RIGOROUS DESIGN OF COMPLEX DISTILLATION COLUMNS USING PROCESS SIMULATORS AND THE PARTICLE SWARM OPTIMIZATION ALGORITHM

J. Javaloyes-Antón, R. Ruiz-Femenia, and J. A. Caballero

September 2013

ABSTRACT

We present a derivative-free optimization algorithm coupled with a chemical process simulator for the optimal design of individual and complex distillation processes using a rigorous tray-by-tray model. The optimal synthesis of complex distillation columns is a non-trivial problem due to the discrete nature of the tray-by-tray column model, and also because of the high degree of non-linearity and non-convexity of the underlying MESH equations (mass balances, equilibrium, summation of molar fractions in both phases equal to 1 and heat balances). The proposed approach serves as an alternative tool to the various models based on nonlinear programming (NLP) or mixed-integer nonlinear programming (MINLP). This is accomplished by combining the advantages of using a commercial process simulator (Aspen Hysys), including especially suited numerical methods developed for the convergence of distillation columns, with the benefits of the particle swarm optimization (PSO) metaheuristic algorithm, which does not require gradient information and has the ability to escape from local optima. The method developed herein is based on the superstructure developed by *Yeomans and Grossmann (2000)*, in which the non–existing trays are considered as simple bypasses of liquid and vapor flows. The implemented tool provides the optimal configuration of distillation column systems, which includes continuous and discrete variables, through the minimization of the total annual cost (TAC). The robustness and flexibility of the method is proven through the successful design and synthesis of three distillation systems of increasing complexity.

CONTENTS

1 INTRODUCTION

The costs of a chemical process are often dominated by the costs for the separation and purification of the products. Between the different separation techniques, distillation is one of the most important and commonly used in all chemical and petrochemical industries, even though these equipment units have very low energy efficiency – provides heat in the reboiler, to cool down afterwards in the condenser –. In fact, about 90% of all separation and purification operations in the United States are distillations, which represented around 3% of the total US energy consumption in 2002, that is, 91 GW or 54 million tons of crude oil (*Soave & Feliu, 2002*). It is therefore desirable to dispose robust and reliable tools to design optimal distillation processes, to reduce the investment and operating costs of these units (specifically the energy consumption, which has a large economic and environmental impact).

The economic optimization of complex distillation columns is a nontrivial problem, and continues to be a major challenge in the design of chemical processes due to the discrete nature of the tray-by-tray column model, and also because of the high degree of non-linearity and non-convexity of the underlying MESH equations (mass, equilibrium, summation and energy). In the optimization, discrete decisions are related to the calculation of the number of trays, feed and side product streams location, whereas continuous variables are related to the operating conditions (e.g. reflux ratio, boilup ratio, distillate to feed ratio…).

A number of different approaches, based on mathematical programming have been developed in the past decades in order to provide reliable rigorous try-by-tray optimization models. Most of these methods are formulated as a Mixer Integer Non-Linear Programming problems (MINLP), where the distillation column is modeled as a superstructure with variable column ends (reflux and reboil location, or even both), or as a Generalized Disjunctive Programing (GDP) representation, where the column is modeled as a superstructure in which the non–existing trays are considered as bypasses of liquid and vapor flows.

Methods that have addressed the solution of MINLP problems, include the branch and bound (BB) (*Gupta & Ravindran, 1985*; *Nabar & Schrage, 1991; Borchers & Mitchell, 1994*; *Stubbs & Mehrotra, 1999*; *Leyffer, 2001*), Generalized Benders Decomposition (GBD) (*Benders, 1962*; *Geofrion, 1972*) Outer Approximation (OA) (*Duran & Grossmann, 1986*; *Yuan et al. 1988; Fletcher & Leyffer, 1994*) among others. But, all these methods assume convexity to guarantee convergence to the global optimum.

A significant advance in the modeling and solution of design problems was the introduction of Disjunctive Programming (*Raman and Grossmann, 1994*; Lee and Grossmann, 2000) both from a point of view of modeling, and solution. The logical part of a model can be represented by a set of disjunctions that allow the researcher focusing on the model itself decoupling, at least partially, the model formulation from the solution. The solution then can be performed either by an automatic reformulation to an MI(N)LP problem (i.e. using a convex hull or a big M approach or using directly a logic-based algorithm i.e. (*Türkay and Grossmann, 1996*). However, there are no many public implementations of logic algorithms and there is a large open field for researching.

For more information about these methods we recommend the lecture of the analysis of the current framework for MI(N)LP and GDP formulation by *Grossmann and Ruiz (2009)* .

The first successful MINLP model to optimize simultaneously the number of trays and feed location for a specified separation, was published by *Viswanathan and Grossmann (1993)*. These authors proposed a superstructure with multiple feed locations for the reflux and boilup flows and assigned binary variables to the existence of each of those potential streams, see Figure 1.1a.

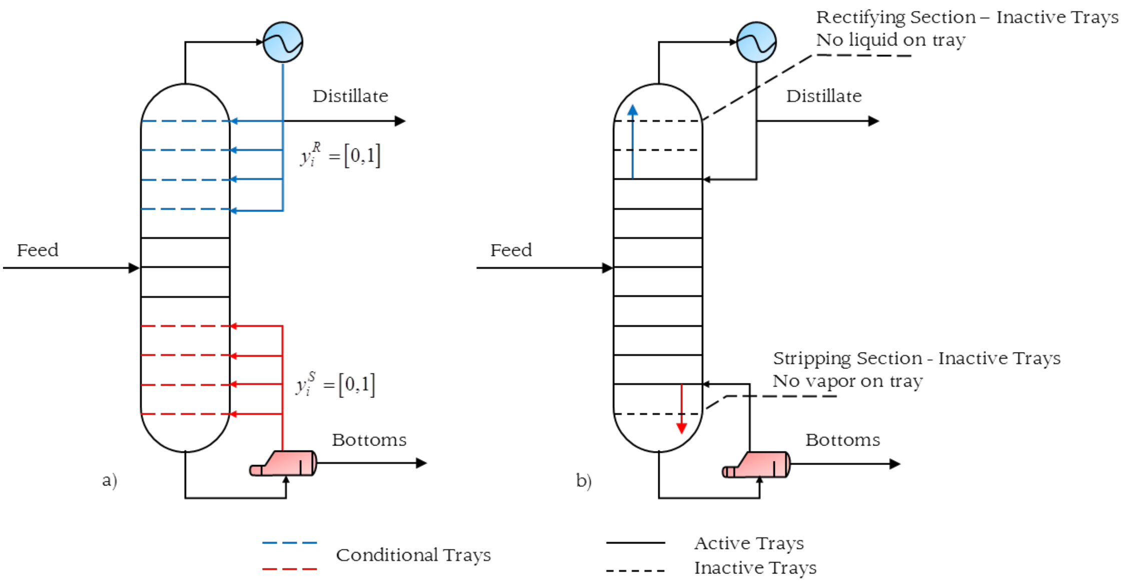
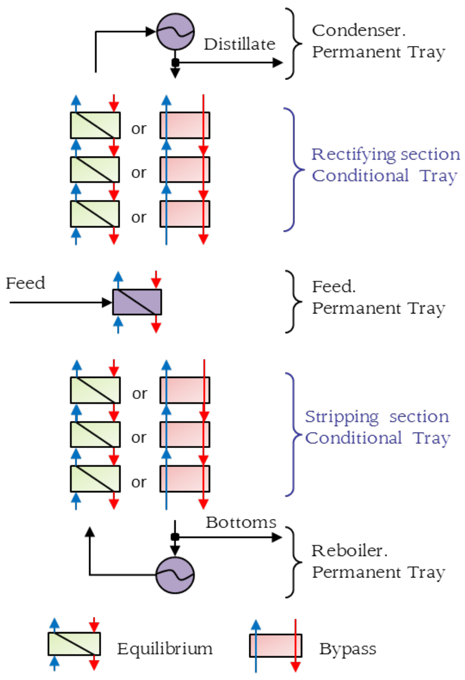


Figure 1.1 a) Superstructure of Viswanathan & Grossmann for the optimal feed location, total number of trays and optimal operation conditions. b) Inactive trays in the Viswanathan & Grossmann model.

A major difficulty of the resulting model is the fact that the vapor-liquid equilibrium conditions are enforced in all trays of the column; even in the case of inactive trays (see Figure 1.1b) where no mass transfer takes place. In these trays there is a zero liquid flow in the rectifying section or a zero vapor flow in the stripping section, which can produce numerical problems due to the convergence of the equilibrium equations (which are high non-linear expressions) with a zero value in the flow of one of the two phases (*Barttfeld et al., 2003*).

Despite these drawbacks, the model of Viswanathan and Grossmann has been successfully applied by different research groups for optimizing individual columns and superstructures. For example, *Ciric and Gu (1994)* used the MINLP approach for the optimization of a reactive distillation column. *Bauer and Stichlmair (1998)* applied the MINLP approach to the synthesis of sequences of azeotropic columns, and *Dünnebier and Pantelides (1999)* used the model to generate sequences of thermally coupled distillation columns, to name some of them.

In order to overcome some of the difficulties in MINLP models, *Yeomans and Grossmann (2000)* proposed a Generalized Disjunctive Programming (GDP), in which the non-existing trays are considered as simple bypasses of vapor and liquid flows without mass transfer. Therefore, mass and energy balances are trivially satisfied, and the only difference with active trays is in the application of the equilibrium equations (see Figure 1.2).



$$\begin{aligned}&\min \quad Cost\\&s.t.\\&\text{Permanent trays}\\&\quad \text{MESH equations}\\&\text{Conditional trays}\\&\quad \text{Mass and energy balances}\\&\quad \begin{bmatrix} Y_n \\ \text{VLE equations} \end{bmatrix} \vee \begin{bmatrix} \neg Y_n \\ \text{Not VLE equations (bypass)} \end{bmatrix}\end{aligned}$$

Figure 1.2 Superstructure of Yeomans and Grossmann for the optimal feed location, total number of trays and optimal operation conditions.

Barttfeld et al., (2003) showed that multiple representations for the optimization of a single distillation column are possible when it is solved with a GDP formulation. However, the computational results showed that the most effective alternative was the original configuration proposed by Yeomans and Grossmann.

All previous models were developed in an equation based environment, and required sophisticated initialization techniques to get a feasible solution. In addition, because of the non-convexity of the problem, only local optima are guarantee by these algorithms and the quality of the solution strongly depend on the initialization point of the search. As consequence of these difficulties, the resulting optimization tools are very complex, and so, only persons skilled in the art are able to utilize and adapt to their own needs. An alternative for handling these very difficult or even unsolvable problems, preserving the mathematical programming approach, is to substitute the rigorous models by simplified ones. Nevertheless, this procedure can entail the loss of good solutions *Gross et al., (1998)*.

Caballero et al., (2005) proposed an algorithm that integrates a process simulator in a Generalized Disjunctive Programming formulation. In this way, all numerical aspects related to the convergence of a distillation column are solved at the level of the process simulator. Therefore, taking advantage of the tailored numerical techniques specially developed to converge distillation columns included in the process simulators, the difficulties related with the initialization are overcome.

However, some important difficulties arises with this approach due to the modular architecture of the process simulators, in which the different blocks (processing units) are “black box” models for the users, and usually there is no access to the original code and derivative information is not available.

This fact may be significant because base gradient algorithms for solving MINLP, as the mentioned before, require accurate derivative information; and when a process simulator is used, derivatives for the design variables can only be obtained by perturbation, which introduces the following two important drawbacks:

- the perturbation of a variable requires solving all the flowsheet each time a variable is perturbed, resulting in a significant increase in the CPU computation time, and
- unit operations in process simulators (“black boxes”) introduce numerical noise preventing the calculation of accurate derivatives.

This numerical noise is due to low sensitivities of some variables to the convergence criteria used by the model. For example, in a distillation column the mass and energy balance are checked and closed with strict tolerance criteria. However, reboiler and condenser heat flows can change in orders of magnitude larger than the mass or energy tolerances. This is not relevant from the point of view of simulation – the conditioning of the system is low enough to assure accurate values – but this could produce catastrophic effects if the derivatives must be calculated using perturbations.

This can be check with the following numerical experiment using a process simulator (*Caballero & Grossmann, 2008*):

1. Converge the distillation column shown in Figure 1.3 with fixed values of the reflux and boilup ratios, and read the heat load in reboiler.
2. Randomly select new values of the reflux and boilup ratios and converge the column again.
3. Recover the values for step one and repeat

The reboiler duty should be the same for the same fixed reflux and boilup rations, but there is a small dispersion in the values (see Figure 1.4a). This noise is usually negligible for practical purposes (cost estimations), but introduce a significant error in the derivatives (see Figure 1.4b). Therefore when a process simulator is used derivative information is expensive to obtain and also somewhat noisy.

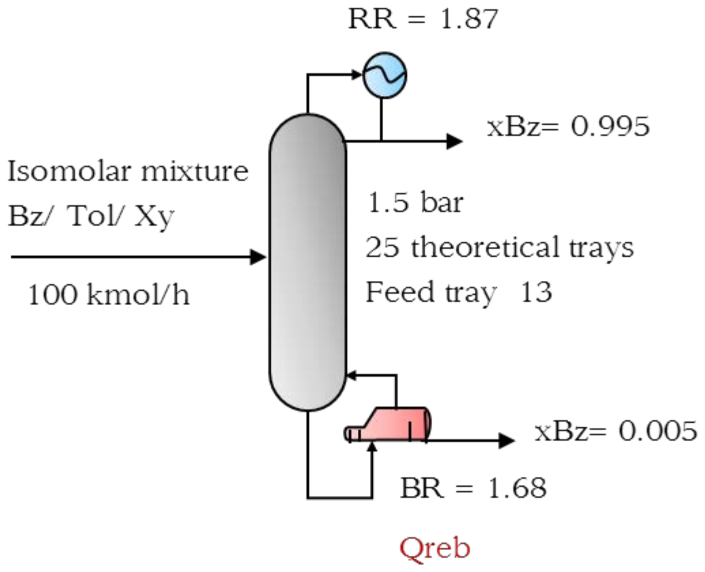


Figure 1.3 Distillation column flowsheet.

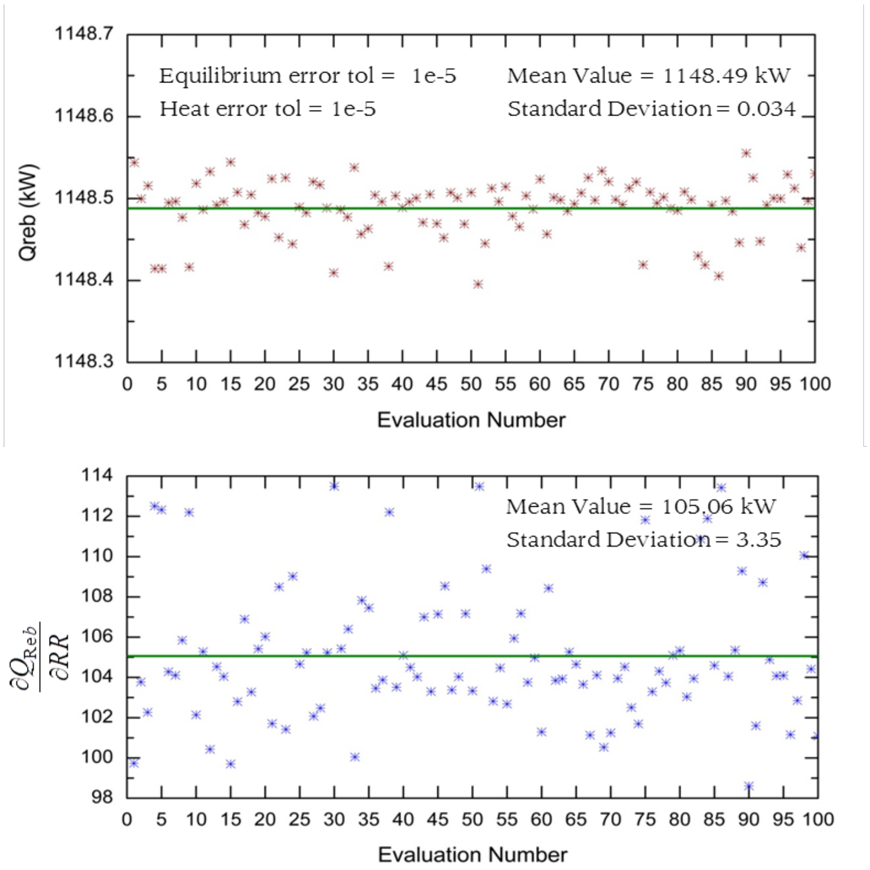


Figure 1.4 a)Effect of the numerical noise in the Reboiler duty and b) in the derivative with respect the reflux ratio.

It is possible to reduce this noise increasing the perturbation parameter (~0.01), and/or tightening the convergence tolerances in the process simulator (equilibrium and heat error ≤ 10^{-6}). However, this makes difficult to converge the flowsheet, especially for complex systems with recycle streams and sequences of distillation columns.

These difficulties can be overcome by derivative–free optimization algorithm. Metaheuristic algorithms coordinate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima. There are a wide variety of metaheuristics. We may cite, among others, simulated annealing (SA), tabu search (TS), variable neighborhood search (VNS), scatter search, ant colony optimization (ACO), iterated local search (ILS), particle swarm optimization21 (PSO), and genetic algorithms (GAs).

Many free-derivative search algorithms are population-based procedures, where an individual represents a particular solution to the optimization problem and a population is a set of individuals competing with each other with respect to their objective function values. Since these algorithms do not require gradient information, it is possible to treat the objective function as a black-box. This black-box evaluation can be performed by any simulation software if an interface between process simulator and optimizer is given. Although metaheuristic algorithms are not able to guarantee the optimality of the solutions found, gradient-based methods (which theoretically can provide such a certification) are often incapable of finding solutions whose quality is close to that obtained by the metaheuristics.[22] This is especially true for real-world problems, which exhibit high levels of complexity. Perhaps the most serious disadvantages of metaheuristic algorithms are that the number of function evaluations to converge could be large, and as well as they exhibit poor performance in highly constrained systems. However, the last problem, is easily solved when a process simulator is used, since all mass and energy balances are solved at the level of process simulator.

2 OBJECTIVE

With the above in mind, the purpose of this paper is to develop an optimization tool for designing complex distillation processes that combines the advantages of using a process simulator (Aspen Hysys) with the benefits of the PSO metaheuristic algorithm, which is implemented in Matlab code. Thus, all numerical aspects related to the convergence of distillation columns, including the selection of thermodynamic models, are specified at the level of the process simulator; and the external PSO optimizer (Matlab) is interfaced with the simulator in order to solve the optimization problem. Everything is controlled by an external executive program.

The rest of the work is organized as follows. In the next section we introduce an overview of the Particle Swarm Optimization algorithm, for then describing the basis of our methodology applied to the optimization of a single conventional distillation column. Next, we illustrate the application of the proposed methodology with three case studies, which show how to develop different superstructures based on the model of the single column. Finally, we discuss the conclusions that can be drawn from this work.

3 THE ORIGINAL PARTICLE SWARM OPTIMIZATION ALGORITHM

The PSO algorithm is a stochastic population-based method for solving global optimization problems, first proposed by Kennedy and Eberhart in 1995 (*Kennedy, J. & Eberhart, R., 1995*). The original algorithm was inspired by social behavior of bird flocking, fish schooling and swarm theory.

The algorithm maintains a population of particles that can move in the D-dimensional search space, and where each particle represents a potential solution to an optimization problem. Figure 3.1 shows a basic flowchart depicting the general PSO algorithm. One of the reasons that make PSO so attractive is that there are few parameters to adjust, and it is easy to implement.

The system is initialized with a population of random particles *i*, distributed in the search space. Each of those □□ and velocity □□, and its personal best position □□ (the best position that the particle has visited since the beginning of the algorithm). For a minimization task, a position yielding the smaller function value is regarded as the best position of particle *i*. The personal best position is updated according to the equation (3.1), with the dependence on the time step *k*.

$$p_{k+1}^{i} = \begin{cases} p_k^i & \text{if } f\left(x_{k+1}^i\right) \geq f\left(p_k^i\right) \\ x_k^i & \text{if } f\left(x_{k+1}^i\right) < f\left(p_k^i\right) \end{cases} \qquad i \in [1,K\ s] \tag{3.1}$$

□□ is updated according to equation (3.2):

$$x_{k+1}^{i} = x_k^i + v_{k+1}^i \tag{3.2}$$

where v_{k+1}^i is the particle’s new velocity vector. Stand out that the new velocity vector is calculated according to its previous □□□ from the best position found by itself □□□ (cognitive □□□□□□□□ (social term), see equation (3.3). In other words, all the particles are accelerated in the direction of the best particle, but also in the direction of their own best experience.

$$v_{k+1}^{i} = w_k v_k^{i} + c_1 r_1 \left(p_k^{i} - x_k^{i}\right) + c_2 r_2 \left(p_k^{global} - x_k^{i}\right) \tag{3.3}$$

□□ is the inertia weight, which is employed to control the impact of the previous velocities on the current one (scaling factor associated with the previous velocities); r_1 and r_2 are two uniform random vectors, whose elements are between 0 and 1, which gives the stochastic nature of the algorithm and c1 and c2 are two positive acceleration coefficients, called the cognitive and social parameter, respectively, and they influence the maximum size of the step that a particle can take in a single iteration.

According to the interaction scheme between the particles, two main versions of the PSO exist. In the first version, called *gbest*, every individual particle is attracted to the best □□□-□□□□□□. This structure then is equivalent to a fully connected social network. In the second one, called *lbest* (*g* and *l* stand for “global” and “local”), each individual particle is □□□-□□□□□□. For instance, if $z = 2$, then each individual particle i will be influenced by the best performance among a group made up of particles $i-1$, i, and $i+1$.

gbest Model

The *gbest* model offers a faster rate of convergence (*Eberhart, P. et al., 1996*) at the expense of robustness. This model maintains only a single best solution, called the *global best particle*, across all the particles in the swarm. Thus, this particle acts as a sole attractor, accelerating all the particles towards it. The main disadvantage of the *gbest* topology is that it is unable to explore multiple optimal □□□-□□□□□□ is therefore

$$\begin{aligned} p_k^{G-global} &\in \left\{p_k^1, p_k^2, \ldots, p_k^s\right\} \middle| f\left(p_k^{G-global}\right) \\ &= \min\left\{f\left(p_k^1\right), f\left(p_k^1\right), \ldots, f\left(p_k^s\right)\right\} \end{aligned} \tag{3.4}$$

and the new velocity vector is obtained according to equation (3.5).

$$v_{k+1}^{i} = w_k v_k^{i} + c_1 r_1 \left(p_k^{i} - x_k^{i}\right) + c_2 r_2 \left(p_k^{G-global} - x_k^{i}\right) \tag{3.5}$$

lbest Model

The *lbest* model tries to prevent premature convergence by maintaining multiple attractors. A subset of particles is *□□,□□-□□□□□□,* is then selected. Assuming that the particle indices wrap around at *s* (*s* is the size of the swarm), which means that the first and last particles are connected, the *lbest* update equation for a neighborhood of size *z* is given by equations (3.6) and (3.7).

$$N_j = \left\{ p_k^{i-z}, p_k^{i-(z+1)}, \ldots, p_k^{i-1}, p_k^{i}, p_k^{i+1}, \ldots, p_k^{i-1}, p_k^{i+z} \right\} \tag{3.6}$$

$$p_k^{L-best,j} \in N_j \left| f\left(p_k^{L-best,j}\right) = \min\left\{ f\left(\Omega\right)\right\}, \quad \forall \Omega \in N_j \right. \tag{3.7}$$

and the new velocity vector is therefore calculated according to equation (3.8).

$$v_{k+1}^{i} = w_k v_k^{i} + c_1 r_1 \left(p_k^{i} - x_k^{i}\right) + c_2 r_2 \left(p_k^{L-global,j} - x_k^{i}\right) \tag{3.8}$$

Note that the *gbest* model is actually a special case of the *lbest* model with z= s.

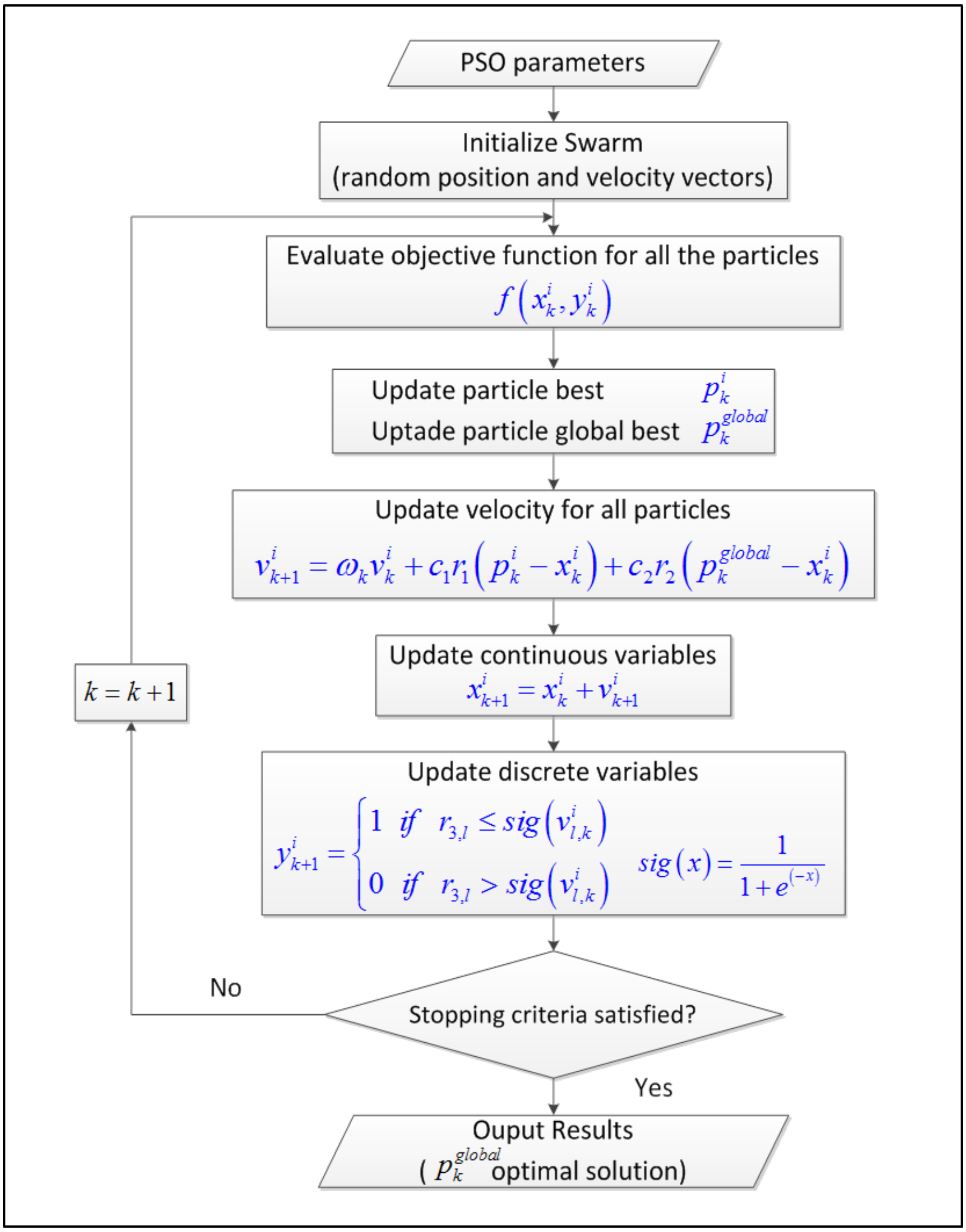

PSO parameters
Initialize Swarm
(random position and velocity vectors)
Evaluate objective function for all the particles
$f\left(x_k^i, y_k^i\right)$
Update particle best p_k^i
Uptade particle global best p_k^{global}
Update velocity for all particles
$v_{k+1}^i = \omega_k v_k^i + c_1 r_1\left(p_k^i - x_k^i\right) + c_2 r_2\left(p_k^{global} - x_k^i\right)$
Update continuous variables
$x_{k+1}^i = x_k^i + v_{k+1}^i$
Update discrete variables
$y_{k+1}^i = \begin{cases} 1 & if \quad r_{3,l} \le sig\left(v_{l,k}^i\right) \\ 0 & if \quad r_{3,l} > sig\left(v_{l,k}^i\right) \end{cases}$
$sig(x) = \frac{1}{1+e^{(-x)}}$
$k = k+1$
No
Stopping criteria satisfied?
Yes
Ouput Results
(p_k^{global} optimal solution)

Figure 3.1 Basic scheme of the gbest PSO algorithm for continuous and discrete variables

The Binary PSO

In the discrete binary version of PSO (*Kennedy, J. & □□*, that contains the binary variables associated with particle , is updated according to:

$$y_{l,k}^i = \begin{cases} 0 & if\ r_{3,l}\ \pounds\ sig\left(v_{l,k}^i\right) \\ 1 & if\ r_{3,l} > sig\left(v_{l,k}^i\right) \end{cases} \tag{3.9}$$

where r_3 is an uniform random vector, whose elements are between □□,□□ is the sigmoid function:

$$sig\left(v_{l,k}^{i}\right)=\frac{1}{1+e^{\left(-v_{l,k}^{i}\right)}} \tag{3.10}$$

The main body of the algorithm consists of successive repeat of the velocity and position update equations. The Figure 3.2 illustrated the pseudocode for the basic PSO algorithm.

```
Create and initialize a D-dimensional PSO with s particles
Initialize the velocities of all particles
Initialize the personal best position of the particle i, pi
Initialize the global best position, pg (gbest model)

repeat
    Update velocity for continuous and binary variables
    Update position for continuous and binary variables

    for i = 1 to number of individuals particles
        if f(xi)<f(pi) then do    % update personal best position
            pi = xi
        end
        if f(pi)<f(pg) then do    % update global best position
            pg = pi
        end
    end
until stopping criterion satisfied
```

Figure 3.2 Basic pseudocode for the PSO algorithm

The stopping criterion depends on the type of problem being solved. Usually the algorithm is run for a fixed number of function evaluations or until a specified error bound is reached.

4 SOLUTION APPROACH. SINGLE DISTILLATION COLUMN DESIGN

The first stage in a mathematical programming synthesis approach consists on developing a superstructure that includes all the alternatives of interest. In this work, the superstructure used for the column design is based on the superstructure developed by *Yeomans and Grossmann (2000)*, in which the non–existing trays are considered as simple bypasses of liquid and vapor flows.

4.1 Problem Statement

For the sake of simplicity, but without loss of generality, let us consider the optimization of a single conventional distillation column, with one feed and two products streams, the distillate and bottom. Thus, the problem can be stated as follows: given a feed with known composition, determine the optimal configuration (feed location and total number of trays), and the optimal operating conditions (e.g. distillate flow rate, molar ratio of distillate to feed, reflux ratio, boilup ratio,…) for separating the feed into two product streams within given specifications to minimize the total annualized cost of equipment and utilities. In next section, we will extend the method to more complex systems.

It is worth nothing that not all of the operating conditions are independent and it is only necessary to select as many design specifications as degrees of freedom the systems have.

4.2 Objective Function: Total Annualized Cost (TAC).

As stated previously, the objective function to minimize comprises the annualized investment cost, or capital cost of the main items (column shell, trays and heat exchangers) and the most relevant operating cost (vapor steam and cooling water). The estimation of the capital costs, which depends on the column diameter, total number of trays and heat exchanger areas, where obtained by means of the nonlinear cost correlations given by

Turton, R. et al. (2002). Detailed data about this model can be found in Appendix A. On the other hand, the operating costs reflected in the objective function where calculated from the corresponding heat loads of the reboiler and condenser, using the steam and cooling utility costs given by *Turton, R.*, (see Appendix A).Thus, the objective function can be stated as:

$$\min TAC\left(\$/\text{yr.}\right) = C_{op} + F \times C_{cap} \tag{4.1}$$

where C_{op} is the operation cost and C_{cap} is the total cost of installed equipment, both updated by the CEPCI cost index (see Appendix A). The annualization factor of the capital cost *F* was calculated by the equation (4.2) recommended by *Smith, R., (2005)*, and takes into account the fractional interest rate per year (*i*) and the years over which the capital is to be annualized (*n*). Typical values are a fixed rate of interest of 10% and an annualization period of 5 years. It is worth mentioning that changing the annualization period can lead to different optimal column configurations due to the trade-off between the capital and operation costs.

$$F = \frac{i\left(1+i\right)^{n}}{\left(1+i\right)^{n} - 1} \tag{4.2}$$

4.3 Single Distillation Column Superstructure

The basic idea is to consider a conventional distillation column as a set of permanent trays among them are the feed tray, the condenser and the reboiler; and a set of conditional trays above and below the feed that can either exist or not (see superstructure proposed by *Yeomans and Grossmann (2000)* presented above, and repeated here for the sake of clarity, Figure 4.1).

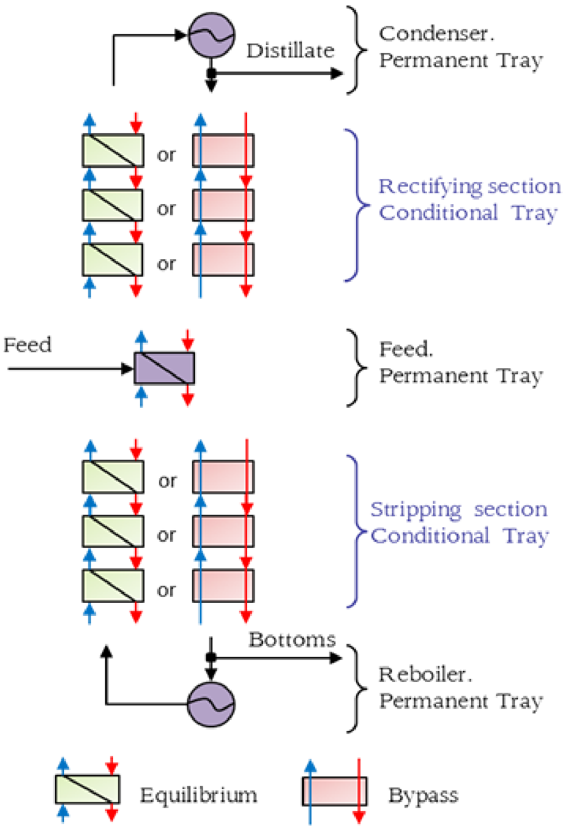


Figure 4.1 Superstructure of Yeomans and Grossmann for the optimal feed location, total number of trays and optimal operation conditions.

For a desired separation the number and distribution of the trays above and below the feed (rectification and striping sections, respectively) is a function of the relative volatilities and composition of the feed. Therefore, initially the number of conditional trays in rectification and striping sections must be larger than the minimum needed to ensure the desired purity of the products, with a view to provide an upper bound to the optimal number of trays. After the optimization, the optimal number of trays and feed location will be defined by the number of active trays in each section.

In a process simulator like Aspen Hysys, it is possible to generate the previously described superstructure using a built-in distillation column module, which includes tailored numerical methods developed for the convergence of these units. The existence or non-existence of the conditional trays (equilibrium stages) can be performed by forcing the trays to behave as a simple bypass of liquid and vapor flows, without mass or heat transfer, simply by fixing the Murphree tray efficiency to zero for the inactive trays (*Caballero et al., 2005*). As can be seen in equation (4.3), the Murphree efficiency is calculated from the vapor mole fraction of the vapors leaving tray n and $n+1$, y_n and y_{n+1} respectively; and the composition of vapor in equilibrium with the liquid leaving the tray n, y_n^*.

$$E_{MV} = \frac{y_n - y_{n+1}}{y_n^* - y_{n+1}} \qquad 0 \text{ £ } E_{MV} \text{ £ } 1 \tag{4.3}$$

Note that equilibrium equations are trivially satisfied in those trays in which the efficiency is set to zero $\left(y_n^* = y_n\right)$.

To avoid equivalent solutions in each section of the column the active trays should be consecutive. Therefore, to prevent “empty spaces” between active trays in the superstructure, we will follow the next criteria: in the rectification section, if a given tray exists, then all trays below it, until the feed tray must exist. And in the stripping section, if a given tray exists, then all trays above it must exist (see Figure 4.2). The index of the existing top ends in the rectification and striping sections are defined by the following two integer variables: N_R and N_S, respectively, where $1 \le NR \le \overline{NR}$ and $\underline{NS} \le NS \le NT$ (NT is the total trays of the column). Therefore, in the optimization process, the PSO will only have to handle these two integer variables to vary the total number of trays and feed tray location of the column.

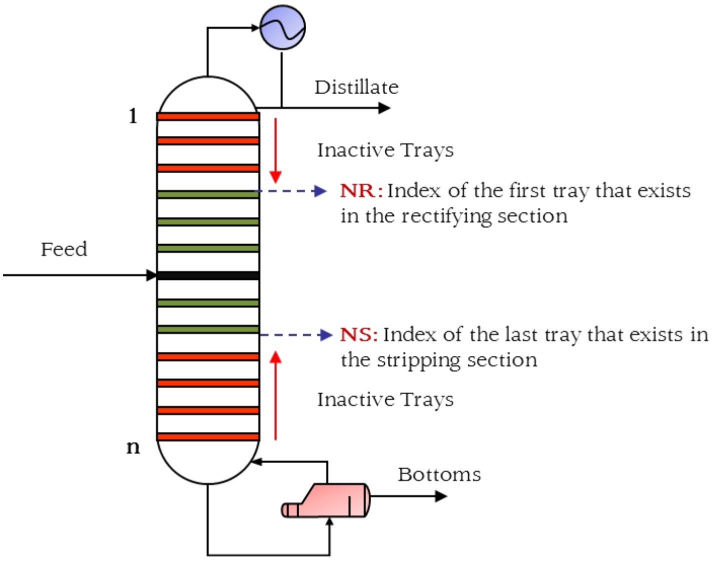

Distillate
1
Inactive Trays
NR: Index of the first tray that exists in the rectifying section
Feed
NS: Index of the last tray that exists in the stripping section
Inactive Trays
n
Bottoms

Figure 4.2 Scheme of the proposed criteria to avoid equivalent solutions

4.4 Design and Process Specifications

On the other hand, once the superstructure is defined, and all the basic design decisions required by the distillation column module are selected in the simulator environment (i.e. selection of the thermodynamic model, feed specification and column pressures), there are different ways for managing the remaining degrees of freedom of the column, that corresponds to the design/operation variables (continuous variables).

Hysys allows us to select as many operating conditions as degrees of freedom in the column taken from a large list of "column specifications". It is worth to nothing that for the case of a conventional distillation column, with a known feed and two products streams, once the operating pressure is fixed there are only two degrees of freedom.

If we define as design variables the purity specification or recovery of the key components required in the product flows, the problem will be completely defined. Therefore, it is not necessary to include any external restriction of purity or recovery. In any other case, if two operating conditions from the column are selected, as the reflux ratio and boilup ratio, it will be necessary to add other extra constraints that consider the specifications of pureness or recovery initially imposed to the product flows.

This can be performed in the simulator environment using an Adjust Operator that, for example, varies the value of the reflux ratio (independent variable) to meet the required value of purity of the light key component (dependent variable) in the distillate; and another one, that varies the value of the boilup ratio to meet the required purity of the heavy key component in the bottom product.

Another alternative would be including these constraints in the PSO. However, although the PSO technique has proven to be efficient for solving non-convex optimization problems (*Kennedy, J. & Eberhart, R., 1995*), the original PSO it is not so successful in solving constrained problems, because the algorithm handles the constrains penalizing infeasible solutions. Therefore, it would be necessary to modify the objective function by adding a penalty term which considers the deviation between the desired purity or recovery specifications in the product streams, $X_{spc,\ i}$, and the obtained value on every iteration of the process x_i. So that, when a constraint is violated it will appear positive contribution in the objective function, as it is shown in equation (4.4).

$$\min TAC = C_{op} \cdot t_{year} + f_c \cdot C_{cap} + \sum_i w_i \left| \left(X_{spc,i} - x_i \right) \right| \tag{4.4}$$

where w_i is a positive penalty parameter of the same magnitude as the costs.

That is why it is advisable, whenever possible, to select as design variables the specifications of purity or recovery required in the distillate and bottoms products. In this way, the PSO does not have to handle the continuous variables, and we do not have to introduce any penalty term in the objective function. However, that is not always possible. In some systems of distillation columns with recycle streams, which convergence by itself (without the optimization process) is quite complicated, it is better to select design variables that result in more robust flowhseets, such as the reflux and boilup ratio, with the aim of making an optimization stage easier. If we need to add any other constraint, such as temperature bounds for security or stability reasons for example, a penalty terms in the objective function should be used.

It is worth remarking that we are using an exact penalty function. In that way we ensure that the solution of the original problem (without penalties) and the reformulated problem is the same if the penalty term is large enough, and at the same time the magnitude of the penalty is not too large (larger than or equal to the Lagrange multiplier related to the constraint).

4.5 Optimization Algorithm with Embedded Process Simulator

A scheme of the proposed approach is shown in Figure 4.3. The external solver (PSO), as well as the objective function and all other auxiliary files were implemented in Matlab. An executive program controls all these files and established the connection with the process simulator, Aspen Hysys. We use the binary-interface standard component object model (COM), by Microsoft, to interact with Aspen Hysys through the objects exposed by the developers of the process simulators. We utilize Matlab as an automation client to access these objects and interact with Aspen Hysys, which works as an automation server.

The next step of the proposed algorithm is the initialization of the PSO parameters. Then, the values of the indices of the top active trays of each column section (NR and NS) are converted in the Matlab environment to a vector of ones and zeros according to the existence or not of the trays, respectively. This information is sent to the built-in distillation column module as a vector of Murphree efficiencies. At this point, the distillation column is automatically updated and converged, and the process simulator returns all the dependent variables needed for calculating the total annual cost (TAC). Usually, the algorithm runs until a specified stopping criterion is reached for instance, until the objective function of all the particles in the PSO has collapsed under a specified tolerance or for a fixed number of function evaluations.

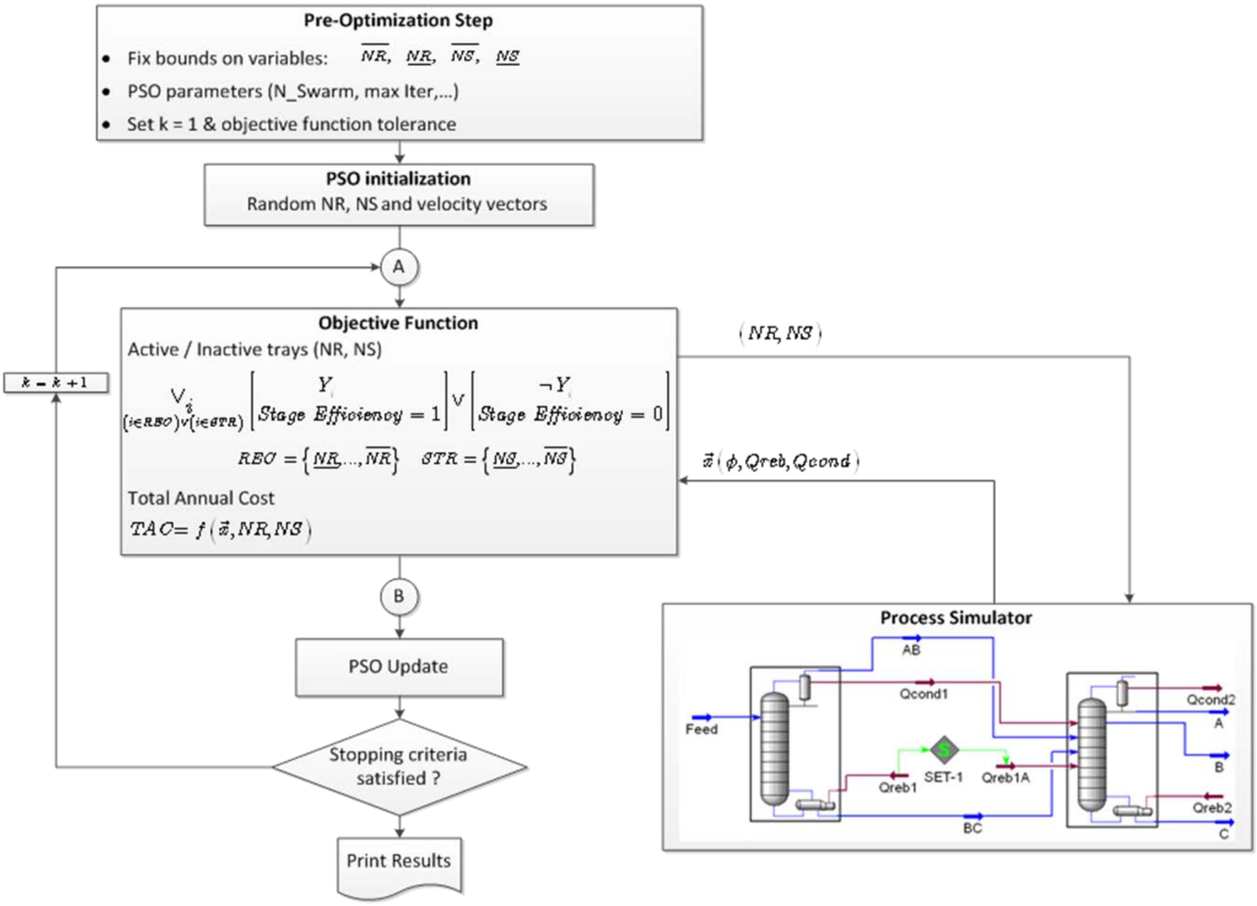

Pre-Optimization Step
Fix bounds on variables: $\overline{NR}$, $\underline{NR}$, $\overline{NS}$, $\underline{NS}$
PSO parameters (N_Swarm, max Iter,...)
Set k = 1 & objective function tolerance
PSO initialization
Random NR, NS and velocity vectors
A
$k = k + 1$
Objective Function
Active / Inactive trays (NR, NS)
$\bigvee_{i\,(i\in REC)\vee(i\in STR)} \begin{bmatrix} Y_i \\ Stage\ Efficiency = 1 \end{bmatrix} \vee \begin{bmatrix} \neg Y_i \\ Stage\ Efficiency = 0 \end{bmatrix}$
$REC = \{\underline{NR},\dots,\overline{NR}\}$ $STR = \{\underline{NS},\dots,\overline{NS}\}$
Total Annual Cost
$TAC = f(\vec{x}, NR, NS)$
(NR, NS)
$\vec{x}(\phi, Qreb, Qcond)$
B
PSO Update
Stopping criteria satisfied ?
Print Results
Process Simulator
AB
Qcond1
Feed
SET-1
Qreb1
Qreb1A
BC
Qcond2
A
B
Qreb2
C

Figure 4.3 Scheme of the proposed algorithm.

5 NUMERICAL EXAMPLES

Three case studies are presented to illustrate the methodology and how to proceed with the proposed method. The first example is the optimization of the single distillation column; afterward, two more complex examples are constructed based on the first example. For all the cases, the first stage in our methodology involves developing a superstructure that covers all the interesting alternatives. In this work, we use the superstructure developed by Yeomans and Grossmann, in which the nonexisting trays are considered as simple bypasses of liquid and vapor flows. We use a population size of 20 particles for all the examples. This key PSO parameter was tuned after a set of computational experiments varying the population size from 5 to 200 particles to show the trade-off between the objective function value and CPU time. The results show that a population size higher than 20 increases considerably the computational time but only yields a negligible improvement of the optimum value.

All the examples were solved on a computer with a 1.66 GHz Intel Core Duo processor and 1 GB of RAM.

5.1 Single Distillation Column

Let us consider the optimization of a single conventional distillation column, with one feed and two products streams: the distillate and bottom. The problem can be stated as follows: given a feed of known composition, determine the optimal configuration (feed location and total number of trays) and the optimal operating conditions (e.g., distillate flow rate, molar ratio of distillate to feed, reflux ratio, boilup ratio, ...) for separating the feed into two product streams within given specifications and needed to minimize the total annualized cost of equipment and utilities. The feed for this example is a multi-component mixture of hydrocarbons from c-4 to c-6, and the objective is to obtain the c-4 hydrocarbons as top products with the minimum Total Annual Cost (TAC). The molar flow rate and composition of the feed, and other data for the problem are shown in Table 5.1. We assume that the maximum heavy impurity in the top product stream leaving the column (isopentane) must be 0.5 mol%, and that the light impurity in the residue (cyclobutane) must also be 0.5 mol%. These constrains can be treated as specifications, design (independent) variables, in the process simulator or as external constraints as mentioned in Section 4. In that last case, two new column specifications must be chosen, for instance, the reflux ration (RR) and the boilup ratio (BR), which the process simulator will attempt to adjust in such a way that the desired purities (added as external constraints) are achieved. Bear in mind that a conventional distillation column has two degrees of freedom once the feed, feed tray location, pressure and number of trays have been fixed.

In this example and the following, unless otherwise stated, the product specifications are treated as column specifications in the simulator environment.

A superstructure with an upper bound of 60 trays is initially considered (condenser and reboiler are not included). The feed tray is placed in the tray 30 numbering from the top to bottom. The number of conditional trays in each column section was 25, as shown in Figure 5.1. We have also specified a minimum number of 10 permanent trays in the middle of the column (including the feed tray). Although this is not strictly necessary, it helps in the optimization search procedure reducing the number of alternatives. For the case where the optimal solution lies at one of these limits, the upper bound of trays is increased or the minimum number of permanent trays is decreased, or even both.

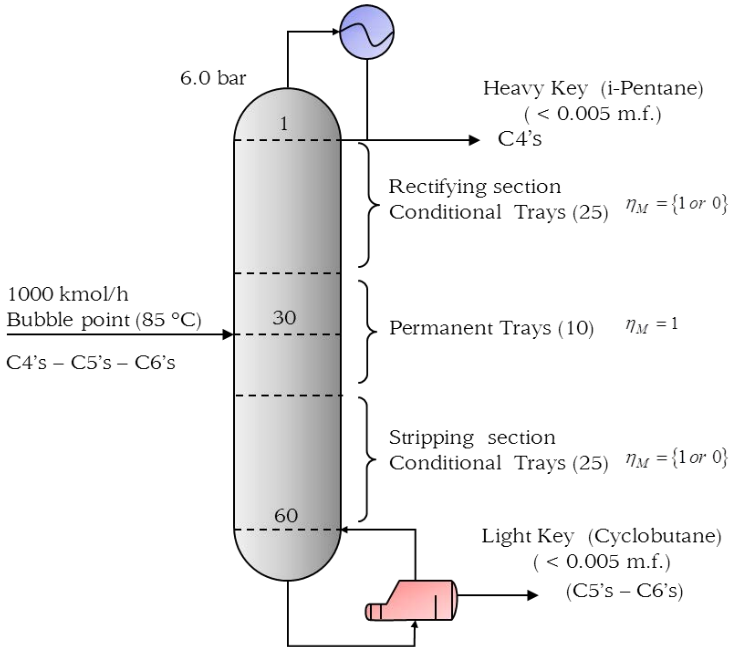


Figure 5.1 Single Distillation Column Superstructure

Table 5.1 Data for Examples

Heat Exchangers		Distillation Columns	
Condenser: U = 800 W/(m^2 K) Kettle reboiler U = 820 W/(m^2 K) Material of construction: carbon steel (shell and tubes)		Calculation based on sieve tray (one pass) Material of construction: carbon steel (sieves and tower) Tray spacing, d, 0.609 m Column height H(m) = 3 + NT · d Tray Sizing based on design limit: flooding (85%)	

Utility Costs	
Low pressure steam (5 barg, 160 ºC)	7.78 $/GJ
Medium pressure steam (10 barg, 184 ºC)	8.22 $/GJ
High pressure steam (41 barg, 254 ºC)	9,83 $/GJ
Cooling water (30 to 45 ºC)	0.354 $/GJ
Electricity	60.0 $/MWh
(Calculation based on 8000 h/yr. of operation)	

Example 1	
Feed	1000 kmol/h (85 ºC)
Composition (mole fraction)	
i-butane	
n-butane	0.17
cyclobutane	0.12
i-pentane	0.06
n-pentane	0.13
cyclopentane	0.09
2-methyl pentane (isohexane)	0.07
n-hexane	0.09
cyclohexane	0.15
pressure	0.12
thermodynamics (fluid package)	600 kPa
specifications	Soave-Redlich-Kwong
molar fraction isopentane (heavy key comp) in distillate	
molar fraction cyclobutane (light key comp) in bottoms	≤ 0.005
	≤ 0.005

Example 2	
Feed	540 kmol/h (47 ºC)
Composition (mole fraction)	
Acetone	
Methanol	0.50
thermodynamics (fluid package)	0.50
specifications	UNIQUAC
Extractive Column	
pressure	
molar fraction acetone in distillate	101 kPa
acetone recovery	≥ 0.9999
Entrainer-Recovery Column	≥ 99.95 %
Pressure	
molar fraction methanol in distillate	101 kPa
molar fraction DMSO in bottoms	≥ 0.9995
	≥ 0.9999

Example 3	
Feed	500 kmol/h (111 ºC)
Composition (mole fraction)	
Benzene	
Toluene	0.30
p-Xylene	0.40
thermodynamics (fluid package)	0.30
specifications	Soave-Redlich-Kwong
Prefractionator Column	
pressure	
benzene recovery	
p-xylene recovery	120 kPa
Main Column	≥ 99.95 %
Pressure	≥ 99.95 %
molar fraction benzene in distillate	
molar fraction toluene in side stream	101 kPa
molar fraction p-xylene in bottoms	≥ 0.999
	≥ 0.999
	≥ 0.999

When presenting the optimization results, it must be taken into account that the Particle Swarm Optimization algorithm is a stochastic optimization method, and hence the convergence to the same solution is not always guaranteed. In addition, although usually the distillation columns have a single global optimal solution, there may be several solutions near the best one (e.g. given a distillation column with n trays, a similar structure but with some trays more – which means a greater capital cost –, can result in a similar objective function, since the energy consumption is decreased. Here the annualization factor, F, has an important effect (see Equation 4.1)). For that reason, one way to prove the performance of the proposed optimization approach is to run the algorithm for a certain number of times and analyze the number of times that the algorithm converge to the best solution founded in all executions.

The main results of twenty consecutive executions of the optimization algorithm are summarized in Table 5.2. As can be seen, there are four different configurations of the distillation column. All of them are very close not only in the structure, but also in the value of the objective function. Between them, the configuration with the best objective function, and also the most repeated (70 %) is the distillation column with 45 trays and feed tray in 21 (NR = 10, NS = 54). It is worth to mention that for a given solution, the small differences in the value of the objective function (TAC) are consequence of the intrinsic numerical noise of the process simulator.

Table 5.2 Results of 20 consecutive executions of the PSO algorithm – Example 1

Execution	Total Trays	Feed Tray	NR	NS	TAC (k$/yr.)	CPU time (s)	Stopping Criterion
1	45	21	10	54	1803.391	30	Criterion 2
2	45	21	10	54	1802.743	25	Criterion 2
3	46	21	10	55	1803.329	38	Criterion 2
4	45	21	10	54	1802.750	22	Criterion 2
5	45	21	10	54	1802.743	63	Criterion 2
6	45	21	10	54	1802.744	25	Criterion 2
7	44	20	11	54	1803.074	30	Criterion

							2
8	46	21	10	55	1803.414	41	Criterion 2
9	**45**	**21**	**10**	**54**	**1802.737**	**23**	Criterion 2
10	45	21	10	54	1803.458	17	Criterion 2
11	45	21	10	54	1803.261	25	Criterion 2
12	45	21	10	54	1803.304	25	Criterion 2
13	47	22	9	55	1802.981	68	Criterion 2
14	45	21	10	54	1803.194	20	Criterion 2
15	45	21	10	54	1802.742	59	Criterion 2
16	45	21	10	54	1802.743	31	Criterion 2
17	45	21	10	54	1803.459	28	Criterion 2
18	45	21	10	54	1803.457	17	Criterion 2
19	44	20	11	54	1802.864	18	Criterion 2
20	46	21	10	55	1803.405	28	Criterion 2

Stopping Criterion 1: stop due to maximum number of iterations is reached (50 major iterations).
Stopping Criterion 2: the tolerance between best and worse particle is under specification ($1 \cdot 10^{-5}$).

The optimal design characteristics and computational results for the best configuration found by the algorithm are shown in Figure 5.2 and Table 5.3.

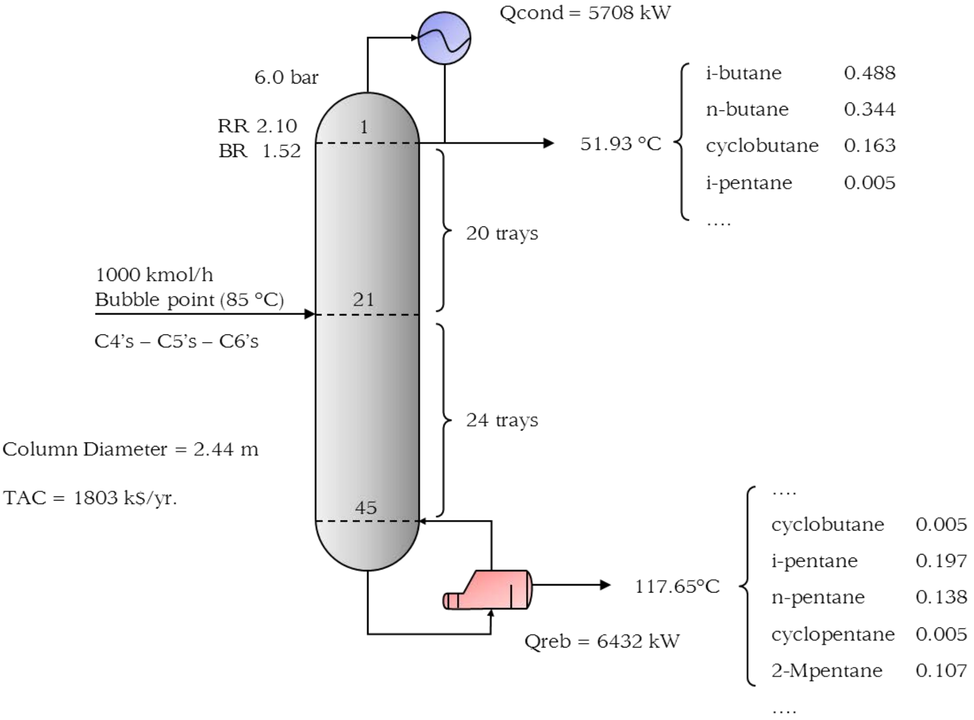


Figure 5.2 Best Solution of Example 1

Table 5.3 Computational Results for Best Solution Example 1

PSO Description	
Number of Particles	20
Major Iterations	18
Function Evaluations	380
Discrete Variables	2
Stopping Criterion	The tolerance between best and worse particles is under specification
CPU Time (s)	23
Optimal Solution	
TAC (k$/yr.)	1802.74
Capital Cost (k$)	1150.3
Operating Cost (k$/yr.)	1499.3

5.2 Extractive Distillation System

The second case study involves the optimization of an extractive distillation process. This sort of assisted distillation is commonly used to separate close boiling or homogenous binary azeotropes by adding a higher-boiling component, the so-called entrainer. This new component facilitates the separation by interacting with the original mixture and attracting one of the components. The proposed case study is illustrated with the separation of an isomolar mixture of acetone and methanol, using dimethyl sulfur oxide (DMSO) as entrainer. For more detailed data about extractive distillation and entrainer selection see *Doherty & Malone* (*2001*) and *Kossack et al.* (*2008*).

This system has the following properties: the acetone/methanol mixture has a binary homogeneous azeotrope with a composition of 77.6 mol% acetone at atmospheric pressure, as shown in Figure 5.4a. The normal boiling points of acetone and methanol are 329 and 338 K, respectively; and the normal boiling point of the entrainer DMSO is 464 K. DMSO is much higher boiling than either acetone or methanol, and it is a very effective solvent. So it is possible to attain high product purities. Figure 5.3b gives the Txy diagrams for acetone/DMSO and methanol/DMSO and shows that both of these separations should be easy.

The classical extractive distillation system comprises a set of two distillation columns: the extractive column, which has two feeds, and the entrainer-recovery column, represented in Figure 5.4. The entrainer is fed into the extractive column above the process feed. One of the original components, the acetone, is obtained at the top of the extractive column, while the methanol, together with the entrainer (DMSO), forms the bottoms product. In the second column, the entrainer is separated from the methanol and recycled back to the first column. It is worth mentioning that different entrainers have different effects on the azeotropic mixture. For example, the chlorobenzene entrainer (T_b = 404 K) drives the methanol overhead in the extractive column.

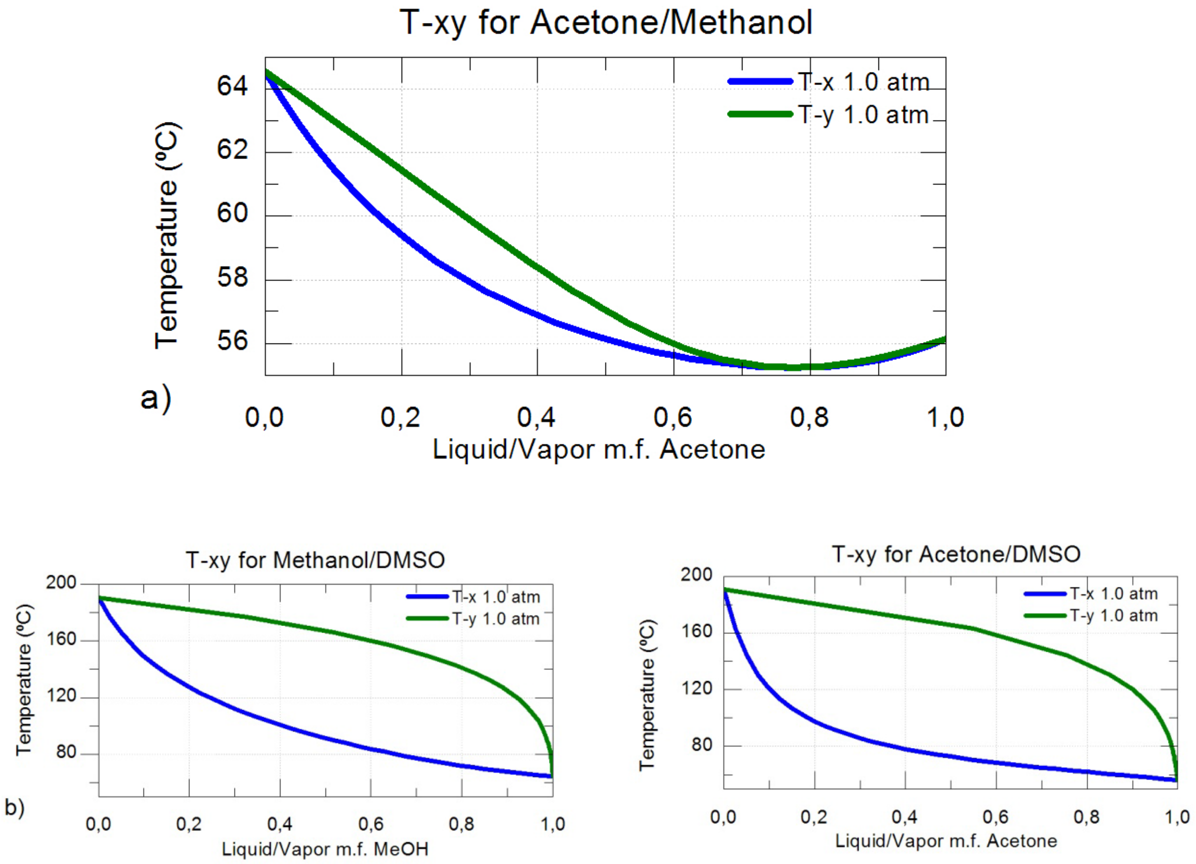


Figure 5.3 a) T-xy diagram for Acetone/Methanol. b) T-xy diagrams for Methanol/DMSO and Acetone/DMSO

Figure 5.4 shows the representation of the proposed superstructure for the optimization of the acetone/methanol extractive distillation process with DMSO. As can be seen, the extractive column has two different feeds that divide the column in three different sections with conditional trays. Note that in this case the relative position of both feeds are fixed (the feed rich in the entrainer is above the acetone/methanol feed). On the other hand, the solvent-recovery column is a conventional column, and the superstructure is basically the same as in the previous example.

In the optimization of this system we must take into account that the entrainer flowrate it is also a variable to optimize that has a direct influence on the reflux. In addition the recycle stream poses an even more difficult challenge.

The upper bound for the number of trays of each column, and the conditional trays of all the columns sections are shown in Figure 5.4. As in the previous examples we have also specified a minimum number of permanent trays in addition to the feed trays, condensers and reboilers. The molar fraction required for the acetone is ≥0.9995 in the overhead of the first column for a minimum recovery of 99.95 %. The methanol molar fraction, top product of the second column, must be greater than 0.9995, and the entrainer must be recovered with a minimum purity of 99.99%. These constraints are stated in the simulator environment as the column specifications. The remaining data for example 2 are shown in Table 5.1.

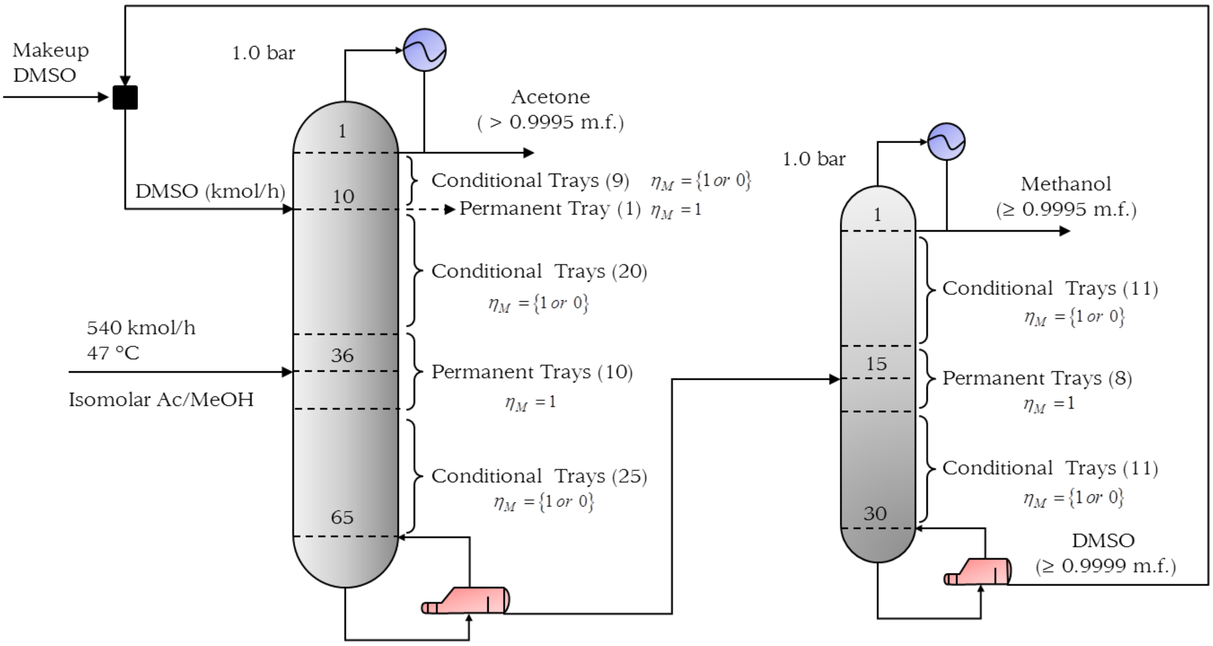


Figure 5.4 Acetone/Methanol Extractive Distillation with DMSO Solvent Superstructure

In Table 5.4 are summarized the big numbers of twenty consecutive executions of the proposed approach. Notice that there are four different configurations of the extractive column (one with 45 trays, two with 46 and one with 47 trays), and only one configuration of the entrainer–recovery column with a total number of 12 trays. The minimal objective function (TAC = 3042.170 k$/yr.) corresponds to one of the configurations with 46 trays in the extractive column. All configurations and objective function values are very close. In Figure 5.5 and Table 5.5 are presented the design and computational results of the best solution founded by the algorithm which is also the most repeated (75%).

Table 5.4 Results of 20 consecutive executions of the PSO algorithm – Example 2

Extractive Column							Entrainer – Recovery Column							
Execution	Total Trays	DMSO Feed Tray	Feed Tray	NR	NM	NS	Total Trays	Feed Tray	NR	NS	DMSO flow rate (kmol/h)	TAC (k$/yr.)	CPU time (s)	Stopping Criterion
1	46	3	29	8	13	55	12	5	11	22	488	3042,407	143	Criterion 1
2	46	4	29	7	14	55	12	5	11	22	488	3042,378	90	Criterion 2
3	46	4	29	7	14	55	12	5	11	22	488	3042,415	87	Criterion 2
4	46	4	29	7	14	55	12	5	11	22	488	3042,406	110	Criterion 2
5	46	4	29	7	14	55	12	5	11	22	488	3042,378	89	Criterion 2
6	47	4	30	7	13	55	12	5	11	22	486	3042,757	134	Criterion 1
7	46	4	29	7	14	55	12	5	11	22	488	3042,415	88	Criterion 2
8	45	4	29	7	14	54	12	5	11	22	489	3042,274	96	Criterion 2
9	46	4	29	7	14	55	12	5	11	22	488	3042,410	116	Criterion 2
10	46	4	29	7	14	55	12	5	11	22	488	3042,413	113	Criterion 2
11	**46**	**4**	**29**	**7**	**14**	**55**	**12**	**5**	**11**	**22**	**488**	**3042,170**	**91**	**Criterion 2**
12	46	4	30	7	14	55	12	5	11	22	488	3042,414	68	Criterion 2
13	47	4	28	7	13	55	12	5	11	22	487	3042,652	69	Criterion 2
14	45	4	29	7	15	55	12	5	11	22	489	3042,838	133	Criterion 1
15	46	4	29	7	14	55	12	5	11	22	488	3042,378	108	Criterion 2
16	46	4	29	7	14	55	12	5	11	22	488	3042,415	95	Criterion 2
17	46	4	29	7	14	55	12	5	11	22	488	3042,347	115	Criterion 2
18	46	4	29	7	14	55	12	5	11	22	488	3042,394	135	Criterion 1
19	46	4	29	7	14	55	12	5	11	22	488	3042,418	82	Criterion 2
20	46	4	29	7	14	55	12	5	11	22	488	3042,414	103	Criterion 2

Stopping Criterion 1: stop due to maximum number of iterations is reached (70 major iterations)

Stopping Criterion 2: the tolerance between best and worse particle is under specification ($1\cdot10^{-5}$).

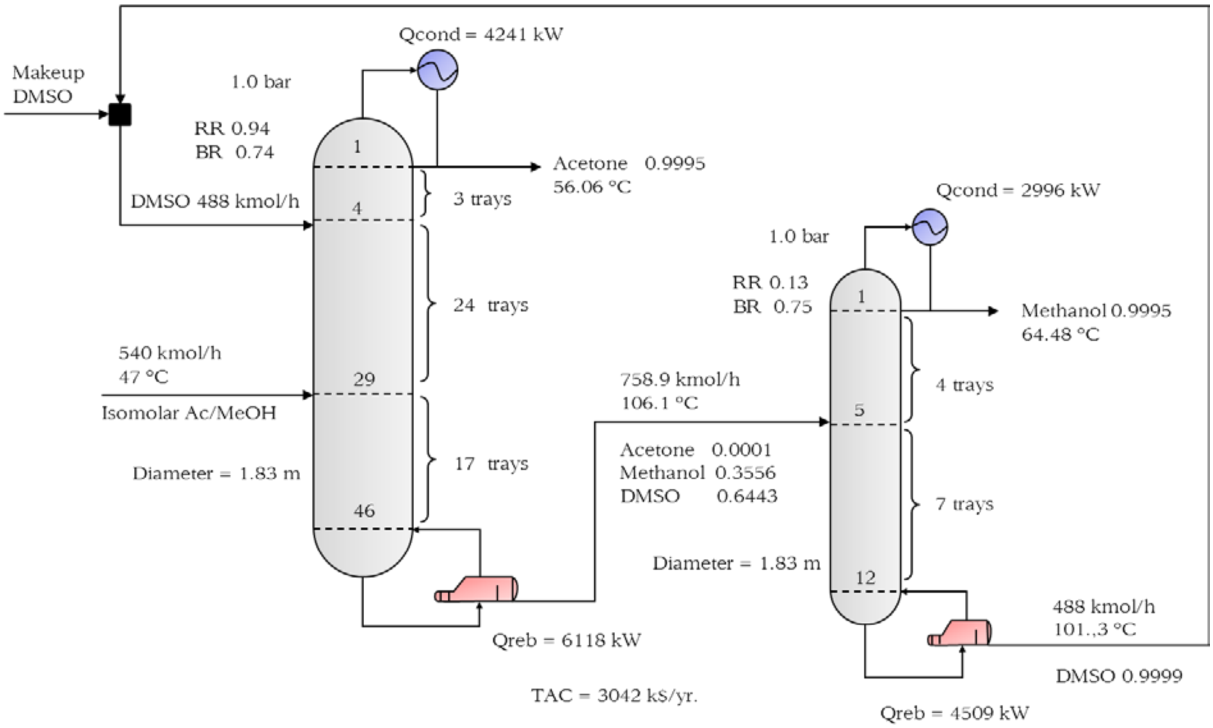


Figure 5.5 Best solution of Example 2

Table 5.5 Computational Results for Best Solution Example 2

PSO Description	
Number of Particles	20
Major Iterations	49
Function Evaluations	1000
Discrete variables	5
Continuous variables	1
Stopping Criterion	The tolerance between best and worse particles is under specification
CPU Time (s)	91
Optimal Solution	
TAC (k$/yr.)	3042.17
Capital Cost (k$)	1217.0
Operating Cost (k$/yr.)	2721.1

5.3 Divided Wall Column

The objective of this example is to optimize a fully thermally coupled distillation system, or Petlyuk column (*Petlyuk, F. B., et al., 1965*), for separating a three component mixture.

The design of thermally coupled distillation systems has been considered with special interest in recent years because of their potential energy savings with respect to the use of conventional distillation sequences. Thermally coupled structures are developed by substituting vapor-liquid interconnections between two columns for a condenser or a reboiler of one of the columns as it is represented in Figure 5.6.

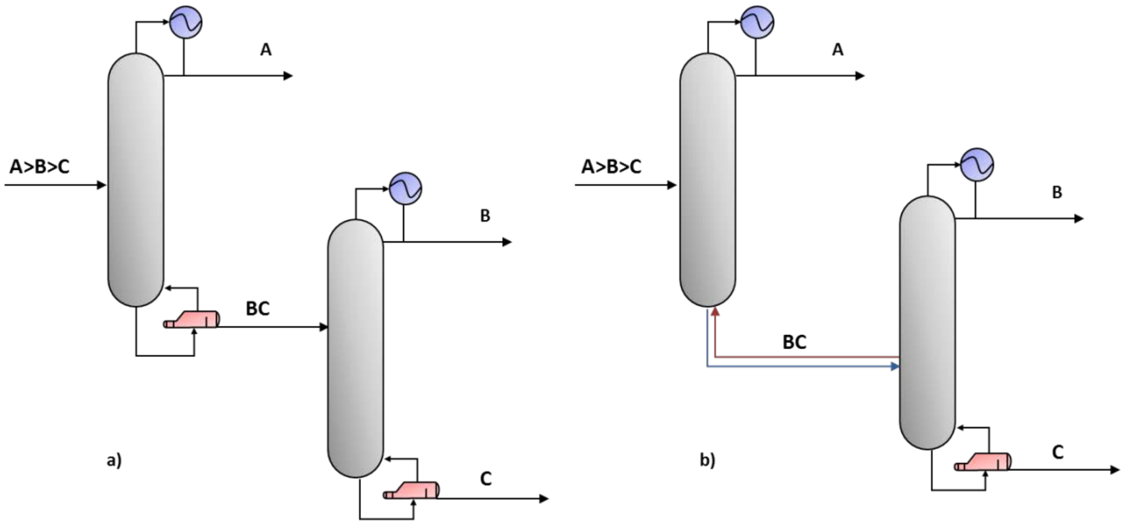


Figure 5.6 a) Direct sequence. B) Thermally coupled direct sequence (side-rectifier arrangement)

In this example the reboiler of the first column is replaced by a thermal couple. As can be seen, the liquid from the bottom of the first column is transferred to the second as before, but now the vapor required by the first column is supplied by the second column, instead of a reboiler on the first column.

To introduce the basis of the Petlyuk configuration it is interesting to make a short review of the different separation systems for obtain three fractions from an initial multicomponent mixture. For the sake of clarity, if we consider a total separation (or close to the total separation within specifications) of certain key components, we do not lose any generality if we consider that the objective is to separate a three-component mixture in three streams composed by each of the pure components.

If there is a three-component mixture (without azeotropes) to be separated into three relatively pure products (A-B-C, with A being the most volatile) and conventional columns (a single feed, two product streams, condenser and reboiler) are employed, then there are only two different alternatives, as represented in Figure 5.7. However, these sequences suffer from an inherent inefficiency, caused by the thermodynamic irreversibility associated with stream mixing at the feed, top, and bottom of the column (*Petlyuk, F. B., et al., 1965*). This inefficiency is intrinsic to any separation that involves an intermediate boiling component.

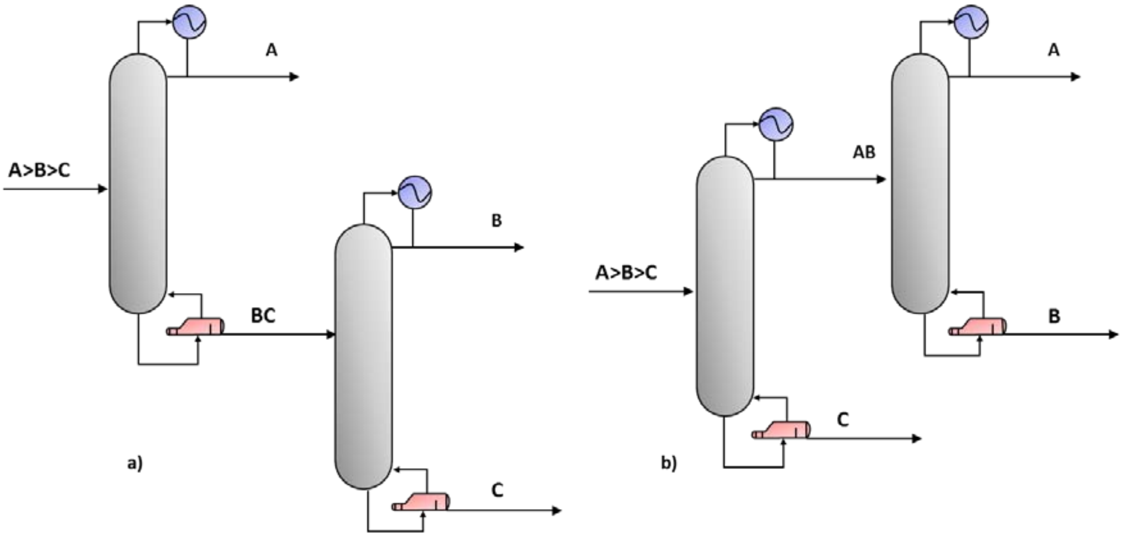


Figure 5.7 a) Direct and b) indirect sequences of conventional distillation columns for a three component separation

In conventional columns, see Figure 5.8, the composition of B (product of intermediate volatility) in the first column increases below the feed as the more volatile component A decreases. However, moving further down the column, the composition of B decreases again when the composition of the less-volatile component C increases. Therefore, the composition of B reaches a peak only to be remixed. This remixing is a source of inefficiency in the separation.

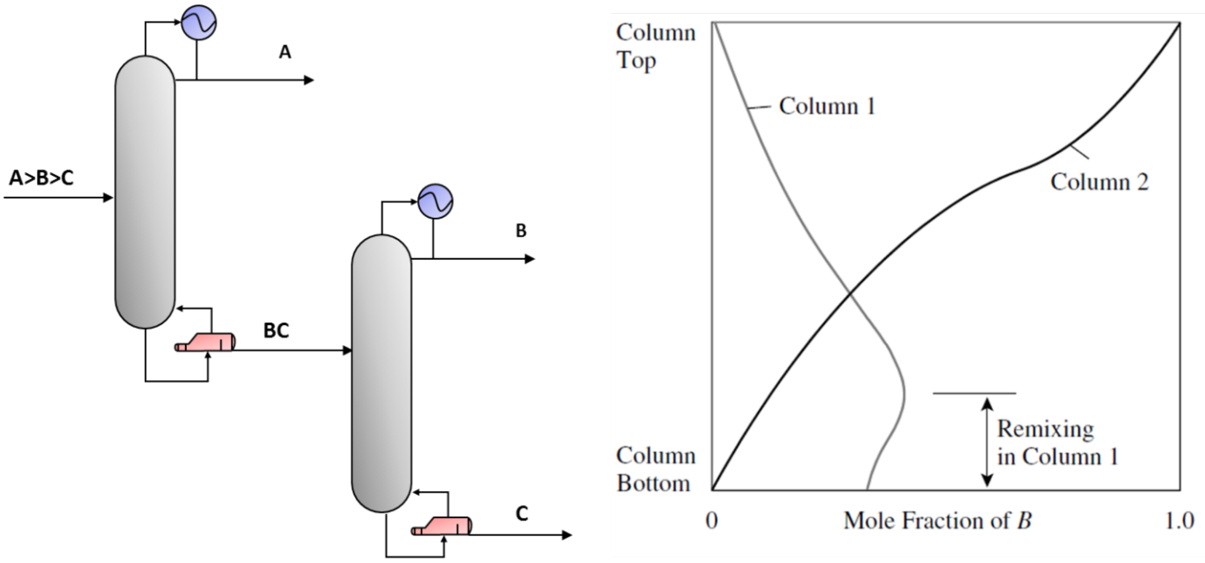


Figure 5.8 Typical composition profile for the intermediate volatility component in the distillation columns of the direct sequence

Another alternative is to consider the separation of the three components as in Figure 5.9a in which the lightest and heaviest components are chosen to be the key components, so that the intermediate volatility component is distributed in both products. This separation system is known as distributed distillation or *sloppy* distillation and needs one more distillation column than the direct/indirect sequences shown previously in Figure 5.7 to produce the three pure products. However, if the second and third columns in Figure 5.9a are operated at the same pressure, then the second and third columns could simply be connected and the middle product taken as a sidestream as shown in Figure 5.9b. This system is known as a *prefractionator* arrangement.

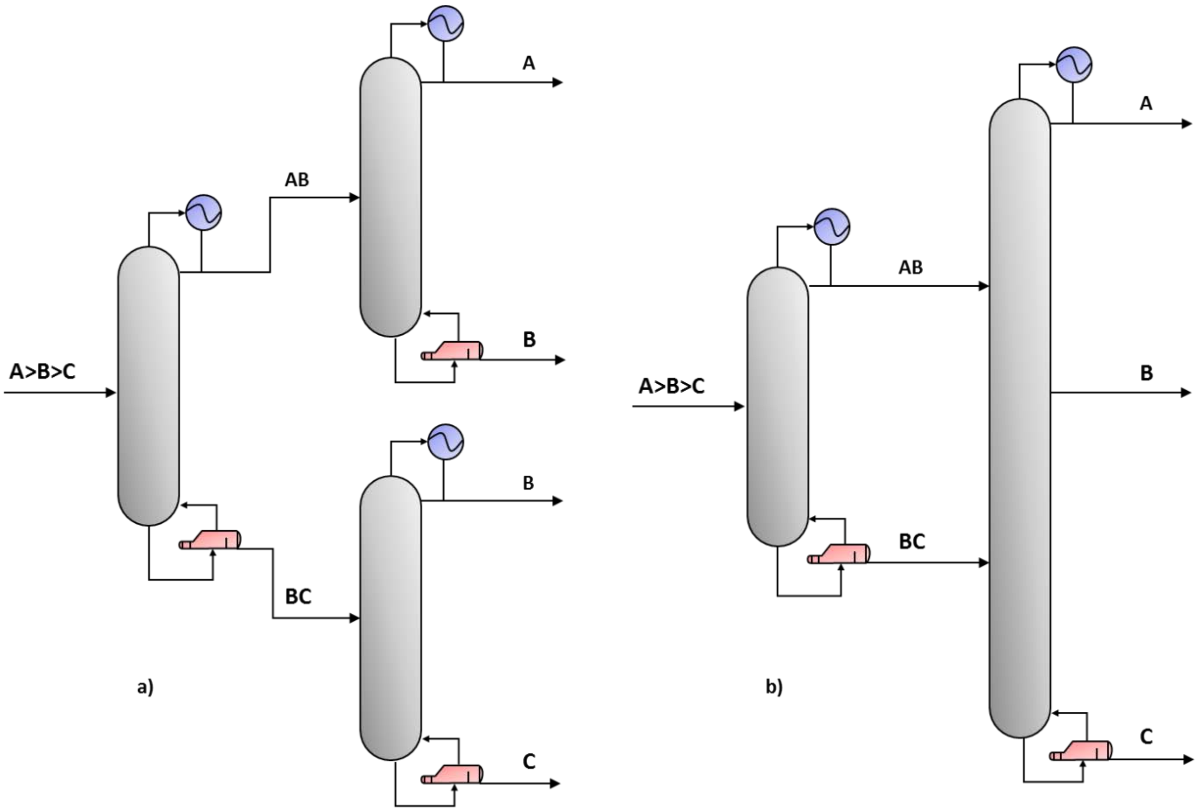


Figure 5.9 a) Distributed distillation. b) Prefractionator arrangement

At first sight, the systems in Figure 5.6 seem to be inefficient in the use of equipment. However, comparing the sloppy distillation and the prefractionator system with the conventional separation sequences shown in Figure 5.4, the distributed and prefractionator systems typically require 10 to 30% (*Fidkowski and Krolikowski, 1987*) less energy than conventional arrangements for the same separation. The reason for this difference is none other than the fact that the distributed distillation and prefractionator systems are fundamentally thermodynamically more efficient than a conventional sequence. As for the direct sequence, the composition profile for the component B is shown in Figure 5.10. As can be seen, the component B is distributed between the top and bottom of the column, so it is possible to achieve that the middle product (B) does not pass through a maximum, but is distributed smoothly across the column. In this way, the remixing effects that are a feature of both simple column sequences are avoided.

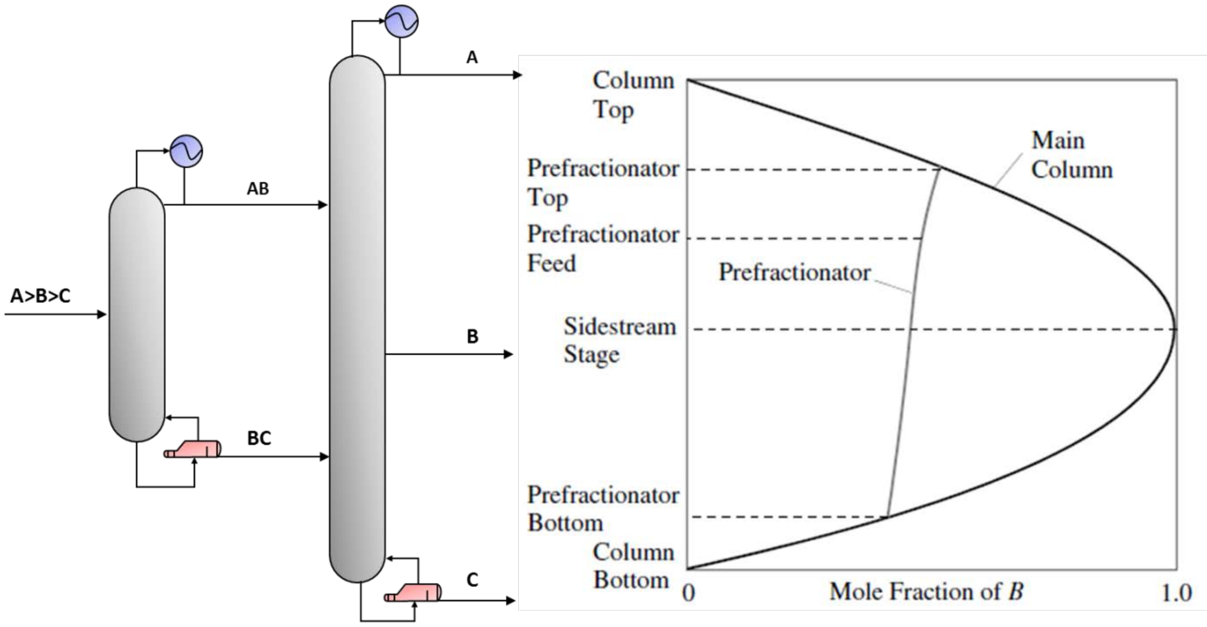


Figure 5.10 Typical Composition profile for the component B in the prefractionator system.

Furthermore, the contribution of the remixing due to the condenser and reboilers can be mitigated considering the thermal coupling of the prefractionator system. If the condenser of the first column is replaced by thermal coupling, the liquid reflux is obtained by a liquid side stream of the second column (usually from the same stage at which arrives the vapor stream from the first column, see Figure 5.11a). In the same way, if the reboiler is removed, the vapor required by the first column is supplied by the second one by means of a vapor side stream. This configuration is named the Petlyuk column, in honor of F. Petlyuk (*Petlyuk et al., 1965*).

The Petlyuk structure only includes two heat exchangers (one condenser and one reboiler) compared with the four heat exchangers of the direct/indirect sequence. Thus, it is possible to obtain savings in the investment costs. It is possible even go one step beyond, and integrate the two columns of the Petlyuk configuration in a single shell, divided by an internal wall.. This configuration is known as Dividing Wall Column (DWC), as it is shown in Figure 5.11. The configurations in Figure 5.11a and 5.11b are thermodynamically equivalents if there is no heat transfer across the partition wall.

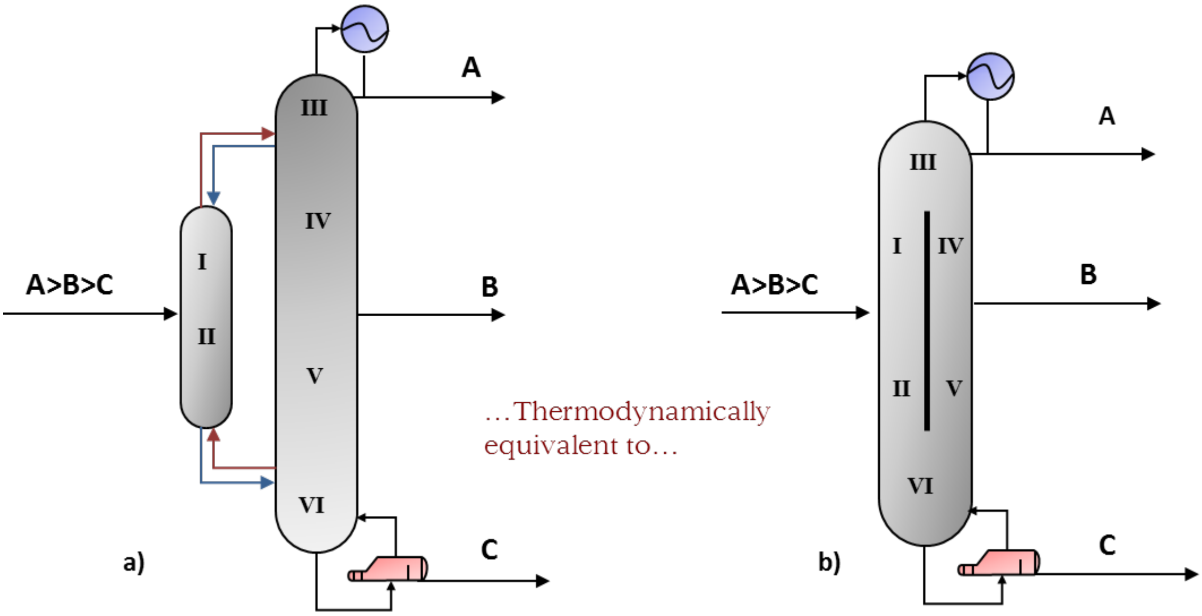


Figure 5.11 a) Fully thermally couple distillation system or Petlyuk configuration (1965). b) Divided Wall Column system

The simulation of a divided wall column using a process simulator is far from straightforward because of the liquid-vapor side streams connecting the two columns produce a flowsheet with a large number of recycle streams. Therefore, at each major iteration both columns must be converged with two undesirable side effects: a) the computation time for a single simulation considerably increases and, b) it is relatively easy that the system becomes prone to errors. In any case, if we try to use the PSO optimization algorithm, the systems should be robust and easy to converge, and this condition is completely lost when a large number of recycles are introduced.

To facilitate an easier simulation of the dividing wall column, we make use of the novel strategy for the simulation of thermally coupled distillation sequences developed by *Navarro, M. A. et al.,* (*2012*) from the Institute of Chemical Process System Engineering of the University of Alicante. For detailed data about this methodology see the Appendix B. Just mention herein that in order to avoid the recycle structure that appears in thermally coupled distillation columns it is possible to use two conventional distillation column modules replacing the material recycle streams by a combination of a material and energy streams.

Figure 5.12 shows the superstructure proposed for the optimization of a dividing wall column for separating a mixture of benzene, toluene and p-xylene in pure components (≥ 99.95 mol%). The upper bound for the number of trays of the first distillation column, that corresponds with sections I and II of the divided wall column was set in 70, and for the second column 122 (sections III to VI). The conditional and permanent trays of each column section, as well as the feed trays and the remaining parameters of the superstructure can be deduced from the Figure 5.12. All the required specifications of the divided wall column are listed in Table 5.1.

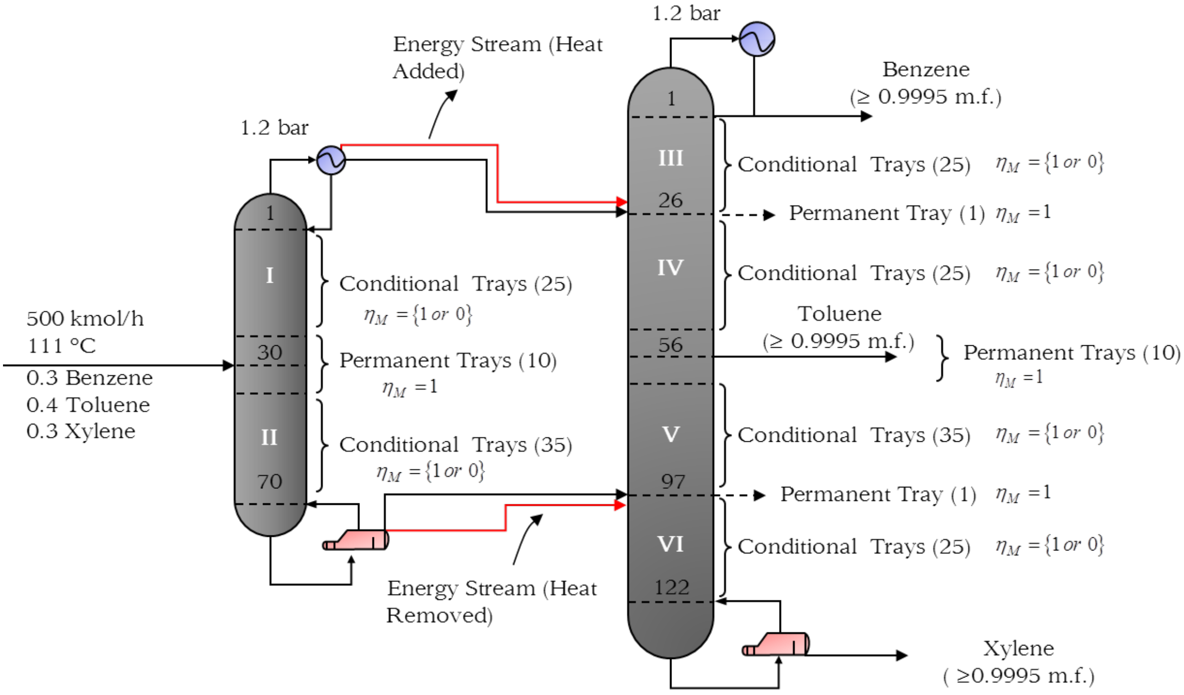

1.2 bar
Energy Stream (Heat Added)
1.2 bar
Benzene
(≥ 0.9995 m.f.)
1
III
26
IV
56
V
97
VI
122
Conditional Trays (25) $\eta_M = \{1 \; or \; 0\}$
Permanent Tray (1) $\eta_M = 1$
Conditional Trays (25) $\eta_M = \{1 \; or \; 0\}$
Toluene
(≥ 0.9995 m.f.)
Permanent Trays (10) $\eta_M = 1$
Conditional Trays (35) $\eta_M = \{1 \; or \; 0\}$
Permanent Tray (1) $\eta_M = 1$
Conditional Trays (25) $\eta_M = \{1 \; or \; 0\}$
1
I
30
II
70
Conditional Trays (25) $\eta_M = \{1 \; or \; 0\}$
Permanent Trays (10) $\eta_M = 1$
Conditional Trays (35) $\eta_M = \{1 \; or \; 0\}$
500 kmol/h
111 °C
0.3 Benzene
0.4 Toluene
0.3 Xylene
Energy Stream (Heat Removed)
Xylene
(≥0.9995 m.f.)

Figure 5.12 Divided Wall Column Superstructure (Acyclic System for simulation)

It should be noted that, although in the process simulator the divided wall column is performed by means of two columns, in fact all the column sections are in a single shell. Thus, in the mathematical model of the superstructure we force the number of active trays in sections [I & IV] and [II & V] to be equal, respectively. Therefore, between the active trays of the mentioned sections, AT_j, where *j* is the set of column sections, $1 \le j \le 6$, (which are obtained by means of the integer variables assigned to the top ends of each column section) we choose the greatest of each pair of sections, $\max([AT_I, AT_{IV}])$ and $\max([AT_{II}, AT_V])$

, in order to ensure that the required separation is achieved in both sides of the wall. Actually this is not strictly necessary, because in each side of the wall it is possible to use different number of trays with different tray spacing, as well as different types of trays or packing. However, as the total height of the divided wall column is given by the most difficult separations, we can assert that the proposed superstructure is a good approach. In addition, consider that the number of active trays in sections [I & IV] and [II & V] are respectively the same, makes easier the mechanic design of the column, being able to use conventional sieve trays of double pass, adapted for locating the intern wall. In any case, using the same number of trays in both sides of the wall will make easier the installation of the fixing rings. Of course, one of the sides of the wall will be executing an over-separation from the initial specifications, due to the extra amount of trays, which is, in any case, always a benefit for the separation.

As in the previous examples, in Table 5.6 are summarized the main results of twenty consecutive executions of the algorithm.

Table 5.6 Results of 20 consecutive executions of the PSO algorithm – Example 3

Execution	Total Trays	Feed Tray	CS_I	CS_{II}	CS_{III}	CS_{IV}	CS_V	CS_{VI}	TAC (k$/yr.)	CPU time(s)	Stopping Criterion
1	86	32	10	61	16	45	91	115	2131,099	613	Criterion 1
2	86	33	7	54	18	35	88	117	2130,071	687	Criterion 1
3	85	28	11	55	19	39	91	118	2126,819	632	Criterion 1
4	**85**	**29**	**11**	**65**	**18**	**48**	**81**	**117**	**2126,626**	**632**	**Criterion 1**
5	85	31	9	55	18	44	91	115	2127,593	623	Criterion 1
6	84	29	11	64	18	45	71	117	2128,346	610	Criterion 1
7	85	29	11	65	18	48	81	117	2126,626	668	Criterion 1
8	85	31	9	55	18	44	91	115	2127,593	626	Criterion 1
9	86	32	10	61	16	45	91	115	2131,099	617	Criterion 1
10	85	29	10	63	19	41	69	119	2126,934	661	Criterion 1
11	85	29	8	51	19	36	89	117	2127,404	648	Criterion 1
12	85	28	11	55	19	39	91	118	2126,819	630	Criterion 1

13	86	34	8	57	16	35	89	115	2131,030	625	Criterion 1
14	85	31	10	66	17	38	76	114	2127,853	635	Criterion 1
15	86	32	7	63	19	37	89	117	2132,872	658	Criterion 1
16	86	32	10	61	16	45	91	115	2131,099	880	Criterion 1
17	85	29	11	65	18	48	81	117	2126,626	660	Criterion 1
18	85	29	11	65	18	48	81	117	2126,626	637	Criterion 1
19	85	32	10	63	19	41	69	119	2126,934	626	Criterion 1
20	85	31	8	51	19	36	89	117	2127,404	665	Criterion 1

Stopping Criterion 1: stop due to maximum number of iterations is reached (150 major iterations).

Stopping Criterion 2: the tolerance between best and worse particle is under specification ($1\cdot10^{-5}$).

An interesting characteristic of thermally coupled systems in general or a Divided Wall Column (DWC) in particular is that exist a relatively large number of different configurations (similar number of total trays but with different arrangements in sections) with very similar total costs. This effect can be observed in the 20 consecutive executions where different solutions are obtained but all of them very close each other in terms of total cost and structure. Curiously, this fact gives the designer an extra degree of freedom to consider other aspects like the controllability, hydrodynamics, etc. to select the most adequate tray distribution.

Notice that there are two main configurations with 85 and 86 trays and the mean value of the objective function of this configurations are 2127 and 2130 k$/yr., respectively. Configuration with 85 trays has the lowest TAC value and was obtained the 65% of the times. All the structures can be considered as good solutions, but the solution number 4 is the best one between the solutions with 85 trays. The computational and design results for the mentioned solution are summarized in Figure 5.13 and Table 5.7.

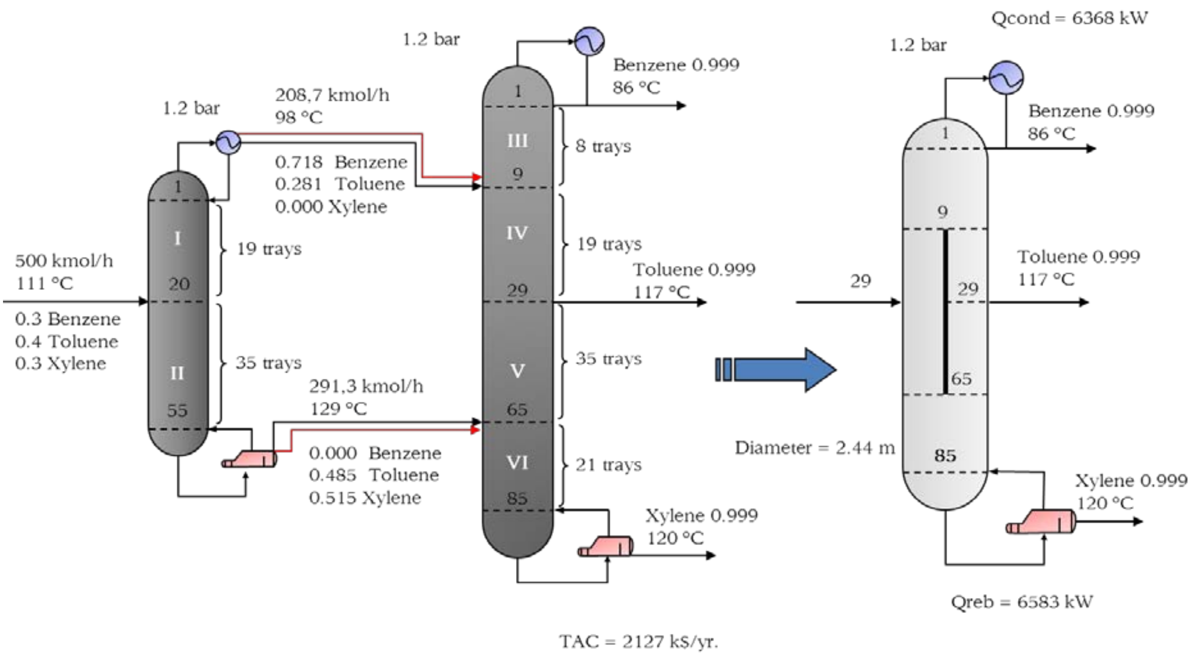


Figure 5.13 Best Solution of Example 3

Table 5.7 Computational Results for Best Solution Example 3

PSO Description	
Number of Particles	20
Major Iterations	150
Function Evaluations	3020
Discrete Variables	6
Stopping Criterion	Stop due to maximum number of iterations is reached
CPU Time (s)	668
Optimal Solution	
TAC (k$/yr.)	2126.63
Capital Cost (k$)	1754.4
Operating Cost (k$/yr.)	1540.0

6 CONCLUSIONS

In this work we have provided a general review of the area of optimal design and synthesis of distillation columns. We have shown that the rigorous optimization of complex distillation processes represents a challenging problem due to the large impact on the investment and operating costs involved. In the literature there are wide array of different approaches, most of them based on the mathematical programming embodied in MI(N)LP or GDP representations. However, these models suffer from important difficulties because of the high degree of nonlinearity and nonconvexity of the equations describing the separations units. This restricts the initial guess to one that has to be very close to a realistic simulation result, and strongly affects the quality of the solution. As a consequence, the resulting optimization models are far from being straightforward, and so, only those skilled in the art are able to utilize and adapt them to their own needs.

In order to overcome the main difficulties that arise in the MI(N)LP and GDP approaches, we have proposed a systematic method that takes advantage of the process simulators and the free derivative optimization algorithm PSO (Particle Swarm Optimization algorithm) for solving global optimization problems: the method simultaneously optimizes the operating parameters of the distillation columns (reflux and reboiler rations, recoveries, ...), as well as the discrete design decisions of the feed and product location, and obtains the optimal number of trays. Three numerical examples were solved to illustrate how to build different superstructures using the process simulator and to assess the robustness and performance of the implemented method. In addition, this approach can be extended to other separation processes. To that end, our future work will be extended to sharp and nonsharp distillation sequences, as well as to thermally coupled distillation sequences.

ACKNOWLEDGMENTS

The authors would like to acknowledge financial support from the Spanish "Ministerio de Ciencia e Innovación" (CTQ2009- 14420-C02-02 and CTQ2012-37039-C02-02).

APPENDIX A: Utility and Capital Costs

The role of the process economics it is important to evaluate the different superstructures proposed in the numerical examples and carry out the process optimization approach described in this work. For this reason, it is essential to define the utility costs used to calculate the different operating costs described in the examples, as well as the nonlinear cost model used to estimate the capital cost of the distillation columns, heat exchangers and compressors.

Utility Costs

The utilities used for the calculation of the operating costs of the proposed flowhseets are cooling water, vapor steam for boilers and electricity. The reported cost of these utilities given by *Turton, R. et al. (2002)*., are shown in Table A.1.

Table A.1 Utility Costs

Utility	Description	Cost ($/GJ)	Cost ($/ Common Unit)
Steam from boilers	a. Low pressure (5 barg, 160 ºC)	7.78	16.22 $/1000 kg
	b. Medium pressure (10 barg, 184 ºC)	8.22	16.40 $/1000 kg
	c. High pressure (41 barg, 254 ºC)	9.83	16.64 $/1000 kg
Cooling water	Process cooling water: 30ºC to 45 ºC	0.354	14.8 $/1000 m^3
Electricity	Electric distribution (110, 220, 440 V)	16.8	60.0 $/MWh

All the operating cost are annualized ($/year) so we consider that the processes operate for 8000 hours per year.

Capital Cost

The estimation of the capital cost for a chemical plant or a single unit operation, as the distillation columns or heat exchangers, must take into consideration many costs, besides the purchase cost of the equipment. These costs can be classified in direct costs (the equipment free on board cost, materials required for installation and labor to install) and indirect costs (freight, insurance and taxes, construction overhead and contractor engineering expenses).

This topic has been covered extensively in the literature. However, the Module Costing Technique, introduced by Guthrie, K. M., 1969/1974, it is generally accepted as the best one for making preliminary cost estimates (*Turton, R. et al., 2003*) and is adopted in this work.

In this approach the equation (A.1) is used to calculate the sum of the direct and indirect costs mentioned above for each piece of equipment (the bare module cost)

$$C_{BM} = C_p^0 F_{BM} \tag{A.1}$$

were:

C_{BM} is the bare module equipment cost that represents the sum of direct and indirect costs.

F_{BM} is the bare module cost factor: multiplication factor to account for the specific materials of construction and operating pressure.

C_P^{0} is the purchase cost for base conditions (carbon steel construction and ambient pressure).

The purchase cost of the equipment can be estimated with the following correlation

$$\log_{10} C_P^0 = K_1 + K_2 (A) + K_3 \left[\log_{10} (A)\right]^2 \tag{A.2}$$

were A is the capacity or size parameter for the equipment. The data for the constants K_i, along with the maximum and minimum values of the size parameter used in the correlation for each piece of equipment used in this work are given in Table A.2.

Table A.2 Equipment cost data for compressors

Equipment Type	K1	K2	K3	A (capacity, Units)	Min Size	Max Size
Compressors	2.2897	1.3604	-0.1027	fluid power, kW	450	3000
Heat exchangers	4.3247	-0.3030	0.1634	area, m^2	10	1000
Towers	3.4974	0.4485	0.1074	volume, m^3	0.3	520

Sieve Trays	2.9949	0.4465	0.3961	area, m^2	0.07	12.30

It is important to notice that the purchase cost of equipment was obtained from a survey of equipment manufactures performed in 2001, so an average value of the CEPCI (Chemical Engineering Plant Cost Index) of 397 should be used in the account of the inflation (*Turton R. et al., 2003*). The value of the Annual CEPCI index for 2011 is 585.7 (Chemical Engineering Journal, May 2012). Therefore, the update bare module equipment cost is given by equation (A.3)

$$C_{p(2011)} = C_p^0 \times \frac{CEPCI_{(2011)}}{CEPCI_{(2001)}} \tag{A.3}$$

On the other hand, the bare module cost factor, F_{BM}, which is related to the material of construction (carbon steel) and pressure operation, is obtained from a set of tables and figures that can be found in the book of *Turton, R. et al., 2003*. Herein are summarized the F_{BM} values used in the calculation of the capital cost of each piece of equipment listed above in Table A.3.

Table A.3 Bare module cost factor

Equipment Type	F_{BM}
Compressors	2.80
Heat exchangers	3.08
Towers	4.07
Sieve Trays	1.0

APPENDIX B: Strategies for the Robust Simulation of Divided Wall Columns

In this appendix the methodology followed to perform the simulation of thermally coupled distillation columns is described. We will focus the attention on the Petlyuk configuration, which is thermodynamically equivalent to a divided wall column. A complete discussion about thermodynamically equivalent configurations and their implications in cost and operability for systems with three or more components can be found in (*Caballero, J.A. & I.E. Grossmann, 2002*).

As can be seen in Figure B.1, we can simulate a divided wall column system using two (or maybe three) conventional columns.

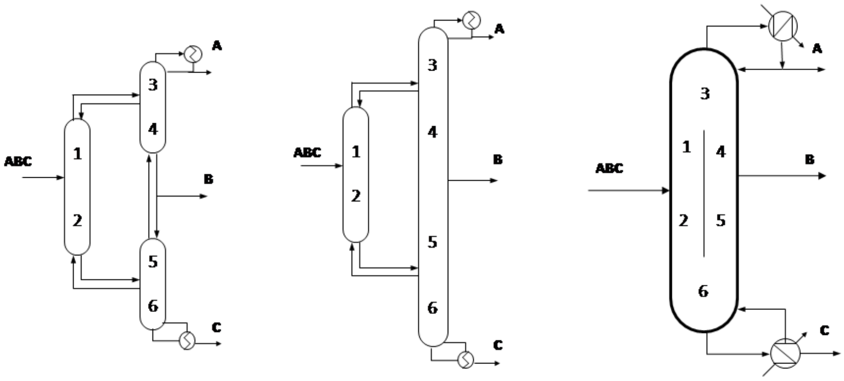


Figure B.1 Petlyuk configuration and the thermodynamically equivalent divided wall column

The problem when we try to simulate a thermally coupled sequence using a process simulator like Aspen Hysys is that it is necessary to introduce a recycle due to the double stream (liquid and vapor) connecting two columns. Therefore, at each major iteration, each column must be converged with two undesirable side effects: a) the computation time for a single simulation increases considerably b) it is relatively easy for the system not to converge.

Simulation Strategies. Acyclic System Simulation

The basic idea is to avoid the recycle structure that appears in Thermally Couple Distillation (TCD) system in a modular simulator. This idea is based on the works by Carlsberg and Westerberg (*Carlberg, N.A. & A.W. Westerberg, 1989*). They proved in the context of Underwood's shortcut method, that the two side streams in a TCD system connecting the rectifying section of the first column (see Figure B.2a) with column 2, is equivalent to a superheated vapor stream whose flow is the net flow (difference between vapor exiting the column and the liquid entering in it) – Figure B.2b –. If the two side streams are connecting the stripping section of the first column with the second column then these two streams are equivalent to a single sub-cooled liquid stream whose flow is the net flow (in this case liquid minus vapor flows). See Figure B.2c,d.

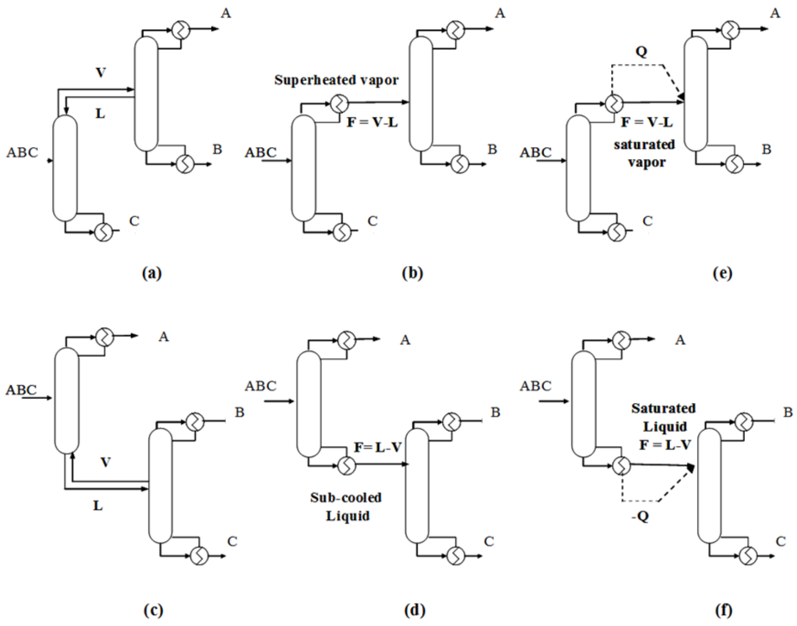


Figure B.2 a, b, e equivalent configurations. c, d, f equivalent configurations

This approach, apparently solve the problem, since each pair of streams could be replaced by a net flow of overheated vapor (enrichment section) or a net flow of subcooled liquid (stripping section) and in this way, the recirculation of information in the flowsheet could be removed.

However, this cannot generally be implemented in a modular process simulator because the degree of superheating and/or subcooling could be so large that it might produce results without physical meaning, and therefore fail in the convergence of the simulator (for example temperatures of liquid below the absolute zero).

Fortunately, it is possible to solve this problem substituting the superheating or subcooling streams by a combination of a material and an energy stream.

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 to provide reflux to the first column. See Figure B.2e. In the stripping section, the material stream is liquid at its bubble point and the energy stream is equivalent to the energy added if we include a reboiler to provide vapor to the first column, see Figure B.2f.

Although this strategy is only an artificial tool to simulate the behavior of the thermally coupled system avoiding the recycles, there is not an approximation at all if the streams introduced/withdraw in/from the column 2 were in equilibrium with the liquid and vapor flowing through this column (V_1^{C1} with L_2^{C2}). See Figure B.3.

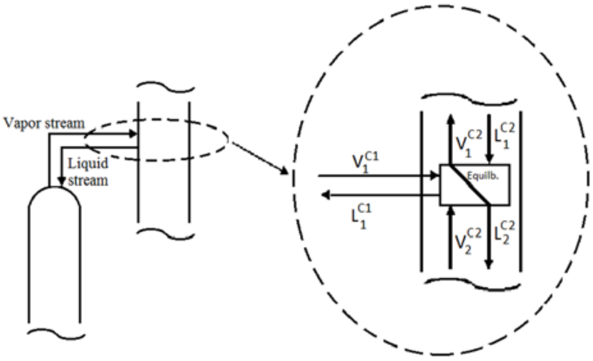


Figure B.3 Details of the connection between columns, "Cyclic system simulation"

Unfortunately, this is not entirely true. The Carlberg & Westerberg approximation considers the idea that there is no mass exchange between the vapor and liquid streams. In the rigorous simulation, the energy streams are used to simulate the elimination of liquid that is withdraw from the column 2 to the column 1, since it vaporizes part of the liquid stream, which is equivalent to the liquid removed, see Figure B.4.

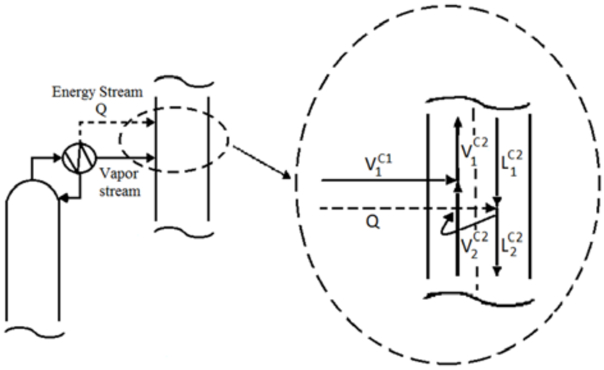


Figure B.4 Details of the connection between columns, "Acyclic system simulation"

This vapor stream is added to the vapor upward flow within the column. This is the main source of error. But if the vapor and liquid streams are introduced/withdrawn in/from the same tray the error introduced is small and usually can be neglected. In any case, in the worst possible scenario the values obtained with this technique are excellent initial points to converge the rigorous simulations of the original system.

Steps for the Simulation of a Divided Wall Column

In the following section we will explain the steps for the simulation of a divided wall column using a process simulator.

First we simulated the acyclic sequence (each thermal coupled is substituted by a mass and an energy stream), using conventional distillation columns, see Figure B.5. To do the thermal couple, we connected the mass and energy streams that leave the condenser in the same tray of the next column, and in the same way, the mass and energy streams that leave the reboiler in the same tray of the next column. Then, we converge the acyclic sequence and the results of the acyclic simulation are used as initial points of the actual system (with cyclic structure). The initial conditions (pressure, temperature, flow and compositions) of the recycled streams – vapor entering in the bottom tray and liquid entering in the top tray – are the conditions of the vapor/liquid exiting from the reboiler/condenser in the acyclic system.

It is important to remark that the distillate of the first column is equivalent to a saturated vapor stream plus an energy stream. Therefore, we are adding heat in the upper part of the second column. However, the bottom stream is equivalent to a saturated liquid stream minus a heat stream and therefore, we are removing heat from the second column. It is very important take into account the sign of the energy stream added (a negative sign in Hysys means that we are removing heat that in fact is what we want in the lower part of the second column).

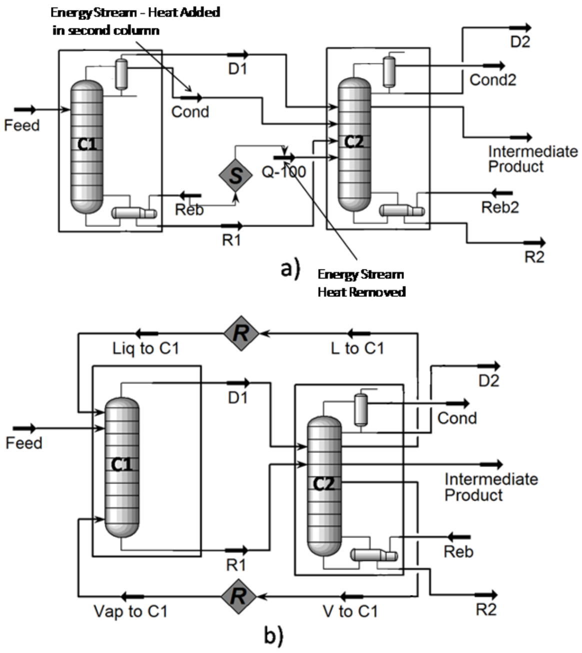


Figure B.5 Simulations of (A) acyclic and (B) cyclic system configurations

REFERENCES

ANNAKOU, O. AND MIZSEY, P., *“Rigorous Investigation of Heat Pump Assisted Distillation”*, Heat Recovery Systems & CHP, **1995**, 15 (3), 241.

BARBARO, A. AND BAGAJEWICZ, M. J., *“Managing financial risk in Planning Under Uncertainty”*, AIChE Journal, **2004**, 50, 5, 963.

BARTTFELD, M., AGUIRRE, P. A. AND I. E. GROSSMANN, *“Alternative Representations and Formulations for the Economic Optimization of Multicomponent Distillation Columns”*, Comput. Chem. Eng., **2003**, 27, 363.

BAUER, M. H., STICHLMAIR, J., *“Design and Economic Optimization of Azeotropic Distillation Processes Using Mixed Integer Nonlinear Programming”*, Comput. Chem. Eng., **1998**, 22, 1271.

BENDERS, J. F., *“Partitioning Procedures for Solving Mixed-Variables Programming Problems”*, Numer. Math., **1962**, vol 4, 238.

BORCHERS, B. AND J. E. MITCHELL, *“An Improved Branch and Bound Algorithm for Mixed Integer Nonlinear Programming*", Computers and Operations Research, **1994**, vol 21, 359.

CABALLERO, J. A. AND I.E. GROSSMANN *"Thermodynamically Equivalent Configurations for Thermally Coupled Distillation Columns*", AIChE Journal, 2003, 49, 2864.

CABALLERO, J. A., MILÁN – YAÑEZ, D. AND GROSSMANN, I. E., *“Rigorous Design of Distillation Columns: Integration of Disjunctive Programming and Process Simulators”*, Industrial and Engineering Chemistry Research, **2005**, 45(25), 8454.

CARLBERG, N. A. and Westerberg, A. W., *“Temperature Heat Diagrams For Complex Columns.3. Underwoods Method For The Petlyuk Configuration”*, Industrial & Engineering Chemistry Research, **1989**, 28, (9), 1386.

CIRIC, A. R. AND GU, D. *“Synthesis of Nonequilibrium Reactive Distillation Processes by MINLP Optimization”*, AIChE, **1994**, 40, 1479.

DOHERTY, M. F., AND MALONE M. F., *“Conceptual Design of Distillation Systems*", Mc Graw Hill, **2001**.

DÜNNEBIER, G. AND PANTELIDES, C. C., *“Optimal Design of Thermally Coupled Distillation Columns”*, Ind. Eng. Chem. Res., **1999**, 38, 162

DURAN, M. A. AND I. E. GROSSMANN, *“An Outer-Approximation Algorithm for a Class of Mixed-integer Nonlinear Programs”*, Math Programming, **1986**, vol 36, 307.

EBERHART, R., SIMPSON, P., AND DOBBINS, R., *“Computational Intelligence PC Tools”*, **1996**, Academic Press Professional.

FIDKOWSKI, Z. AND KROLIKOWSKI, L., *“Minimum Energy-Requirements Of Thermally Coupled Distillation Systems”.* AIChE Journal, **1987,** 33, (4), 643.

FLETCHER, R. AND S. LEYFFER, *“Solving Mixed Integer Nonlinear Programs by Outer Approximation”,* Math Programming, **1994**, vol 66, 327.

GEOFFRION A. M., *“Generalized Benders Decomposition”*, JOTA, **1972**, vol 10, 237.

GUPTA, O. K. AND RAVINDRAN, V., *“Branch and Bound Experiments in Convex Nonlinear Integer Programming”*, Management Science, **1985**, vol 31(12), 1533.

GUTHRIE, K. M., *“Capital Cost Estimating”*, Chem. Eng., **1969**, 76, 114

GUTHRIE, K. M., *“Process Plant Estimating, Evaluation and Control”,* Craftsman Book Company of America (1974).

GROSS, B. AND ROOSEN, P., *“Total process optimization in chemical engineering with evolutionary algorithms”.* Comput. Chem. Eng. **1998**, 22(0)xS229-S236

I.E GROSSMANN AND RUIZ, J. P., *“Generalized Disjunctive Programming: a Framework for Formulation and Alternative Algorithms for MINLP Optimization”, **2009**.*

KENNEDY, J. AND EBERHART, R., *“A Discrete Binary Version of the Particle Swarm Algorithm”*, **1997**, in Proceeding conference on systems, man, and cybernetics, 4104.

KENNEDY, J. AND EBERHART, R., *Particle Swarm Optimization* Proceeding of IEEE *International Conference on Neural Networks*, **1995**, vol 4, 1942.

KOSSACK, S., KRAEMER, K., GANI, R. AND MARQUARDT, W., *“A Systematic Synthesis Framework for Extractive Distillation Processes”*, Chem. Eng. Research and Design, **2008**, 86, 781.

LANG, H. J., “*Cost Relationships in Preliminary Cost Estimates”*, Chem.Eng., **1947**, 54, 117.

LANG, H. J., *“Engineering Approach to Preliminary Cost Estimates”*, Chem. Eng, **1947**, 54, 130.

LANG, H. J., *“Simplified Approach to Preliminary Cost Estimates”*, Chem. Eng., **1948**, 55, 112.

LEE, S. AND I. E. GROSSMANN, *“Global Optimization of Nonlinear Generalized Disjunctive Programming with bilinear inequality constraints: Application to Process Networks”*, Comput. Chem. Eng., **2003**, vol 27, pp. 1557*.*

LEE, S. AND I. E. GROSSMANN, *“New Algorithms for Nonlinear Generalized Disjunctive Programming”*, Comput. Chem. Eng., **2000**, vol 24, pp. 2125*.*

LEE, S. AND I. E. GROSSMANN, *“New Algorithms for Nonlinear Generalized Disjunctive Programming”, Computers and Chemical Engineering”*, **2000**, vol 24, 2125.

LEYFFER, S., *“Integrating SQP and Branch and Bound for Mixed Integer Nonlinear Programing”*, Computational Optimization and Applications, **2001**, vol 18, 295.

NABAR, S. AND SCHRAGE, L., "*Modeling and Solving Nonlinear Integer Programming Problems"*, Presented at Annual AIChE Meeting, Chicago, **1991**.

NAVARRO, M. A., JAVALOYES, J., CABALLERO J. A. AND I. E. GROSSMAN, *“Strategies for the Robust Simulation of Thermally Coupled Distillation Sequences”*, Computers and Chemical Engineering, **2012**, 36, 149.

PETLYUK, F. B., PLATONOV, V. M., AND SLAVINSK D., *“Thermodynamically Optimal Method for Separating Multicomponent Mixtures”*, International Chemical Engineering, **1965**, 5(3), 555.

PIKULIK, A AND H. E. DIAZ, *“Cost Estimating for Major Process Equipment”*, Chem. Eng., **1977**, 84, 106

RAMAN, R. AND I. E. GROSSMANN, *“Modeling and Computational Techniques for Logic Based Integer Programming”*, Comput. Chem. Eng., **1994,** vol 18, pp. 563.

RAMAN, R. AND I. E. GROSSMANN, *“Relation Between MILP Modeling and Logical Inference for Chemical Process Synthesis”*, Comput. Chem. Eng., **1991,** 15, 73.

SMITH, R., *“Chemical Process Design and Integration”*, John Wiley & Sons, **2005**.

SOAVE, G. AND FELIU, J. A. *“Saving Energy in Distillation Towers by Feed Splitting”*, Applied Thermal Eng., **2002**, 22, 889.

STUBBS R. AND S. MEHROTRA, *“A Branch-and-Cut Method for 0-1 Mixed Convex Programing”, Mathematical Programming,* ***1999****, vol 86(3), 515.*

TÜRKAY, M. AND I.E. GROSSMANN, *“A Logic Based Outer – Approximation Algorithm for MINLP Optimization of Process Flowsheets”,* Comput. Chem. Eng., **1996**, vol 20,. 959.

TURTON, R., BAILER, R. C., WHITING, W. B., AND SHAEIWITZ, J. A., *“Analysis, Synthesis and Design of Chemical Processes”*, Prentice Hall, **2002**, second edition.

VISWANATHAN, J. AND I. E. GROSSMANN, *“An Alternate MINLP Model for Finding the Number of Trays Required for a Specified Separation Objective”*, Comput. Chem. Eng., **1993**, 17, 949.

VISWANATHAN, J. AND I. E. GROSSMANN, *“Optimal Feed Locations and Number of Trays for Distillation Columns with Multiple Feeds”*, Ind. Eng. Chem., **1993**, 32, 2942.

YEOMANS, H., AND I. E. GROSSMANN, *“Disjunctive Programing Models for the Optimal Design of Distillation Columns and Separation Sequences”*, Ind. Eng. Chem. Res., **2000**, 39 (6), 1637.

YUAN, X., S. ZHANG, L. PIBOLEAU AND S. DOMENECH, "*Une Methode d'optimisation Nonlineare en Variables Mixtes pour la Conception de Procedes*", RAIRO, **1988**, 22, 331.