Brochu, E., Cora, V.M., de Freitas, N., 2010. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning.

Brunet, R., Reyes-Labarta, J.A., Guillén-Gosálbez, G., Jiménez, L., Boer, D., 2012. Combined simulation–optimization methodology for the design of environmental conscious absorption systems. Comput. Chem. Eng. 46, 205–216. doi:10.1016/j.compchemeng.2012.06.030

Caballero, J.A., 2015. Logic hybrid simulation-optimization algorithm for distillation design. Comput. Chem. Eng. 72, 284–299. doi:10.1016/j.compchemeng.2014.03.016

Caballero, J.A., Grossmann, I.E., 2008. An Algorithm for the Use of Surrogate Models in Modular Flowsheet Optimization. AIChE J. 54, 2633–2650. doi:10.1002/aic

Caballero, J.A., Grossmann, I.E., 2003. Thermodynamically equivalent configurations for thermally coupled distillation. AIChE J. 49, 2864–2884. doi:10.1002/aic.690491118

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. doi:10.1021/ie050080l

Chemical plant cost indexes [WWW Document], n.d. URL https://en.wikipedia.org/wiki/Chemical_plant_cost_indexes (accessed 12.23.19).

Chia, D.N., Duanmu, F., Sorensen, E., 2021. Optimal Design of Distillation Columns Using a Combined Optimisation Approach, Computer Aided Chemical Engineering. Elsevier Masson SAS. doi:10.1016/B978-0-323-88506-5.50025-5

Cozad, A., Sahinidis, N. V, Miller, D.C., 2014. Learning Surrogate Models for Simulation-Based Optimization. AIChE J. 60, 2211–2227. doi:https://doi.org/10.1002/aic.14418

D'Ambrosio, C., Frangioni, A., Liberti, L., Lodi, A., 2010. On interval-subgradient and no-good cuts. Oper. Res. Lett. 38, 341–345. doi:10.1016/j.orl.2010.05.010

Dantus, M.M., High, K.A., 1999. Evaluation of waste minimization alternatives under uncertainty: a multiobjective optimization approach. Comput. Chem. Eng. 23, 1493–1508. doi:10.1016/S0098-1354(99)00307-5

Díaz, M.S., Bandoni, J.A., 1996. A mixed integer optimization strategy for a large scale chemical plant in operation. Comput. Chem. Eng. 20, 531–545. doi:10.1016/0098-1354(95)00209-X

Douglas, J.M., 1988. Conceptual Design of Chemical Processes. McGraw-Hill.

Duran, M.A., Grossmann, I.E., 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. 36, 307–339. doi:10.1007/BF02592064

Emet, S., Westerlund, T., 2004. Comparisons of solving a chromatographic separation problem using MINLP methods. Comput. Chem. Eng. 28, 673–682. doi:10.1016/j.compchemeng.2004.02.010

Eronen, V.P., Mäkelä, M.M., Westerlund, T., 2015. Extended cutting plane method for a class of nonsmooth nonconvex MINLP problems. Optimization 64, 641–661. doi:10.1080/02331934.2013.796473

Gross, B., Roosen, P., 1998. Total process optimization in chemical engineering with evolutionary algorithms. Comput. Chem. Eng. 22, S229–S236. doi:10.1016/S0098-1354(98)00059-3

Halvorsen, I.J., Skogestad, S., 2003a. Minimum energy consumption in multicomponent distillation. 2. Three-product Petlyuk arrangements. Ind. Eng. Chem. Res. 42, 605–615.

Halvorsen, I.J., Skogestad, S., 2003b. Minimum energy consumption in multicomponent distillation. 3. More than three products and generalized Petlyuk arrangements. Ind. Eng. Chem. Res. 42, 616–629.

Halvorsen, I.J., Skogestad, S., 2003c. Minimum energy consumption in multicomponent distillation. 1. Vmin diagram for a two-product column. Ind. Eng. Chem. Res. 42, 596–604.

Henao, C.A., Maravelias, C.T., 2011. Surrogate-Based Superstructure Optimization Framework. AIChE J. 57, 1216–1232. doi:10.1002/aic

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. doi:10.1021/acs.iecr.7b01014

Javaloyes-Antón, J., Kronqvist, J., Caballero, J.A., 2018. Simulation-Based Optimization of Chemical Processes Using the Extended Cutting Plane Algorithm. Comput. Aided Chem. Eng. 43, 463–469. doi:https://doi.org/10.1016/B978-0-444-64235-6.50083-8

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. doi:10.1021/ie400918x

Kennedy, J., Eberhart, R.C., 1997. Discrete binary version of the particle swarm algorithm, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. IEEE, pp. 4104–4108.

Kenneth M. Guthrie, 1974. Process Plant Estimating, Evaluation and Control. Craftsman Book Co.

Kronqvist, J., Bernal, D.E., Grossmann, I.E., 2020. Using regularization and second order information in outer approximation for convex MINLP. Math. Program. 180, 285–310. doi:10.1007/s10107-018-1356-3

Kronqvist, J., Bernal, D.E., Lundell, A., Grossmann, I.E., 2019. A review and comparison of solvers for convex MINLP, Optimization and Engineering. Springer US. doi:10.1007/s11081-018-9411-8

Kronqvist, J., Lundell, A., Westerlund, T., 2017. A center-cut algorithm for solving convex mixed-integer nonlinear programming problems. Comput. Aided Chem. Eng. 40, 2131–2136. doi:10.1016/B978-0-444-63965-3.50357-

3

Leboreiro, J., Acevedo, J., 2004. Processes synthesis and design of distillation sequences using modular simulators: a genetic algorithm framework. Comput. Chem. Eng. 28, 1223–1236. doi:10.1016/j.compchemeng.2003.06.003

Lundell, A., Kronqvist, J., 2020. On Solving Nonconvex MINLP Problems with SHOT, in: Le Thi, H.A., Le, H.M., Pham Dinh, T. (Eds.), Optimization of Complex Systems: Theory, Models, Algorithms and Applications. Springer International Publishing, Cham, pp. 448–457.

Luyben, W.L., 2006. Distillation Design and Control Using Aspen Simulation. John Wiley & Sons.

Martín, M.M., 2014. Introduction to Sofware for Chemical Engineers. CRC Press.

Mezura-Montes, E., Coello Coello, C.A., 2011. Constraint-handling in nature-inspired numerical optimization: Past, present and future. Swarm Evol. Comput. 1, 173–194. doi:10.1016/j.swevo.2011.10.001

Mezura-Montes, E., Flores-Mendoza, J.I., 2009. Improved particle swarm optimization in constrained numerical search spaces, in R. Chiong (Ed.), Nature-Inspired Algorithms for Optimization, Studies in. ed, Studies in computational Intelligence series. Springer. doi:10.1007/978-3-642-00267-0

Nannicini, G., Belotti, P., 2012. Rounding-based heuristics for nonconvex MINLPs. Math. Program. Comput. 4, 1–31. doi:10.1007/s12532-011-0032-x

Navarro-Amorós, M. a., Ruiz-Femenia, R., Caballero, J. a., 2014. Integration of modular process simulators under the Generalized Disjunctive Programming framework for the structural flowsheet optimization. Comput. Chem. Eng. 67, 13–25. doi:10.1016/j.compchemeng.2014.03.014

Navarro, M.A., Javaloyes, J., Caballero, J.A., Grossmann, I.E., 2012. Strategies for the robust simulation of thermally coupled distillation sequences. Comput. Chem. Eng. 36, 149–159.

Petlyuk, F.B., Platonov, V.M., Slavinsk, D., 1965. Thermodynamically optimal method for separating multicomponent mixtures. Int. Chem. Eng. 5, 555.

Quirante, N., Javaloyes-Antón, J., Caballero, J.A., 2018. Hybrid simulation-equation based synthesis of chemical processes. Chem. Eng. Res. Des. 132, 766–784. doi:10.1016/j.cherd.2018.02.032

Raman, R., Grossmann, I.E., 1994. Modelling and computational techniques for logic based integer programming. Comput. Chem. Eng. 18, 563–578. doi:10.1016/0098-1354(93)E0010-7

Rios, L.M., Sahinidis, N. V., 2012. Derivative-free optimization: a review of algorithms and comparison of software implementations. J. Glob. Optim. 56, 1247–1293. doi:10.1007/s10898-012-9951-y

Seader, J.D., J. Henley, E., 2006. Separation Process Principles, 2nd Editio. ed. John Wiley & Sons, Inc.

Smith, R., 2005. Chemical Process Design and Integration. John Wiley & Sons Ltd.

The MathWorks - Inc.,A. Genetic Algorithm - R2021b. URL https://es.mathworks.com/help/gads/genetic-algorithm.html (accessed 12.23.21).

The MathWorks - Inc.,B. Nonlinear Constraint Solver Algorithms. URL https://es.mathworks.com/help/gads/description-of-the-nonlinear-constraint-solver.html (accessed 12.23.21).

Thebelt, A., Kronqvist, J., Mistry, M., Lee, R.M., Sudermann-Merx, N., Misener, R., 2020. ENTMOOT: A Framework for Optimization over Ensemble Tree Models. doi:10.1016/j.compchemeng.2021.107343

Trespalacios, F., Grossmann, I.E., 2014. Review of Mixed-Integer Nonlinear and Generalized Disjunctive Programming Methods. Chemie Ing. Tech. 86, 991–1012. doi:10.1002/cite.201400037

Türkay, M., Grossmann, I.E., 1996. Logic-based MINLP algorithms for the optimal synthesis of process networks. Comput. Chem. Eng. 20, 959–978. doi:10.1016/0098-1354(95)00219-7

Turton, R., Baile, R.C., Whiting, W.B., 2008. Analysis, synthesis and design ofchemical processes. Pearson Education.

U. M. Diwekar, Grossmann, I.E., and E. S. Rubinf, 1992. An MINLP Process Synthesizer for a Sequential Modular Simulator. Ind. Eng. Chem. Res. 31, 313–322.

Vazquez–Castillo, J.A., Venegas–Sánchez, J.A., Segovia–Hernández, J.G., Hernández-Escoto, H., Hernández, S., Gutiérrez–Antonio, C., Briones–Ramírez, A., 2009. Design and optimization, using genetic algorithms, of intensified distillation systems for a class of quaternary mixtures. Comput. Chem. Eng. 33, 1841–1850. doi:10.1016/j.compchemeng.2009.04.011

Viswanathan, J., Grossmann, I.E., 1990. A combined penalty function and outer-approximation method for MINLP optimization. Comput. Chem. Eng. 14, 769–782. doi:10.1016/0098-1354(90)87085-4

Westerlund, T., Pettersson, F., 1995. An extended cutting plane method for solving convex MINLP problems. Comput. Chem. Eng. 19, 131–136. doi:10.1016/0098-1354(95)87027-X

Yeomans, H., Grossmann, I.E., 2000. Disjunctive programming models for the optimal design of distillation columns and separation sequences. Ind. Eng. Chem. Res. 39, 1637–1648.

Yeomans, H., Grossmann, I.E., 1999. A systematic modeling framework of superstructure optimization in process synthesis. Comput. Chem. Eng. 23. doi:10.1016/S0098-1354(99)00003-4