Rigorous Design of Distillation Columns: Integration of Disjunctive Programming and Process Simulators

José A. Caballero,* Daniel Milán-Yañez, and Ignacio E. Grossmann†

Department of Chemical Engineering, University of Alicante, Apartado de Correos 99. E-03080 Alicante, Spain, and Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

The economic optimization of a distillation column involves the selection of the number of trays and the feed- and side-streams locations, as well as the operating conditions that minimize the total investment and operation cost. In this paper, we present a superstructure-based optimization algorithm that combines the capabilities of commercial process simulators—taking advantage of the tailored algorithms designed for distillation and property estimation implemented in these simulators—and generalized disjunctive programming (GDP). The proposed algorithm iterates between two types of subproblems: A nonlinear programming (NLP) subproblem, in which the trays are divided into existing and nonexisting (nonexisting trays behave like simple bypasses without mass or heat exchange through the use of Murphree efficiencies), and a specially tailored master mixed integer linear programming (MILP) problem. The NLP subproblems are solved by integrating the process simulator with an external NLP solver. Several examples are presented which show promising results.

1. Introduction

Distillation is the most commonly used separation technique in chemical processes and it seems this will remain true for many years, despite the advances in alternative separation technologies (e.g., membranes and PSA). Therefore, it is not surprising that distillation continues to be one of the most studied unit operations. This research effort is justified in light of progress for improving distillation capabilities such as divided wall columns,[1] thermally coupled distillation systems,[2−9] reactive distillation,[10,11] membrane distillation,[12] etc.

As a consequence of the important research efforts in distillation, the simulation of these systems has reached a significant degree of maturity and a good number of robust and reliable numerical methods have been developed for distillation. For instance, some of the most important methods are the following:[13] The bubble point methods (BP), the sum rates models (SR), the 2N Newton methods, or the global Newton or simultaneous correction methods (SC), such as the Naphtali−Sandholm,[14] Goldstein−Stanfield,[15] and Ishii−Otto,[16] or the widespread inside-out methods.[17,18] Additional classes of rigorous computational methods include the relaxation methods by Rose, Sweeny, and Schrodt[19] or the Homtopy continuation methods.[20] Most of these methods are nowadays implemented in commercial process simulators. However, the optimal economic synthesis of distillation columns continues to be a challenging problem. The reason is that the designer has to deal not only with continuous variables (pressure, reflux ratio, and boilup ratio) but with discrete decisions (i.e., number of trays and feed tray location). The recent trends address models of increasing complexity through the use of mathematical programming. The high degree of nonlinearity and the difficulty of solving the corresponding optimization models, however, have prevented methods with rigorous models from becoming tools that can be widely used, except by a specialized community.[21]

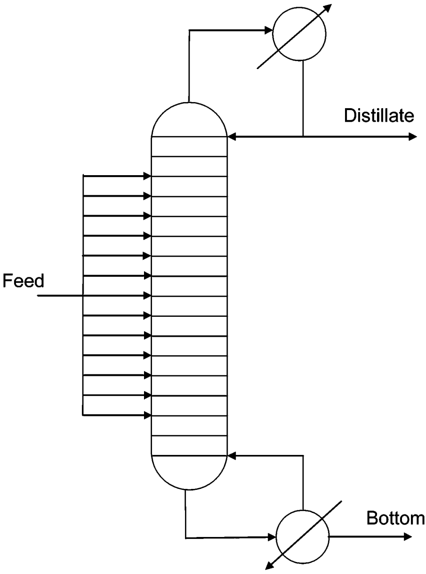

Figure 1. Superstructure of Sargent and Gaminibandara[22] for the optimal location of the feed tray together with the optimal operational conditions.

The first successful approach using mathematical programming techniques applied to distillation was attributable to Sargent and Gaminibandara.[22] It corresponds to the simplest distillation design in which the number of trays is fixed, and the goal is to locate the optimal position of the feed tray together with the optimal operational conditions. Figure 1 shows the postulated superstructure for that problem. The basic idea is to consider the feed to be sent to any tray, except for the reboiler and the condenser, by splitting it into as many streams as trays in the column. The model is formulated as a nonlinear programming (NLP) problem,

* To whom all correspondence should be addressed. Tel.: + 34 965903400 ext 2322. Fax: +34 965903826. E-mail: caballer@ua.es.

† Carnegie Mellon University.

10.1021/ie050080l CCC: $30.25 © 2005 American Chemical Society
Published on Web 07/16/2005

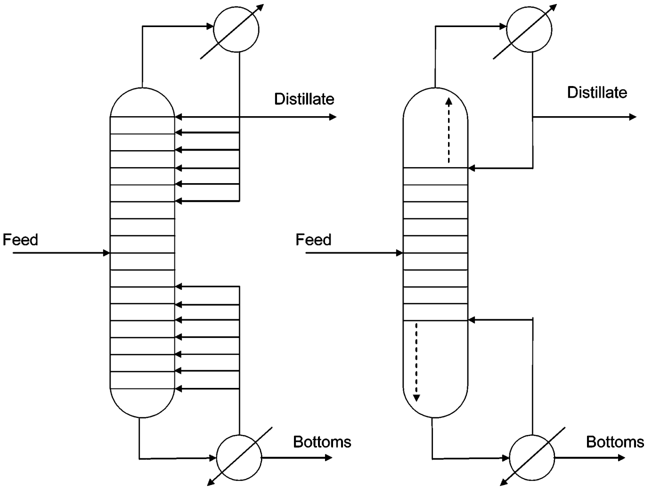

Figure 2. Superstructure of Viswanathan and Grossmann[24] for the optimal location of feed, total number of trays, and optimal operational conditions.

which is equivalent to a relaxed mixed integer nonlinear programming (MINLP) problem.[23]

If the objective is to simultaneously determine the total number of trays and the feed location, the complexity of the problem greatly increases. The first successful approach was attributable to Viswanathan and Grossmann[24] (see Figure 2). These authors proposed a superstructure in which the feed location is fixed, but the total number of trays above or below the feed is not. An upper bound to the number of total trays above or below the pinch is postulated; the reflux is then returned to all trays above the feed, and the reboil is returned to all trays below the feed. Binary variables are assigned to the existence of each of the reflux and reboil returns. The problem is formulated as an MINLP in which the constraints include the mass and enthalpy balances, the phase equilibrium equations, and the summation of the mole fraction equations (MESH) together with the mixed integer relations to ensure the correct return of reflux and reboil streams.

A major difficulty in the model of Viswanathan and Grossmann is related to the nonexisting trays. In these trays, there is a zero liquid flow (rectifying section) or a zero vapor flow (stripping section), which can produce numerical problems due to the convergence of equilibrium equations with a zero value in the flow of one of the phases. In other words, the vapor−liquid equations must be satisfied in trays where no mass transfer takes place.

Despite the increase of the computational time of the model and convergence problems, the model of Viswanathan and Grossmann has been successfully applied by different research groups. For example, Ciric and Gu[25] used the MINLP approach for the synthesis of ethylene glycol via ethylene oxide in a kinetic controlled reactive distillation column. Bauer and Stichlmair[26] applied the MINLP approach to the synthesis of sequences of azeotropic columns. Dünnebier and Pantelides[27] used the model to generate sequences of thermally coupled distillation columns.

The introduction of generalized disjunctive programming (GDP)[28] allowed researchers to overcome some of the difficulties of the MINLP models. Yeomans and Grossmann[29] proposed a GDP model in which the nonexisting trays are considered as simple bypasses of vapor and liquid flows without mass transfer; in other words, it is similar to the vapor and liquid flowing by the two sides of an imaginary wall without mass or heat transfer between them. Mass and energy balances are trivially satisfied. The only difference with existing trays is in the application of the equilibrium equations. Figure 3 shows a scheme of the disjunctive model with conditional trays. An extension of the GDP model for the case of reactive distillation columns was proposed by Jackson and Grossmann.[30]

Barttfeld et al.[31] showed that multiple representations and models are possible for the optimization of single distillation columns. Figure 4 shows some possible representations to determine the optimal feed tray and the number of trays with a MINLP formulation. A reboiler (Figure 4a) or a condenser (Figure 4b) is placed in all candidate trays and the feed is split into a set of streams in order to determine its optimal location. In Figure 4c, both a reboiler and a condenser are placed in all candidate trays. While in the original model of Viswanathan and Grossmann[24] (Figure 2) the problem consists of finding the location of the reflux (reboil) streams, in the representation of Figure 4, the problem consists of finding the optimal location for the energy exchanged. This last approach allows heat to be exchanged at intermediate tray temperatures with the inherent possibility of obtaining more energy-efficient designs.

Barttfeld et al.[31] also studied different representations for the optimization of a single distillation column when it is solved with a GDP formulation. Figure 5 shows the alternatives considered. However, in this case, the computational results showed that the most effective alternative was the original configuration presented by Yeomans and Grossmann.[29]

As mentioned above, GDP formulations provide better numerical behavior than MINLP, but because of the nonlinearities and nonconvexities inherent to the distillation models, both MINLP and GDP formulations require good initial values and bounds to converge. Barttfeld and Aguirre[32] proposed an initialization procedure based on reversible distillation. Barttfeld et al.[31] have shown that convergence is greatly enhanced by including a preprocessing step.

Although very important improvements have been carried out in these last years for the robust optimal design of distillation columns, the models lack flexibility as compared with process simulation systems. Even with the important numerical improvements introduced with GDP, these models continue to rely on generic solvers and do not take advantage of specific numerical algorithms for distillation processes.

One interesting approach that makes use of available process simulators and optimization tools is the one by Lang and Biegler.[33] These authors proposed a distributed stream-tray optimization method (DSTO) in which the reflux and the feed flow rates can be distributed and directed to a set of candidate trays according to a differentiable distribution function (DDF). Using this DDF, the location of the feed, reflux, and other side streams can be treated as continuous instead of integer variables. The drawback, however, is that the DDF function is highly nonconvex, and the method can easily be trapped in local solutions.

On the other hand, the use of process simulators in a modular environment for solving MINLP has been addressed by Diwekar et al.,[34] Reneaume et al.,[35] and Díaz and Bandoni.[36] All these works are based on the

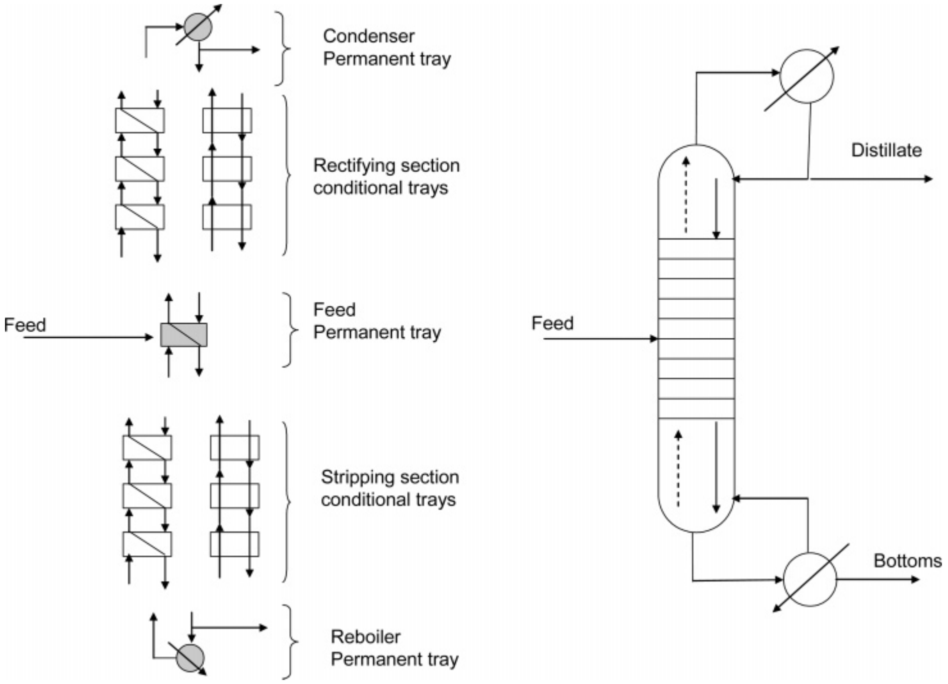

Figure 3. Superstructure of Yeomans and Grossmann[29] used as the basis for the generalized disjunctive programming formulation.

augmented penalty/equality relaxation outer-approximation algorithm.[23] Kravanja and Grossmann[37] followed a similar approach, adapting the modeling/decomposition (M/D) developed by Kocis and Grossmann[38] that can be considered a precursor of generalized disjunctive programming.

In this paper, we propose an algorithm for designing distillation columns that integrates a process simulator in a generalized disjunctive programming formulation. All numerical aspects related to the convergence of a distillation column, the selection of thermodynamic models, the transport properties, etc. are specified in the simulation environment. An external optimizer is connected to the simulator in order to solve the NLP subproblems. Derivatives for the design variables were obtained by perturbations (forward differences), and a "specially tailored" master problem that takes the form of a mixed integer linear programming (MILP) problem is solved. Everything is controlled by an external executive program. The next sections provide details of the algorithm and are followed by several numerical examples.

2. GDP Algorithm for the Rigorous Design of Distillation Columns

For the sake of simplicity, but without loss of generality, let us consider a conventional column with one feed and two products (distillate and bottom streams). Later, we will extend the algorithm to complex columns. Therefore, the problem we are dealing with can be stated as follows: given an *N*-component mixture, determine the optimal configuration (feed location and total number of trays) and the optimal operating conditions (i.e., reflux ratio, heat load, etc.) for separating the mixture into two streams within given specifications using a conventional distillation column.

The superstructure used in this work for the column design is based on the superstructure proposed by Yeomans and Grossmann[29] (Figures 3 and 5a), which, according to Barttfeld et al.,[31] is the superstructure that has shown better computational results when solved with a GDP formulation. This GDP formulation can be adapted to work in conjunction with a process simulator.

The method is based on the outer-approximation algorithm,[23] in which the master problem is modified to deal with the special characteristics of distillation columns in a modular environment. A scheme of the algorithm is shown in Figure 6. As can be seen, it iterates between NLPs that correspond to configurations with fixed numbers of existing trays and a master (MILP) problem specially developed for distillation columns. NLPs provide upper bounds to the optimal solution, while the master problems predict a new configuration (number of existing trays and position of the feed tray) as well as continuous variables to be solved by the next NLP. The main steps of the algorithm are explained in the next sections.

2.1. Initialization: Parameter Setting for the Column Optimization. In a simulator working in a modular environment, it is possible to generate, in principle, any of the previously described superstructures, for example, sequentially connecting flash units. But this approach is neither practical nor convenient because we then lose all the advantages of the specially suited numerical methods developed for distillation columns, introducing a large number of recycles with corresponding tear streams and the consequent numerical difficulties that often prevent convergence. Therefore, it is much better to use one of the built-in columns in the process simulator.

All the required parameters are initialized in the simulator environment (either directly or through an external executive program): selection of thermodynamic model(s), feed specification, etc. Initially, the total number of trays in both the rectifying and stripping sections must be selected to be large enough so as to provide an upper bound to the optimal number of trays.

2.2. Solving the Nonlinear Programming (NLP) Problem. Once all the parameters of the column have been specified, a first NLP must be solved. Instead of solving a relaxed NLP in which a given tray could exist

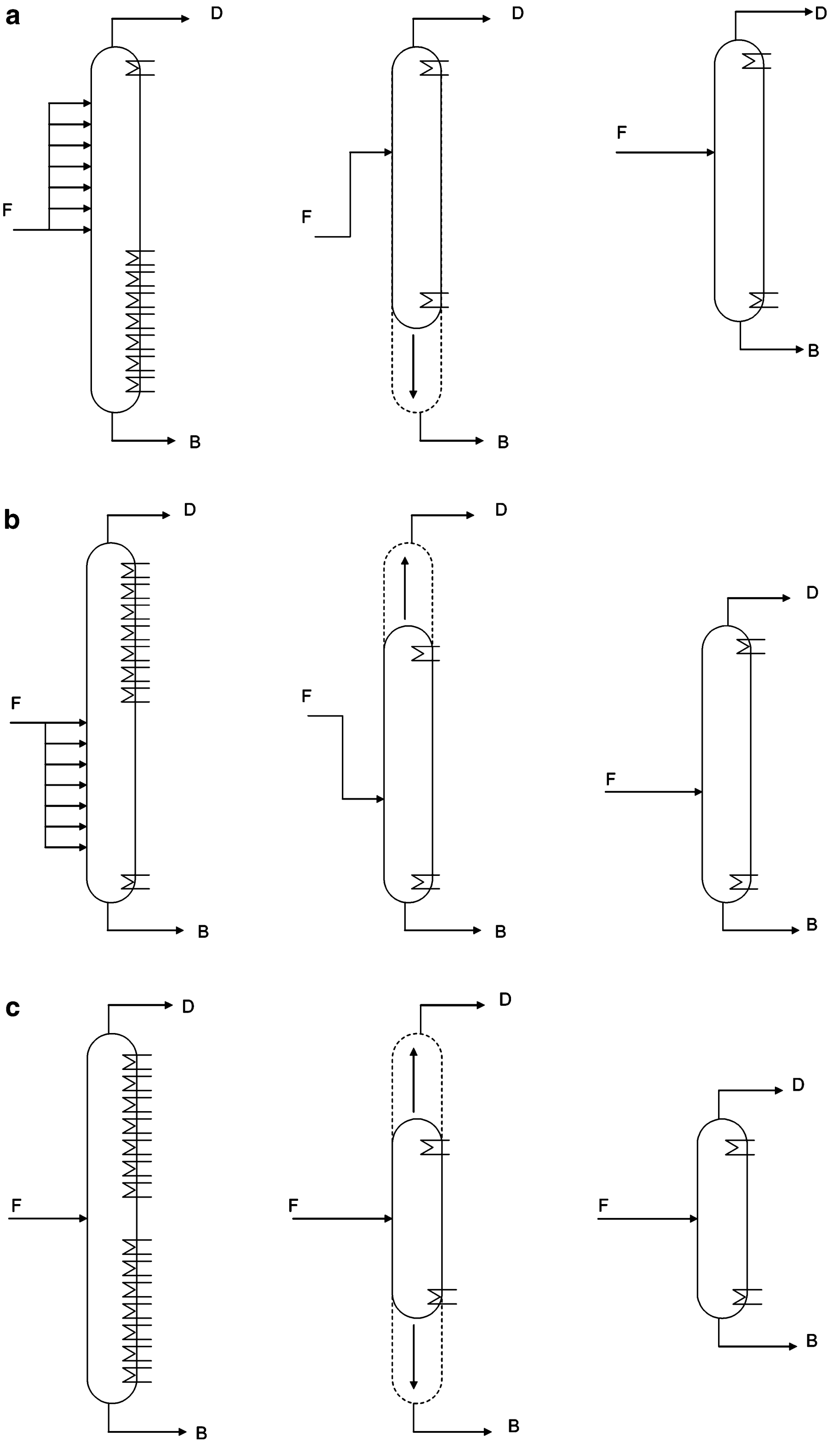

Figure 4. Alternative representations to determine the optimal feed tray and the number of trays with a MINLP formulation. (a) A reboiler or (b) a condenser are placed in all candidate trays and the feed is split into a set of streams in order to determine its optimal location. (c) Both a reboiler and a condenser are placed in all candidate trays.

or not, a feasible configuration with a fixed number of trays is selected. There are at least two ways of generating this configuration. The first one consists of "redefining" the number of trays in the column. However, usually this means that we are starting the problem again, all the values from previous iterations (if any) are lost, and, consequently, the convergence is slower. The simpler and most direct approach consists of forcing a distillation tray to behave as a simple bypass of liquid and vapor flows, without mass or heat transfer. Fortunately, in a process simulator, this can be done by simply fixing the Murphree efficiency, eq 1, to zero for the nonexisting trays.

$$\eta_{\mathrm{M}} = \frac{y_n - y_{n-1}}{y_n^* - y_{n-1}} \tag{1}$$

In eq 1, y_n and y_{n-1} are the vapor mole fractions of the

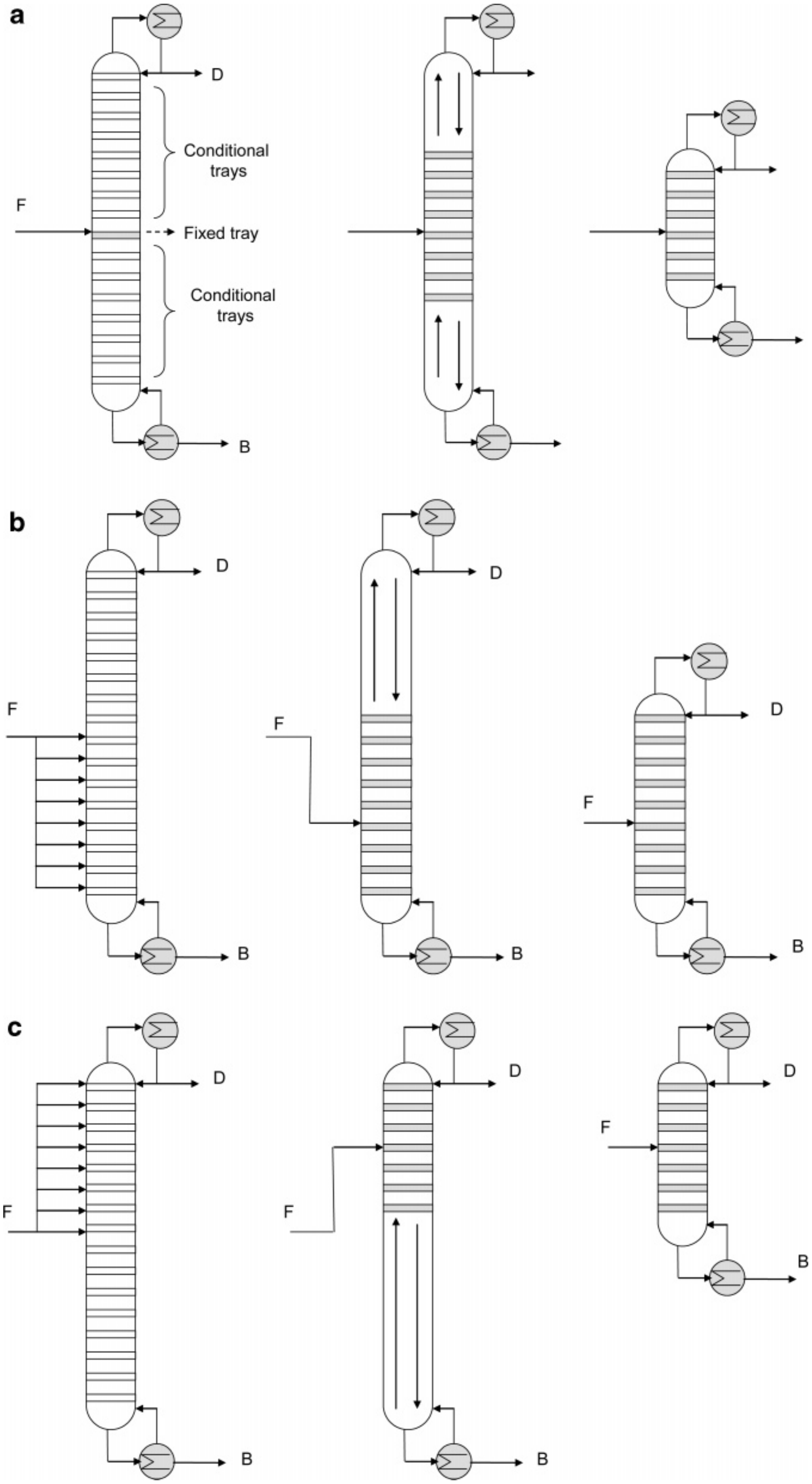

Figure 5. Different representation for the optimization of a single distillation column when it is solved with a GDP formulation.

vapor streams leaving tray n and $n-1$, respectively, and y_n^* is the mole fraction of vapor in equilibrium with the liquid leaving stage n. Note that equilibrium equations trivially hold in those trays in which the efficiency is set to zero.

To avoid alternate equivalent solutions, the existing trays should be consecutive, for example, as shown in Figure 3. In this work, we will follow the following criteria. In the rectifying section, if a given tray exists, then all trays below it, until the feed tray, must exist (see Figure 5a). In the stripping section, if a given tray exists, then all the trays above it must exist. Initially, a good option consists of solving the initial NLP assuming that all trays exist. The result of this first NLP is an upper bound to the optimal solution of the problem.

If the process simulator has optimization capabilities (including the possibility of handling extra constraints), then it can be used to solve the NLPs. However we have

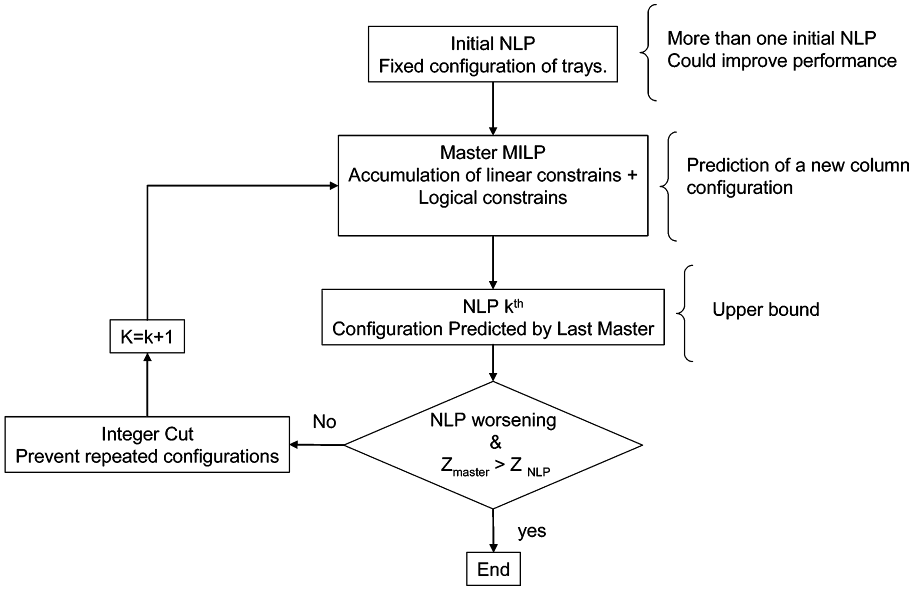
Initial NLP
Fixed configuration of trays.
More than one initial NLP
Could improve performance
Master MILP
Accumulation of linear constrains +
Logical constrains
Prediction of a new column
configuration
NLP kth
Configuration Predicted by Last Master
Upper bound
K=k+1
Integer Cut
Prevent repeated configurations
No
NLP worsening
&
Z$_{master}$ > Z $_{NLP}$
yes
End

Figure 6. Scheme of the proposed algorithm.

obtained better results using an external optimizer. Most specifically, the idea is as follows.

The NLP to be solved can be written as follows:

$$\min_{x_D} z = f(x,u,x_D)$$

$$s.a.\ h_I(x,u,x_D) = 0$$

$$\left.\begin{array}{l} g_E(x,u,x_D) \le 0 \\ h_E(x,u,x_D) = 0 \end{array}\right\} \qquad \text{(P1)}$$

In problem P1, x_D corresponds to the design (independent) variables, which are the degrees of freedom of the problem. x corresponds to all the other variables of the process that are calculated by the simulator and can only be read. u corresponds to a set of fixed parameters that are not modified during the calculation. This set of variables includes all the discrete decisions which, in the distillation case, are the number of existing trays in the rectifying and stripping sections. The equations h_I are all the equations solved by the process simulator that, in general, cannot be accessed by the user. These equations include mass and energy balances, property estimations, equilibrium equations, etc. Equations g_E and h_E are explicit external constraints added to the problem. Here, we can add almost any constraint we are interested in: purity or recovery constraints, temperature constraints for security or stability reasons, etc. Finally, f is a scalar objective function, for example, total annualized cost.

The external solver was developed in Matlab.[39] Results provided in examples were obtained with the "fmincon" solver of Matlab and connected to the process simulator, HYSYS,[40] through a client−server ActiveX automation technology. The fmincon solver is an SQP (successive quadratic programming) based algorithm; the estimation of the Hessian of the Lagrangian is updated at each iteration using the BFGS formula. All derivatives were calculated by finite forward differences.

A very important point in the method is that the process simulator must converge each time the solver sends a set of values of design variables. If the process simulator does not converge, the complete procedure fails. Therefore, it is important, in order to avoid infeasible solutions, to use a set of independent variables that allow the process simulator to converge from almost any initial point. In other words, it is important that the set of implicit equations $h_I(x,u,x_D) = 0$ be solved as easily as possible because we do not have control of how these equations are solved. Difficult specifications should be left to the external constraints.

It is convenient to modify problem P1 to properly handle infeasible solutions. Adding slack variables and an exact penalty to the objective function ensures that infeasible solutions are not due to external constraints

$$\min_{x_D} z = f(x,u,x_D) + \mathbf{\Pi}^T\,(\mathbf{s} + \mathbf{s}_2 + \mathbf{s}_2)$$

$$\text{s.a. } h_I(x,u,x_D) = 0$$

$$\left.\begin{array}{l} g_E(x,u,x_D) \le \mathbf{s} \\ h_E(x,u,x_D) + \mathbf{s}_1 - \mathbf{s}_2 = 0 \end{array}\right\}$$

$$\mathbf{s} \ge 0; \quad \mathbf{s}_1 \ge 0; \quad \mathbf{s}_2 \ge 0 \qquad \text{(F-P1)}$$

where $\mathbf{\Pi}$ is a penalty parameter vector whose value is finite but chosen to be sufficiently large and $\mathbf{s}$, $\mathbf{s}_1$, and $\mathbf{s}_2$ are vectors of positive slack variables.

It is worth remarking that the entire problem is posed in terms of design variables, and therefore, all gradients are written in terms of these variables; explicitly, we only have access to the Jacobian of the external constraints. The rest of the variables and constraints are implicit. The situation is similar when using a reduced gradient algorithm, but in this case, we do not have access to the gradients of basic variables, only to the superbasic (design) variables.

2.3. The Master Problem. When we solve an MINLP or GDP problem by a decomposition method (see, for example, ref 28), the objective of the master problem is to provide a new set of values of binary (or Boolean) variables that are likely to produce better results than the values tested in the previous iterations. In convex problems, the solution of the master problem provides a sequence of nondecreasing lower bounds. The iterations finish when the lower and upper bound lie within a specified tolerance. In nonconvex problems, however,

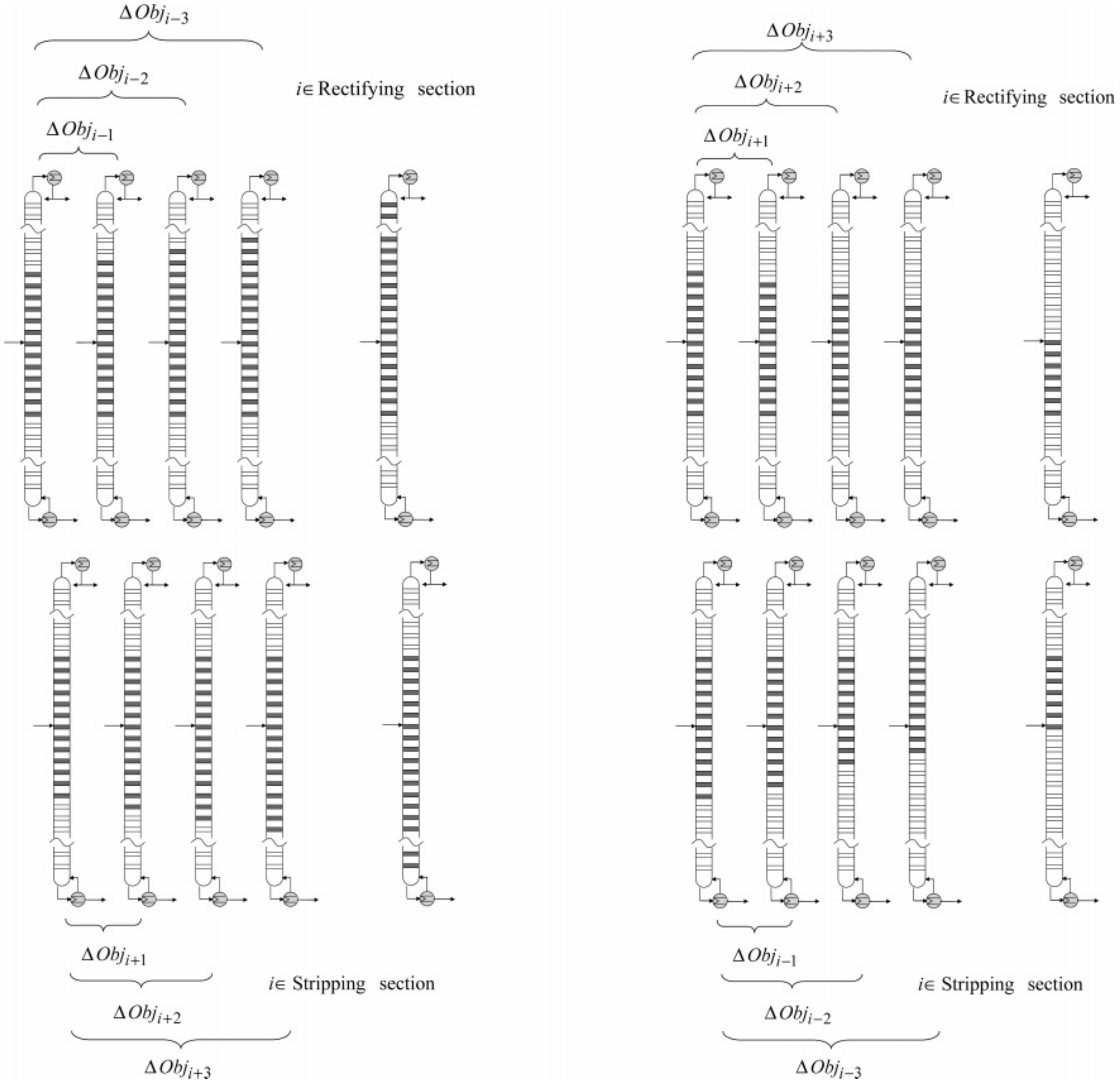

Figure 7. Graphical representation of the steps followed to generate the master problem.

the lower bounding property cannot be assured, and different heuristic modifications are introduced in the master problem in order to minimize the effects of nonconvexities. One widely used modification is the outer-approximation−equality relaxation−augmented penalty algorithm originally developed by Viswanathan and Grossmann.[23]

Here, we present a specially tailored heuristic−master problem for the design of distillation processes. Although it cannot guarantee a rigorous lower bound, numerical experiments have shown that, in most of the cases, this property holds true; as will be showed in the examples section, promising results were obtained.

Assume that we are at the iteration k, and we have solved the corresponding NLP problem. We define the following sets:

T = {i | i is a tray in the distillation column}

$\mathrm{TR_T}$ = {r | r is a tray in the rectifying section of the column}

$\mathrm{TS_T}$ = {s | s is a tray in the stripping section of the column}

EQ = {j | j is an external (explicit) equality constraint}

IEQ = {j | j is an external (explicit) inequality constraint}

D = {n | n is a design (independent) variable}

To generate the master problem, the design variables x_{D} are fixed to the optimal value obtained in the previous NLP^k problem, and a series of simulation problems are solved (note that these are not optimization problems). First, a problem adding one more tray to the rectifying section is solved, then another with two more trays, then three trays, etc., until all the trays in the rectifying section exist. The procedure is repeated, starting again from the configuration of iteration k, but now removing one by one the trays in the rectifying section. The same calculations are performed in the stripping section. Figure 7 illustrates the procedure. Note that all these simulations can be performed very efficiently because the starting point is a converged configuration that only differs in one tray from the previous one. Since initial values are very good approximations of the solution, the calculation time is usually only a small fraction of the time consumed in an ordinary simulation (and similar to the time consumed in calculating a derivative by perturbations).

Considering the previous paragraph we define the following parameters:

$\Delta\mathrm{obj}_i^k$ = Difference between the objective function calculated when the tray i exists in the column and the objective function of the original NLP^k problem. Remember that if the tray i is in the rectifying (stripping) section, all the trays below (above) it in this section must exist.

$\Delta g_{i,j}^{k}(x,u,x_{\mathrm{D}}^{k})$ = Difference between the values of the inequality constraint j, when the tray i exists, and the constraint j in the original NLP^k problem.

$\Delta h\mathrm{E}_{i,j}^{k}(x,u,x_{\mathrm{D}}^{k})$ = Difference between the value of the external equality constraint j, when the tray i exists, and the equality constraint j in the original NLP^k problem.

The master problem can be written as follows.

M-P1:

$$\min:\ \alpha + \pi\Big(\sum_{j\in\mathrm{IEQ}} \mathrm{slack1}_j + \sum_{j\in\mathrm{EQ}} \mathrm{slack2}_j\Big)$$

$$\text{s.t. } f(x^k,u^k,x_{\mathrm{D}}^k) + \sum_n \left(\frac{\partial f}{\partial x_{\mathrm{D}_n}}\right)_{x_{\mathrm{D}}=x_{\mathrm{D}^i}} (x_{\mathrm{D}_n} - x_{\mathrm{D}}{}^{k}) + \sum_{i\in T} W_i\ \Delta\mathrm{obj}_i^k \le \alpha$$

$$g_j(x^k,u^k,x_{\mathrm{D}}^k) + \sum_n \left(\frac{\partial g_j}{\partial x_{\mathrm{D}_n}}\right)_{x_{\mathrm{D}}=x_{\mathrm{D}^i}} (x_{\mathrm{D}_n} - x_{\mathrm{D}}^{k}) + \sum_{i\in\mathrm{T}} W_i \Delta g_{i,j}^k \le \mathrm{slack1}_j\ \forall\, j\in\mathrm{IEQ}$$

$$\mathrm{sign}(\lambda_j^k)\left[h_{\mathrm{E}_j}(x^k,u^k,x_{\mathrm{D}}^k) + \sum_n \left(\frac{\partial h_{\mathrm{E}_j}}{\partial x_{\mathrm{D}_n}}\right)_{x_{\mathrm{D}}=x_{\mathrm{D}^i}} (x_{\mathrm{D}_n} - x_{\mathrm{D}}^{k}) + \sum_{i\in\mathrm{T}} W_i \Delta h_{\mathrm{E}i,j}^k\right] \le \mathrm{slack2}_j\ \forall\, j\in\mathrm{EQ} \qquad k = 1, 2, 3, ..., K$$

$$\begin{bmatrix} \mathrm{slack1}_j \ge 0 & \mathrm{slack2}_j \ge 0 \\ \sum_{i\in\mathrm{T}R_r} W_i = 1 & \sum_{i\in\mathrm{T}S_r} W_i = 1 \\ W_i\in\{0, 1\} & \end{bmatrix}$$

In problem M-P1, α is an auxiliary variable used to place the objective function as a constraint and slack1 and slack2 are positive slack variables to ensure feasibility. The objective function is formed by the auxiliary variable α and a penalty for constraint violation in terms of the penalty parameter π that multiplies the slack variables.

The left-hand side of the first constraint is formed by two parts. The first is a linearization of the objective function with respect to the design (independent) variables, such as in the outer-approximation (OA) algorithm (Duran and Grossmann[41]). The second part is the contribution of the existence (or nonexistence) of the tray "i" with respect to the configuration in iteration k. This approach deserves some detailed comments. In the outer-approximation algorithm, the master problem is generated by linearizations with respect to continuous and binary variables. In this case, the situation is more complicated. Binary variables are not explicitly included in the formulation. Instead, the nonexistence of a tray is forced by setting the tray efficiency to zero, which becomes the tray in a simple liquid and vapor bypass. It would be possible to include linearizations with respect to the efficiencies and then either force the efficiency to be a binary variable or introduce an external binary variable that made the efficiency be zero if a given tray does not exist (in this way, we could introduce the efficiencies as optimization variables, if desired). However, these approaches produced poor numerical results. In most of the cases, the low sensitivity prevents accuracy for the derivatives. There is another problem with that approach. Consider, for example, a configuration with N trays in the rectifying section. Since all the trays above the N tray are equivalent, there are no differences in derivatives (or any other properties) between one tray and another . Therefore, the prediction of the master problem is usually very poor. To introduce the effect of adding or removing trays from the column, we substituted the linearizations with respect to binary variables (or efficiencies) by the extra contribution of these trays (of course, keeping constant all the design variables) with respect to the original NLP problem.

The same procedure is followed for the external equality and inequality constraints. In this case, deviations from the values of $g_{\mathrm{E}j}$ and $h_{\mathrm{E}j}$ when the tray i exists, with respect to their value in the optimal solution of the NLP^k problem, are included. In the equality constraints, the parameter sign(λ_j^k) makes reference to the sign of the Lagrange multiplier of equality constraint j in the last NLP. This parameter is necessary in order to correctly relax the equalities into inequalities.[38]

W_i is a binary variable that, in the rectifying section, takes the value of 1 for the highest existing tray (all trays above it do not exist, and all trays below it exist) and, in the stripping section, takes the value of 1 for the lowest existing tray (all trays below it do not exist, and all trays above it exist). The last constraints explicitly state that, in each section, only one of those binary variables can take the value 1. This approach is slightly different from the conventional approach, in which a binary variable is assigned to each tray in the column. In this case, the parameters $\Delta\mathrm{obj}_i^k$, $\Delta g_{i,j}^k(x,u,x_{\mathrm{D}}^k)$, and $\Delta h\mathrm{E}_{i,j}^k(x,u,x_{\mathrm{D}}^k)$ include the cumulative effect of extra trays (or defects of trays) in each column section on the objective and on the inequality and equality constraints, respectively. Therefore, we only need to determine which is the highest existing tray in the rectifying section (or which is the lowest existing tray in the rectifying section).

It is worth mentioning that, in problem M-P1, all the linear constraints are accumulated. In other words, at iteration k, the problem is formed by the constraints generated at the kth iteration plus all the constraints of all previous iterations. To avoid repeated solutions, a binary cut[38] is added at each iteration.

Finally, a remark on implicit constraints: these are sets of equations solved by the process simulator and are not explicitly included in the master problem, but they are implicitly included in the linearizations of the objective function and explicit constraints, as well as in the extra contributions of additional trays. Assume, for example, that we calculate the derivatives with respect to $x_{\mathrm{D}j}$ by forward finite differences. We have to solve two simulation problems: one for x_{D} and another for $x_{\mathrm{D}} + h$ (h is a small perturbation). However, the values of the dependent variables x are not the same in both cases. Calling $x1$ the solution of the implicit equations before perturbation and $x2$ the solution after perturbation, the derivatives of the objective function

Table 1. Data for the Examples

Heat Exchangers			
kettle reboiler: $U = 820$ W/(m^2 K)			
condenser: $U = 800$ W/(m^2 K)			
Utilities			
high-pressure steam	(254 °C)	5.09 \$/GJ	
medium-pressure steam	(184 °C)	3.66 \$/GJ	
low-pressure steam	(160 °C)	3.17\$/GJ	
low-pressure steam	(120 °C)	1.30\$/GJ	
cold water	(30 °C)	0.19\$/GJ	
calculated based on 8000 h/year of operation			
Columns			
calculation based on sieve tray (except second column in example 4; packed column)			
stainless steel			
tray separation 0.609 m			
design limit flooding (85%)			
Example 1		Example 2	
feed	150 kmol/h (bubble point)	feed	100 kmol/h (bubble point)
composition (mol fraction)		composition (mol fraction)	
pentane	0.2	2-propanol	0.3
hexane	0.2	1-propanol	0.2
heptane	0.6	*i*-butanol	0.4
pressure	100 kPa	1-butanol	0.1
thermod	Peng Robinson, HYSYS defaults	pressure	30 kPa
specifications		thermod	NRTL, HYSYS defaults
pentane recovery	≥ 98%	specifications	
pentane molar fraction in distillate	≥ 0.98	molar fraction C_3OH distillate	≥ 0.98
		molar fraction C_4OH bottoms	≥ 0.98
Example 3		Example 4	
feed	150 kmol/h (bubble point)	feed	100 kmol/h (bubble point)
composition (mol fraction)		composition (mol fraction)	
acetone	0.15	ethanol	0.85
methanol	0.55	water	0.15
water	0.30	pressure	101.3 kPa
pressure	101.3 kPa	thermod	NRTL, HYSYS defaults
thermod	NRTL, HYSYS defaults	specifications	
specifications		molar fraction ethanol	≥ 0.999
recovery methanol	≥ 99%	molar fraction water	≥ 0.999
recovery water	≥ 99%	molar fraction ethylene glycol	≥ 0.999

are approximated by

$$\frac{\partial f}{\partial x_{\mathrm{D}j}} \approx \frac{f(x_2,u,x_{\mathrm{D}} + h) - f(x_1,u,x_{\mathrm{D}})}{h} \tag{2}$$

Similar approximations are used for the rest of the constraints. If we have access to analytical (or numerical) derivatives with respect to the implicit variables, then we can directly calculate the reduced gradient. In fact, we are working in the projection space of the independent variables.

The optimization process is controlled from an executive program, in this case developed under Matlab. Once all the initial parameters have been set in the process simulator HYSYS, which is done directly from Matlab through a client−server application, the NLP solver is executed. Then Matlab calculates all derivatives and necessary data for the master problem using HYSYS to perform all extra necessary simulations. The master problem was solved with GAMS-CPLEX, and communications between GAMS and Matlab were carried out using text files.

3. Examples

3.1. Example 1. In this first example, the objective is to recover pentane from a mixture formed by pentane−hexane−heptane. This example (example 1a) has been included to provide a problem that is easy to reproduce and in which equations for calculating rigorous costs do not hide the essence of the algorithm. A very simple objective function is first considered. The problem will also be solved using a rigorous total annualized cost calculation (example 1b).

The data for the problem are shown in Table 1. A configuration with a maximum of 40 trays is initially considered (condenser and reboiler are not included). The feed tray is placed in tray 20 and the column is numbered from top to bottom. For illustration purposes, the problem is first solved using a simplified objective function

$$\min Z = 5Q_{\mathrm{reb}} + Q_{\mathrm{cond}} + 30(\text{total number of trays}) \tag{2}$$

where Q_{reb} and Q_{cond} are the heat loads in the condenser and the reboiler (kW), respectively. The feed is completely specified. Therefore, in the NLPs, there are only two degrees of freedom (pressure was also fixed). We have chosen as the design (independent) variables the reflux ratio (R) and the molar ratio of distillate to feed (D/F). The external constraints that were specified are that the pentane molar fraction in the distillate must be at least 0.98 and that the recovery of pentane in the distillate must be at least 98%. These constraints can also be treated as specifications in the simulator. However, in our experience, convergence is then much more difficult and the optimization usually fails. Then, the molar fraction of the distillate and the recovery must be specified as external constraints. Therefore, in this

Table 2. Major Iterations in Example 1a[a]

	master	NLP
Initial NLP		existing trays from 1 to 40 (feed tray 20) $R = 1.916$ $D/F = 0.2000$ objective = 5452.9
Iteration 1	trays from 12 to 30 $R = 2.0399$ $D/F = 0.2009$ objective = 4995.4	$R = 2.1165$ $D/F = 0.2000$ objective = 5071.1
Iteration 2	trays from 12 to 31 $R = 2.0205$ $D/F = 0.2007$ objective = 5093	$R = 2.0699$ $D/F = 0.2000$ objective = 5040
Iteration 3	trays from 12 to 32 $R = 2.0386$ $D/F = 0.1996$ objective = 5119	$R = 2.0390$ $D/F = 0.2000$ objective = 5030.3
Iteration 4	trays from 11 to 32 $R = 2.0209$ $D/F = 0.1997$ objective = 5130	$R = 2.0127$ $D/F = 0.2000$ objective = 5026.3 (optimum)
Iteration 5	trays from 12 to 33 $R = 2.0245$ $D/F = 0.1998$ objective = 5136	$R = 2.0185$ $D/F = 0.2000$ objective = 5033.3

[a] R = reflux ratio. D/F = molar ratio of distillate to feed.

particular example, in each NLP we solve a problem with two degrees of freedom (R and D/F) and two external constraints. All other variables are implicitly calculated by the process simulator.

In this example, we have also specified a minimum number of trays in each section. Although this is not necessary, it helps in the optimization search procedure. For the case where the optimal solution lies at one of these limits, the maximum number of trays is increased or the minimum number of trays is decreased.

We first solve the NLP in which all trays exist. The objective was 5425 units, and the optimal values for design variables were reflux ratio 1.916 and ratio D/F = 0.200. The ratio of distillate to feed could have been fixed a priori, but to illustrate the flexibility of the proposed algorithm, we treated it as a variable. The MILP master problem (M-P1) is then generated and solved. The solution obtained by the master included trays 12−30 and values for reflux ratio (R = 2.04) and distillate-to-feed ratio (D/F = 0.2009) with an objective of 4995.4 units. The optimal solution for the NLP for the configuration predicted by the master (trays 12−30) was R = 2.12, D/F = 0.200, and objective = 5071. Therefore, the values obtained by the master problem are very good initial values to the NLP problem. Table 2 shows the progress of the iterations for this problem. The optimal solution involves 22 trays and the feed is located in tray 10 (see Figure 8a). We considered as stopping criteria when both worsening of the objective function in two consecutive NLPs and crossing of the objective values of NLPs and master problems are simultaneously fulfilled. We should note that, since the lower bounding property of the master does not always hold, this condition cannot be used as the stopping criteria. In some cases, especially in the first iterations, the NLPs do not necessarily decrease monotonically. For example, if we start from a configuration in which all trays exist, it is common that the first master predicts a configuration with a lower number of trays, and it is possible that the objective of the second NLP is worse than the first one.

To verify that the predicted configuration is the global optimum, another 25 NLPs with different configurations around the optimum were solved. In all cases, the objective was worse than the objective of the best configuration that was obtained.

The previous problem was solved again, but now using the total annualized cost as an objective function. The correlations for capital costs estimation (column shell, trays, and heat exchangers) were obtained from Turton et al.[42] The annualized cost to be minimized follows the recommendation of Douglas[43]

$$\mathrm{TAC} = (2.43\mathrm{kcap} + 0.19)\mathrm{Cap} + \mathrm{Cop} \qquad (3)$$

where Cap is the total cost of installed equipment and Cop is the operation (variable) cost. The parameter kcap is a factor that takes into account the value of money with time, typically 1/3−1/4 per year.

The data are shown in Table 1. All other parameters are equal to those in the previous example.

We start by solving the first NLP with all trays. The initial objective was $370 600/year. The first master problem predicts an optimal configuration that involves the minimum number of trays (trays from 15 to 25) with an objective of $210 000/year. Although this result does not always occur, it is not uncommon. If a minimum number of trays in each section is not fixed, infeasible configurations can be generated and a feasibility problem must be solved. Alternatively, another option is to generate the master problem with two NLPs, for example, from configurations with a maximum and minimum number of trays. The remaining iterations are shown in Table 3, and the optimal solution is shown in Figure 8b. Note that the total computational time required was only 76 s. As in the previous case, a series of NLPs changing the number of trays around the best solution obtained showed that the configuration obtained is most likely the global optimum.

For the sake of comparison, because the behavior of the mixture in the operation conditions showed above is near-ideal behavior, it is possible to perform shortcut calculations. We have used the shortcut column implemented in HYSYS with the following specifications: molar fraction of light key (pentane) in bottoms (0.005) and of heavy key (hexane) in distillate (0.02) and reflux ratio 2.45. These values are those obtained in the rigorous simulation for those components. In these conditions, the minimum number of trays is 8.024 (~8) and the actual number of trays is 14.023 (~14). This is coincident with the number of trays predicted by the model. The optimal reflux ratio (2.45) is significantly larger than the minimum reflux ratio (1.5), which is a consequence of the relatively large importance of the capital cost with respect to the energy costs in the TAC

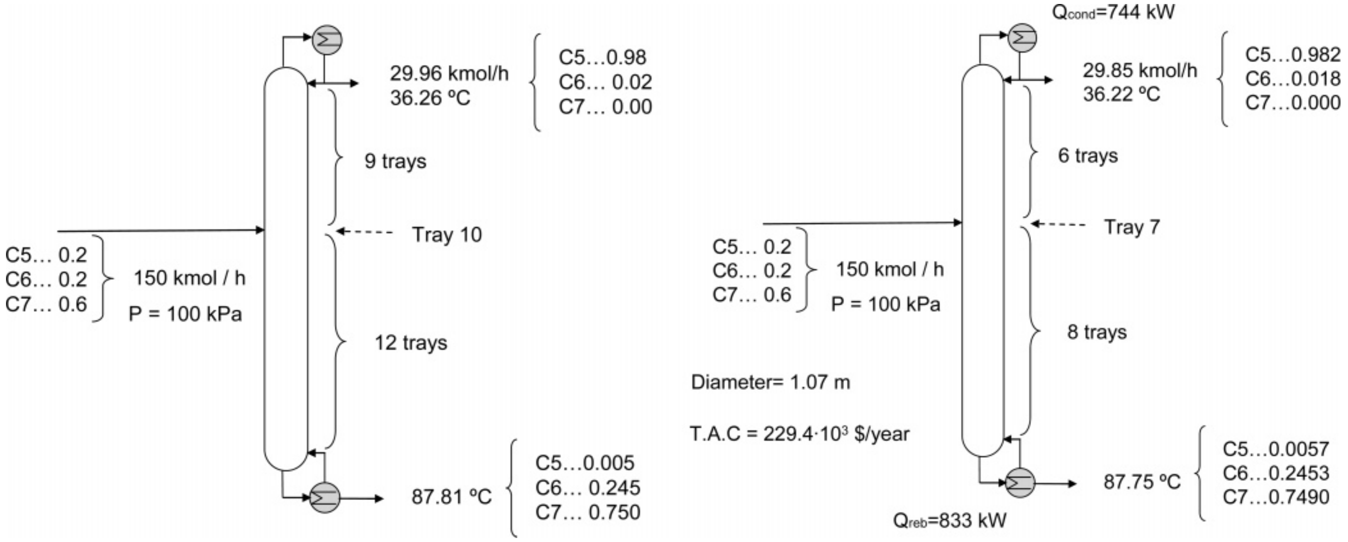

Figure 8. Optimal solution for example 1: (a) simplified objective function and (b) rigorous total annual cost minimization.

Table 3. Major Iterations in Example 1b[a]

Initial NLP	
existing trays from 1 to 40 (feed tray 20) $R = 1.916$ $D/F = 0.200$ objective = 370.6 × 10^3 \$/year; CPU time = 8.81 s	
Iteration 1	
master	NLP
trays from 15 to 25 $R = 2.366$ $D/F = 0.200$ objective = 210.0 × 10^3 \$/year CPU time (master gen.)	$R = 3.7084$ $D/F = 0.2000$ objective = 244.4 × 10^3 \$/year CPU time = 6.26 s
Iteration 2	
master	NLP
trays from 14 to 27 $R = 2.210$ $D/F = 0.200$ objective = 229.9 × 10^3 \$/year CPU time master gen.) = 14.9 s	$R = 2.6249$ $D/F = 0.2000$ objective = 236.75 × 10^3 \$/year CPU time = 6.03 s
Iteration 3	
master	NLP
trays from 14 to 28 $R = 2.447$ $D/F = 0.200$ objective = 241.3 × 10^3 \$/year CPU time (master gen.) = 15.0 s	$R = 2.4511$ $D/F = 0.2000$ objective = 229.5 × 10^3 \$/ year (optimum) CPU time = 1.95 s
Iteration 4	
master	NLP
trays from 14 to 26 $R = 2.878$ $D/F = 0.200$ objective = 242.6 × 10^3 \$/year CPU time (master gen.) = 14.9 s	$R = 2.8939$ $D/F = 0.2000$ objective = 237.3 × 10^3 \$/year CPU time =3.06 s

[a] R = reflux ratio. D/F = molar ratio of distillate to feed. Solution of the master problem took <1 s in all cases.

calculated by the Douglas rule (eq 3), which indirectly validates the model.

3.2. Example 2. In this example, the objective is to separate C3 from C4 alcohols. The minimum number of stages and the minimum reflux ratio calculated using the shortcut column implemented in HYSYS are 13.7 and 1.86, respectively. The independent variables are basically the same as in the previous examples. Molar fractions of C3s in the distillate and C4s in the bottoms >0.98 are specified as external constraints. A total of 70 trays are initially postulated with the feed located in tray 35. Trays 24−44 were considered as fixed. Data for example 2 are shown in Table 1. In this example, two NLPs were solved for the initialization step, including the maximum and minimum number of trays. The best solution obtained, trays 24−51, includes the postulated minimum number of trays in the rectifying section. Decreasing the number of fixed trays showed that the best configuration that was obtained continues to be optimal. The progress of the iterations are shown in Table 4, and the optimal configuration is shown in Figure 9. In this case, the total CPU time required was 202.7 s.

3.3. Example 3. Examples 1 and 2 involve mixtures with near-ideal equilibrium behavior. The next two examples involve nonideal mixtures. Example 3 consists of the separation of methanol from water in a mixture formed by acetone−methanol−water. The objective is a 99% recovery of both compounds in the distillate and bottoms, respectively. Data for this example are shown in Table 1. The procedure is similar to that in the previous examples. The optimum configuration is located in three major iterations, as shown in Table 5, and required 41.9 s of computational time. Figure 10 shows the optimal configuration obtained in this example.

It should be noted that, in all previous examples, the CPU time for solving the NLPs ranges between 3 and 15 s. The CPU time for solving the master problems was, in all cases, <1 s. The CPU time for generating the master problem is not negligible, and it is directly related to the number of total trays. It is ~15 s in example 1, ~45 s in example 2, and ~10 s in example 3. Note that these times are larger that the time needed to solve the NLPs. However, the CPU time for generating the master increases linearly with the number of total trays. Therefore, it is a penalty we have to pay for generating the master problem, but it is not expected to increase too much with the problem complexity, as will be shown in the next example.

3.4. Example 4: Extension to Complex Columns. The extension of the algorithm to complex columns is straightforward. If we define a column section to be the

Table 4. Major Iterations in Example 2[a]

Initial NLP

existing trays from 1 to 70 (feed tray 35)
$R = 2.1420$
$D/F = 0.500$
objective = 774.7 × 10^3 \$/year; CPU time = 23.7 s

Initial NLP

existing trays from 24 to 44
$R = 3.8585$
$D/F = 0.500$
objective = 608.2 × 10^3 \$/year; CPU time = 11.89

Iteration 1

master	NLP
trays from 24 to 50	$R = 2.7210$
$R = 2.322$	$D/F = 0.5000$
$D/F = 0.499$	objective = 546.8 × 10^3 \$/year
objective = 532.8 × 10^3 \$/year	CPU time = 7.75 s
CPU time (master gen.) = 94.4 s	

Iteration 2

master	NLP
trays from 24 to 51	$R = 2.6448$
$R = 2.638$	$D/F = 0.5000$
$D/F = 0.500$	objective = 534.14 × 10^3 \$/year (optimum)
objective = 563.21 × 10^3 \$/year	CPU time = 5.54 s
CPU time (master gen.) = 44.5 s	

Iteration 3

master	NLP
trays from 24 to 49	$R = 2.8156$
$R = 2.816$	$D/F = 0.500$
$D/F = 0.500$	objective = 547.7 × 10^3 \$/year
objective = 563.23 × 10^3 \$/year	CPU time = 3.66 s
CPU time (master gen.) = 46.9 s	

[a] R = reflux ratio. D/F molar ratio of distillate to feed.

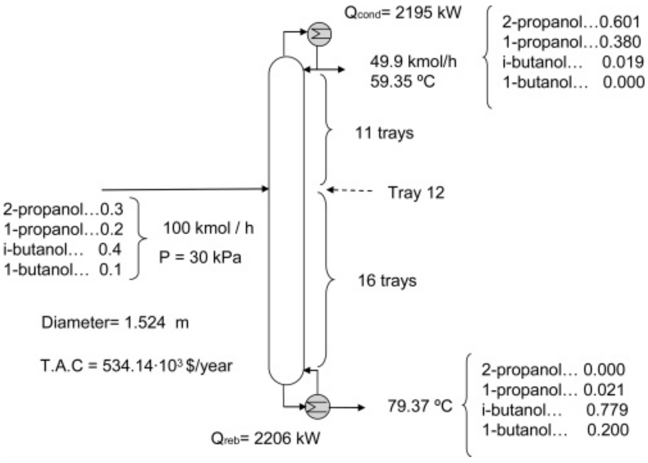

Figure 9. Optimal solution of example 2.

Table 5. Major Iterations in Example 3[a]

Initial NLP

existing trays from 1 to 30 (feed tray 20)
$R = 0.7987$
$D/F = 0.6975$
objective = 368.2 × 10^3 \$/year; CPU time = 3.95 s

Iteration 1

master	NLP
trays from 10 to 22	$R = 1.2028$
$R = 1.055$	$D/F = 0.6975$
$D/F = 0.696$	objective = 314.9 × 10^3 \$/year
objective = 308.1 × 10^3 \$/year	CPU time = 3.42 s
CPU time (master gen.) = 10.2 s	

Iteration 2

master	NLP
trays from 8 to 23	$R = 0.9947$
$R = 0.964$	$D/F = 0.6974$
$D/F = 0.697$	objective = 306.6 × 10^3 \$/year (optimum)
objective = 314.9 × 10^3 \$/year	CPU time = 2.59 s
CPU time (master gen.) = 10.3 s	

Iteration 3

master	NLP
trays from 7 to 23	$R = 0.9526$
$R = 0.952$	$D/F = 0.6975$
$D/F = 0.698$	objective = 308.7 × 10^3 \$/year
objective = 320.0 × 10^3 \$/year	CPU time = 5.08 s
CPU time (master gen.) = 10.4 s	

[a] R = reflux ratio. D/F molar ratio of distillate to feed.

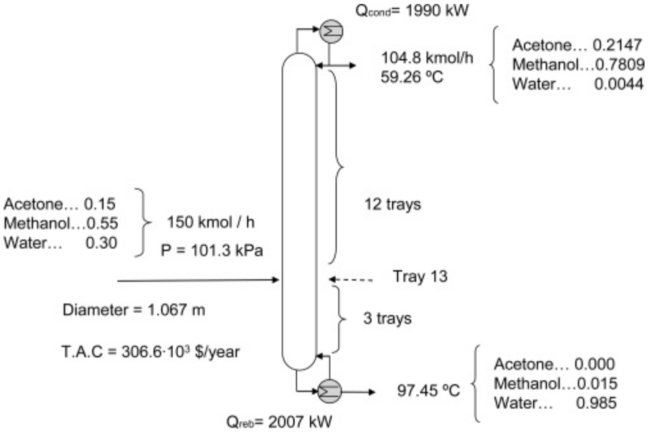

Figure 10. Optimal solution of example 3.

portion of a distillation column that is not interrupted by entering or exiting streams or heat flows, we only need to define a set of conditional trays in each column section and then apply the same procedure.

We illustrate the extension to complex columns with the extractive distillation of ethanol using ethylene glycol as an entrainer. Data for this example are shown in Table 1. Figure 11 shows the superstructure for the selection of feed location and number of trays for this example. The first column is given initially by 80 trays. The entrainer is fed in tray 20 and the feed in tray 60 (column numbered from top to bottom). These two trays are permanent. In the upper section of the column, if a given tray exists, all trays below it must exist. In the middle section, if a given tray exists, all trays above it (until the entrainer tray) must exist; and in the lower section, if a given tray exists, all trays above it must exist (see Figure 11). In the second column, 15 trays were postulated and the feed tray is located in tray 7. If the feed is completely specified and the pressure is fixed, there are five degrees of freedom for the NLPs. We have chosen for column 1 as independent variables the ratio of entrainer−feed, the reflux ratio, and the molar fraction ratio of water−ethanol in the bottom stream; for the second column we have chosen the reflux ratio and the molar fraction ratio of ethylene glycol−water in the bottoms. The molar fraction of each component must be >0.999, which is added through the external constraints.

The total CPU time required was 1280.9 s. The increase in the number of variables, as well as the difficulty of the problem, produces an increase in the CPU time for solving the NLPs, ranging from 100 to

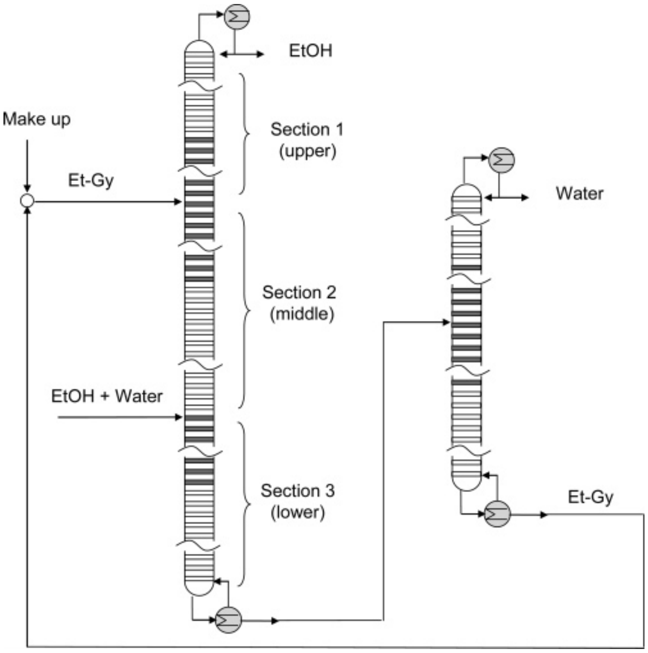

Figure 11. Superstructure for example 4.

200 s. The master problems continue to be solved in <1 s, and convergence is achieved in five major iterations. Note that the CPU time used for generating the master is ~95 s, as expected at the view of the total number of trays in the superstructure. The CPU time used in solving the NLPs is now larger than the time consumed in generating the master problem.

It is worth mentioning that the recycle stream can be torn and converged simultaneously during the optimization process. Finally, we should note that, in the objective function, only the cost of equipment and utilities, but not the cost of raw materials or considerations of cost vs purity, has been taken into account. Figure 12 shows the optimal configuration that was obtained. Table 6 shows the result of the major iterations.

Conclusions

This paper has proposed a systematic method for the rigorous design of distillation columns in which operational conditions (reflux and reboil ratios, recoveries, etc.) as well as structural parameters (number of trays, location of feed and product streams, etc.) are simultaneously optimized. It has been shown that the disjunctive representation of a distillation column in which trays are divided into permanent and conditional can be used in conjunction with a commercial process simulator. Deactivation of a tray can be achieved by fixing the tray efficiency to zero. In that way, the equilibrium equations are trivially satisfied and the tray becomes a simple vapor and liquid bypass without mass exchange. A decomposition algorithm that iterates between NLP problems (with a fixed structural configuration) and a special MILP master problem has shown good performance in different examples.

The master problem is defined by the accumulation of two contributions. The first is a linearization in terms of continuous variables similar to that with the outer-approximation algorithm that is widely used in MINLP or GDP problems. If all the equations relax like convex inequalities, then this linearization would be supporting hyperplanes that underestimate the objective function and overestimate the feasible region. The second contribution to the definition of the master problem takes into account the effect of adding or deleting a set of trays with respect to a base configuration. This extra contribution due to the existence or nonexistence of a set of trays is approximate, and therefore, no rigorous bounding properties for the master problem can be established. However, the numerical examples indicate that the proposed algorithm provide good solutions. Around 20 NLPs with different configurations for each of the examples were also solved. In all the cases, the best solution was that provided by the algorithm. However, only an extensive use of the algorithm by the scientific community could completely validate the algorithm.

The computational times have shown that the NLPs are solved in <2 min of CPU time, and the master problem is solved in <1 s on a PC (Pentium IV 2800 MHz). Note that, if the number of degrees of freedom increases and if there are a large number of recycle streams, the computational times for the NLPs can increase considerably. However, the optimization of continuous variables using process simulators is a problem that is out of the scope of this paper, and there

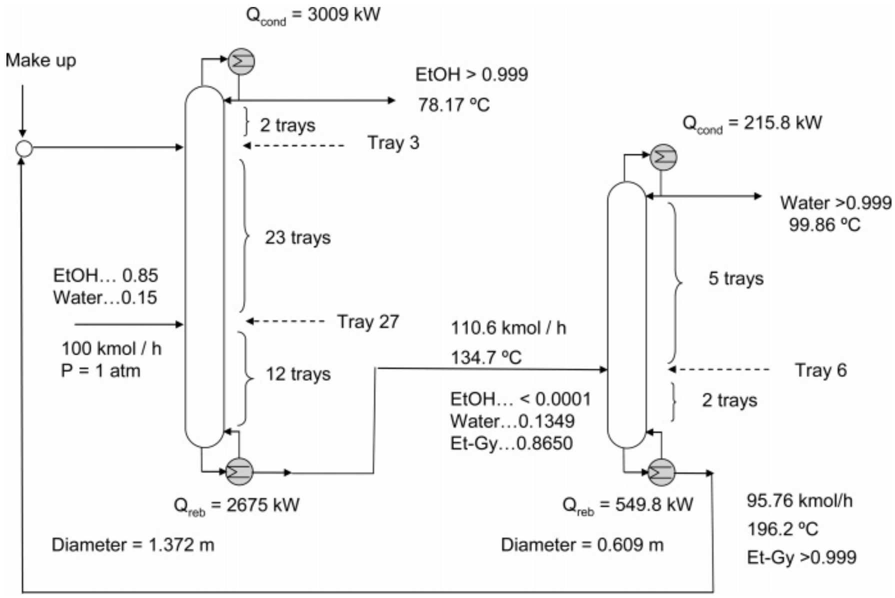

Figure 12. Optimal solution of example 4.

Table 6. Major Iterations in Example 4

E/F = molar ratio of entrainer to feed; $R1$ = reflux ratio column 1 $R2$ = reflux ratio column 2 $CR1$ = molar fraction ratio water−ethanol in the bottom stream (column 1) $CR2$ = molar fraction ratio ethylene glycol−water in bottoms (column 2)	
Initial NLP	
trays in column 1 from 1 to 80; feed tray 60; entrainer tray 20 trays in column 2 from 1 to 15 (feed tray 7) $E/F = 0.7818$ $R1 = 1.9624$ $CR1 = 2698$ $R2 = 0.1904$ $CR2 = 1.0772$ objective = 1122.9 × 10^3 \$/year; CPU time = 168 s	
Iteration 1	
master	NLP
trays column 1: from 18 to 27 and 60 to 71 trays column 2: from 3 to 10 $E/F = 1.00$ $R1 = 2.622$ $CR1 = 2000$ $R2 = 0.125$ $CR2 = 500$ objective = 794.1 × 10^3 \$/year CPU time (master generation) = 91.5 s	$E/F = 3.8369$ $R1 = 8.8051$ $CR1 = 1094.7$ $R2 = 1.7757$ $CR2 = 1887.4$ objective = 1881.4 × 10^3 \$/year CPU time = 190 s
Iteration 2	
master	NLP
trays column 1 from 18 to 29 and 60 to 72 trays column 2 from 3 to 9 $E/F = 3.1$ $R1 = 2.38$ $CR1 = 1299$ $R2 = 0.403$ $CR2 = 500$ objective = 905.27 × 10^3 \$/year CPU time (master generation) = 97.3 s	$E/F = 2.1969$ $R1 = 4.8156$ $CR1 = 1247.6$ $R2 = 0.8726$ $CR2 = 999$ objective = 1232.8 × 10^3 \$/year CPU time = 108 s
Iteration 3	
master	NLP
trays column 1 from 18 to 35 and 60 to 72 trays column 2 from 3 to 9 $E/F = 2.5$ $R1 = 2.23$ $CR1 = 1781$ $R2 = 0.404$ $CR2 = 500$ objective = 925.41 × 10^3 \$/year CPU time (master generation) = 96.4 s	$E/F = 1.2582$ $R1 = 2.8995$ $CR1 = 1673.5$ $R2 = 0.4807$ $CR2 = 999$ objective = 945.1 × 10^3 \$/year CPU time = 96.2
Iteration 4	
master	NLP
trays column 1 from 18 to 43 and 60 to 72 trays column 2 from 2 to 9 $E/F = 2.1$ $R1 = 2.231$ $CR1 = 1781$ $R2 = 0.404$ $CR2 = 500$ objective = 972.1 × 10^3 \$/year CPU time (master generation) = 95.8 s	$E/F = 0.9490$ $R1 = 2.3086$ $CR1 = 1989.3$ $R2 = 0.2843$ $CR2 = 1002.2$ objective = 885.1 × 10^3 \$/year (optimum) CPU time = 103.8 s
Iteration 5	
master	NLP
trays column 1 from 18 to 42 and 60 to 72 trays column 2 from 3 to 9 $E/F = 1.90$ $R1 = 2.44$ $CR1 = 3875$ $R2 = 0.336$ $CR2 = 500$ objective = 975.1 × 10^3 \$/year CPU time (master generation) = 96.2 s	$E/F = 0.9747$ $R1 = 2.3530$ $CR1 = 2154.9$ $R2 = 0.3809$ $CR2 = 999$ objective = 886.0 × 10^3 \$/year CPU time = 137.7 s

are reliable and well-established methodologies; see, for example, the work of Biegler and Hughes[44] or Lang and Biegler.[45]

Future directions of this research include testing the proposed method in a larger number of academic and industrial cases and comparison of the results with actual designs. Future directions also include the extension to the synthesis of distillation sequences where a full column can completely disappear in a superstructure, extensions to thermally coupled systems and other complex systems, and the integration in a general superstructure synthesis framework.

Acknowledgment

The authors gratefully acknowledge the financial support from the “Ministerio de Ciencia y Tecnología” in Spain under project PPQ−2002-01734.

Literature Cited

(1) Shultz, M. A.; Stewart, D. G.; Harris, J. M.; Rosenblum, S. P.; Shakur, M. S.; O'Brien, D. E. Reduce costs with dividing wall columns. *Chem. Eng. Prog.* **2002**, May, 64−70.

(2) Agrawal, R. Synthesis of Distillation Column Configuration for a Multicomponent Separation. *Ind. Eng. Chem. Res.* **1996**, *35*, 1059−1071.

(3) Agrawal, R.; Fidkowski, Z. T. More operable arrangements of fully thermally coupled distillation columns. *AIChE J.* **1998**, *44*, 2565.

(4) Agrawal, R. More operable fully thermally coupled distillation column configurations for multicomponent distillation. *Trans. Inst. Chem. Eng.* **1999**, 77.

(5) Agrawal, R.; Fidkowski, Z. T. New Thermally Coupled Schemes for Ternary Distillation. *AIChE J.* **1999**, *45* (3), 485.

(6) Caballero, J. A.; Grossmann, I. E. Generalized disjunctive programming model for the optimal synthesis of thermally linked distillation columns. *Ind. Eng. Chem. Res.* **2001**, *40* (10), 2260−2274.

(7) Caballero, J. A.; Grossmann, I. E. Logic based methods for generating and optimizing thermally coupled distillation systems. In *Computer Aided Chemical Engineering 10*; Grievnik, J., van Schijndel, Eds.; ESCAPE-12; Elsevier: New York, 2002; pp169−174.

(8) Caballero, J. A.; Grossmann, I. E. Thermodynamically equivalent configurations for thermally coupled distillation. *AIChE J.* **2003**, *49* (11), 2864−2884.

(9) Caballero, J. A.; Grossmann, I. E. Design of Distillation Sequences: From conventional to fully thermally coupled distillation systems. *Comput. Chem. Eng.* **2004**, *28*, 2307−2329.

(10) Barbosa, D.; Doherty, M. F. Simple distillation of homogeneous reactive mixtures. *Chem. Eng. Sci.* **1988**, *43* (3), 541−550.

(11) Blessling, B.; Schembecker, G.; Simmrock, K. H. Design of process with reactive distillation line diagrams. *Ind. Eng. Chem. Res.* **1997**, *36*, 3032−3042.

(12) Lipnizki, F.; Field, R. W.; Ten, P. K. Pervaporation-Based Hybrid Process: A Review of Process Design, Application and Economics. *J. Membr. Sci.* **1999**, *153*, 183−210.

(13) Kister, H. Z. *Distillation Design*; McGraw-Hill: New York, 1992.

(14) Naphtali, L. M.; Sandholm, D. P. Multicomponent distillation by linearization. *AIChE J.* **1971**, *17*, 148.

(15) Goldstein, R. P.; Stanfield R. B. Flexible Method for the Solution of Distillation Design Problems Using the Newton−Raphson Technique. *Ind. Eng. Chem. Proc. Des. Dev.* **1970**, *9*, 78.

(16) Ishii, Y.; Otto, F. F. A General Algorithm for Multistage Multicomponent Separation Calculations. *Can. J. Chem. Eng.* **1973**, *51*, 601.

(17) Boston, J. F.; Sullivan, S. L., Jr. A New Class of Solution Methods for Multicomponent, Multistage Separation Processes. *Can. J. Chem. Eng.* **1974**, *52*, 52−63.

(18) Russell, R. A. A Flexible and Reliable Method Solves Single-Tower and Crude-Distillation-Column Problems. *Chem. Eng.* **1983**, October, 53−59.

(19) Rose, A.; Sweeny, E. F.; Schrodt, V. N. Continuous Distillation Calculation by Relaxation. *Method. Ind. Eng. Chem.* **1958**, *50*, 737.

(20) Vickery, D. J.; Taylor, R. Path-following approaches to the solution of multicomponent, multistage separation process problems. *AIChE J.* **1986**, *32* (4), 547.

(21) Grossmann, I. E.; Aguirre, P. A.; Barttfeld, M. Optimal Synthesis of Complex Distillation Columns Using Rigorous Models. In *Computer Aided Chemical Engineering 18*; Barbosa-Pavoa, A., Matos, H., Eds.; ESCAPE-14; Elsevier: New York, 2004; pp 53−74.

(22) Sargent, R. W. H.; Gaminibandara, K. Approaches to Chemical Process Synthesis. In *Optimization in Action*; Dixon, L. C., Ed.; Academic Press: London, 1976.

(23) Viswanathan, J.; Grossmann, I. E. A Combined Penalty Function and Outer Approximation Method for MINLP Optimization. *Comput. Chem. Eng.* **1990**, *14* (7), 769.

(24) Viswanathan, J.; Grossmann, I. E. An Alternate MINLP Model for Finding the Number of Trays Required for a Specified Separation Objective. *Comput. Chem Eng.* **1993**, *17* (9), 949.

(25) Ciric, A. R.; Gu, D. Synthesis of Nonequilibrium Reactive Distillation Processes by MINLP Optimization. *AIChE J.* **1994**, *40* (9), 1479−1487.

(26) Bauer, M. H.; Stichlmair, J. Design and Economic Optimization of Azeotropic Distillation Processes Using Mixed Integer Nonlinear Programming. *Comput. Chem. Eng.* **1998**, *22* (9), 1271−1286.

(27) Dünnebier, G.; Pantelides, C. C. Optimal Design of Thermally Coupled Distillation Columns. *Ind. Eng. Chem. Res.* **1999**, *38*, 162−176.

(28) Grossmann, I. E. Review of Nonlinear Mixed Integer and Disjunctive Programming Techniques. *Opt. Eng.* **2002**, *3* (3), 227−252.

(29) Yeomans, H.; Grossmann, I. E. Disjunctive Programming Models for the Optimal Design of Distillation Columns and Separation Sequences. *Ind. Eng. Chem. Res.* **2000**, *39*, 1637−1648.

(30) Jackson, J. R.; Grossmann, I. E. A Disjunctive Programming Approach for the Optimal Design of Reactive Distillation Columns. *Comput. Chem. Eng.* **2001**, *25*, 1661−1663.

(31) Barttfeld, M.; Aguirre, A. P.; Grossmann, I. E. Alternative Representations and Formulations for the Economic Optimization of Multicomponent Distillation Columns. *Comput. Chem. Eng.* **2003**, *27*, 363−383.

(32) Barttfeld, M.; Aguirre, A. P. Optimal Synthesis of Multicomponent Zeotropic Distillation Processes. 1. Preprocessing Phase and Rigorous Optimization for a Single Unit. *Ind. Eng. Chem. Res.* **2002**, *41*, 5298−5307.

(33) Lang, Y.-D.; Biegler, L. T. Distributed Stream Method for Tray Optimization. *AIChE J.* **2002**, *48* (3), 582−595.

(34) Diwekar, U. M.; Grossmann, I. E.; Rubin, E. S. An MINLP Process Synthesizer for a Sequential Modular Simulator. *Ind. Eng. Chem. Res.* **1992**, *31*, 313−322.

(35) Reneaume, J.-M. F.; Koehret, B. M.; Joulia, X. L. Optimal Process Synthesis in a Modular Simulator Environment: New Formulation of the Mixed-Integer Nonlinear Programming Problem. *Ind. Eng. Chem. Res.* **1995**, *34*, 4378−4394.

(36) Díaz, M. S.; Bandoni, J. A. A Mixed Integer Optimization Strategy for a Large Chemical Plant in Operation. *Comput. Chem. Eng.* **1996**, *20* (5), 531−545.

(37) Kravanja, Z.; Grossmann, I. E. A computational Approach for the Modeling/Decomposition Strategy in the MINLP Optimization of Process Flowsheets with Implicit Models. *Ind. Eng. Chem. Res.* **1996**, *35*, 2065−2070.

(38) Kocis, G. R.; Grossmann, I. E. Relaxation Strategy for the Structural Optimization of Process Flowsheets. *Ind. Eng. Chem. Res.* **1987**, *26*, 1869−1880.

(39) *Matlab*, release 13; The Mathworks: Natick, MA, copyright 1984−2002.

(40) *HYSYS*, version 3.1; Hyprotech Ltd.: Calgary, Alberta, Canada, 1995−2002.

(41) Duran, M.; Grossmann, I. E. An Outer Approximation Algorithm for a Class of Mixed Integer Nonlinear Programs. *Math. Program.* **1986**, *36*, 307−339.

(42) Turton, R.; Bailie, R. C.; Whiting, W. B.; Shaeiwitz, J. A. *Analysis, Synthesis and Design of Chemical Processes*. McGraw-Hill: New York, 1998.

(43) Douglas, J. M. Conceptual Design of Chemical Processes. McGraw-Hill: New York, 1988.

(44) Biegler, L. T.; Hughes, R. R. Infeasible Path Optimization with Sequential Modular Simulators. *AIChE J.* **1982**, *28* (6), 994–1002.

(45) Lang, Y.-D.; Biegler, L. T. A Unified Algorithm for Flowsheet Optimization. *Comput. Chem. Eng.* **1987**, *11* (2), 143–158.

Received for review January 20, 2005
Revised manuscript received May 9, 2005
Accepted June 20, 2005

IE050080L