

Optimal Synthesis of Distillation Columns: Integration of Process Simulators in a Disjunctive Programming Environment

José A. Caballero^a, Daniel Milán-Yañez, Ignacio E. Grossmann^b,

^aUniversity of Alicante

Apartado de Correos 99, E-03080, Alicante Spain.

^bCarnegie Mellon University

5000 Forbes Avenue. Pittsburgh, PA USA

Abstract

The optimal economic design of a distillation column involves the selection of the number of trays, feed and side-streams locations and operating conditions. In this paper we present a superstructure based optimization algorithm that combines the capabilities of commercial process simulators – taking advantage of the specially tailored algorithms designed for distillation and property estimation implemented in these simulators- and generalized disjunctive programming (GDP).

The algorithm iterates between two types of sub-problems: an NLP sub-problem, in which the trays are divided in existing and non-existing (non-existing trays behave like simple bypasses without mass or heat exchange) and an especially suited master (MILP) problem. NLP sub-problems are solved connecting the process simulator with an NLP external solver. An example is also included showing promising results.

Keywords: Distillation, Generalized Disjunctive Programming, MINLP, Column Sequences.

1. Introduction

As a consequence of the important research effort in distillation, the simulation of these systems has acquired an important degree of maturity and a good number of robust and reliable numerical methods have been especially developed for distillation; some of the most important can be found in Kister (1992); most of them are nowadays included 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, etc), but with discrete decisions (i.e. number of trays, 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, Grossmann et al (2004).

The most successful approach using mathematical programming techniques applied to distillation was due to Viswanathan and Grossmann (1990). These authors proposed a superstructure in which the feed location is fixed, but the total number of trays above or below the feed it is not. The problem was formulated as a MINLP in which the MESH equations together with mixed integer relations were the constraints. A major difficulty in that model is due to the non-existing trays which can produce important numerical problems. Even though, the convergence problems, this model has been successfully applied by different research groups, see for example, Ciric and Gu (1994), Bauer and Stichlmair (1998), Dünnebier and Pantelides (1999).

The introduction of Generalized Disjunctive Programming (GDP, allowed overcome some of the difficulties of the MINLP models. Yeomans and Grossmann (2000) proposed a GDP model in which the non existing trays were considered as simple bypasses of vapour and liquid flows without mass or heat transfer. Mass and energy balances are trivially satisfied.

In this paper we propose an algorithm for designing distillation columns, integrating a process simulator in a Generalized Disjunctive Programming formulation. All numerical aspects related to the convergence of a distillation column, selection of thermodynamic models, transport properties etc, are specified in the simulation environment. An external optimizer is connected to the simulator in order to solve the NLP sub-problems, and a 'specially suited' Master problem that takes the form of a Mixed Integer Linear Programming (MILP) Problem is solved. In next sections we provide details of the algorithm and will show a validating example.

2. Description of the GDP algorithm for distillation column design

For the sake of simplicity, but without losing generality, let us centre in a conventional column: one feed and two products. Extension to complex columns is very straightforward. 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 operation conditions (i.e. reflux ratio, heat load), for separating the mixture in two streams within given specifications, using a conventional distillation column.

The superstructure used in this work is shown in Figure 1

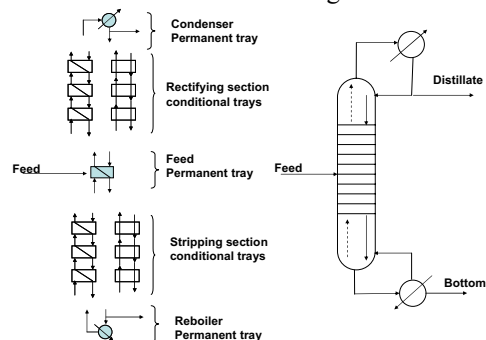


Figure 1. Superstructure for the optimal design of a distillation column.

The superstructure is based on that proposed by Yeomans and Grossmann (2000) that according with Barttfeld et al (2003) is the superstructure which has shown better computational results when is solved with a GDP formulation, Figure (1).

The algorithm iterates between NLPs that correspond to configurations with fixed number 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) and estimates of continuous variables to be solved by the next NLPs.

The main steps of the algorithm are comment in next paragraphs.

2.1 Initialization

All the needed 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, rectifying and stripping sections, must be selected large enough to be an upper bound to the optimal number of trays.

2.2 Solving the nonlinear programming problems (NLPs)

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 or not, a feasible configuration with a fixed number of trays is selected. A simple and direct approach consist in forcing to the non existing trays 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 to zero.

To avoid equivalent solutions the existing trays should be consecutive. Initially a good option consists in 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.

The NLP to be solved can be written as follows:

$$\begin{aligned}
 \min_{x_D} : z &= f(x, u, x_D) \\
 \text{s.a. } h_I(x, u, x_D) &= 0 \text{ implicit} \\
 \left. \begin{aligned} g(x, u, x_D) &\leq 0 \\ h_E(x, u, x_D) &= 0 \end{aligned} \right\} &\text{ explicit}
 \end{aligned} \tag{P1}$$

In the problem P1 x_D makes reference to the design (independent) variables. These are equal to the degrees of freedom of the problem. x makes reference to all the other variables of the process. These variables are calculated by the simulator and can only be read. u makes reference to a set of fixed parameters that are not modified during the calculation procedure In the distillation case includes 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 viewed by the user. Equations g and h_E are explicit external constraints added to the problem. Finally, f is a scalar objective function, for example total annualized cost.

The external solver was developed in MATLAB and connected to the process simulator, HYSYS, through a client-server Active-x automation technology.

In order to avoid infeasible solutions it is convenient to use a set of independent variables that allow the process simulator to converge in almost any situation. In other words, it is important that the set of implicit equations $h_I(x, u, x_D) = 0$ can be solved as easily as possible because we do not have control on how these equations are solved. Difficult specifications should be left to the external constraints.

2.3 The Master Problem.

Let us define the following sets:

$T = \{ i \mid i \text{ is a tray in the distillation column} \}$; $TR_T = \{ r \mid r \text{ is a tray in the rectifying section of the column} \}$; $TS_T = \{ s \mid s \text{ is a tray in the stripping section of the column} \}$; $EQ = \{ j \mid j \text{ is an external (explicit) inequality constraint} \}$; $IEQ = \{ j \mid j \text{ is an external (explicit) equality constraint} \}$; $D = \{ n \mid n \text{ is a design (independent) variable} \}$

In order to generate the Master, first 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. First a problem adding one more tray to the rectifying section is solved, then another with two more trays, three trays etc., until all the trays in the rectifying section exists. 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 (2) helps to clarify the procedure.

Taking previous paragraph in mind let us define the following parameters:

ΔObj_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 bellow (above) it in this section must exist.

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

$\Delta hE_{i,j}^k(x, u, x_D^k)$ = Difference between the value of the external equality constraint j, when the tray i exist, and the equality constraint j in the original NLP^k problem.

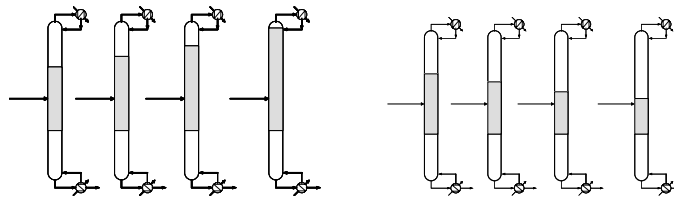


Figure 2. Sequence of simulations solved for generating the master (rectifying section)

The master problem can be written as follows:

$$M - P1 \quad \min : \alpha + P \left(\sum_{j \in IEQ} slack1_j + \sum_{j \in EQ} slack2_j \right)$$

$$\begin{aligned}
s.t. \quad & f(x, u, x_D^k) + \sum_n \left(\frac{\partial f}{\partial x_{D_n}} \right)_{x_D=x_D^k} (x_{D_n} - x_D^k) + \sum_{i \in T} W_i \Delta obj_i^k \leq \alpha \\
& g_j(x, u, x_D^k) + \sum_n \left(\frac{\partial g_j}{\partial x_{D_n}} \right)_{x_D=x_D^k} (x_{D_n} - x_D^k) + \sum_{i \in T} W_i \Delta g_{i,j}^k \leq slack1_j \quad \forall j \in IEQ \\
& sign(\lambda_j^k) \left[h_{E_j}(x, u, x_D^k) + \sum_n \left(\frac{\partial h_{E_j}}{\partial x_{D_n}} \right)_{x_D=x_D^k} (x_{D_n} - x_D^k) + \sum_{i \in T} W_i \Delta h_{E_{i,j}}^k \right] \leq slack2_j \quad \forall j \in EQ \\
& slack1_j \geq 0; \quad slack2_j \geq 0 \\
& \sum_{r \in TR_T} W_r = 1 \quad ; \quad \sum_{s \in TS_T} W_s = 1 \quad \quad k = 1, 2, 3, \dots, K \\
& W_i \in \{0, 1\}
\end{aligned}$$

In problem M-P1 α is an auxiliary variable used to place the objective function with the constraints of the problem; slack1 and slack2 are slack positive variables to assure feasibility. The objective function is formed by the auxiliary variable α and an exact penalty.

The left hand side of the first constraint is formed by two parts; the first is a linearization of the objective function respect to the design (independent) variables. The second part is the contribution of the existence (or non-existence) of the tray “i” respect to the configuration in iteration k. The same procedure is followed for the external equality and inequality constraints. In this case, deviations from the value of g_j and h_{E_j} when the tray i exists, respect to their value in the optimal solution of the NLP^k problem, are included. In the equality constraints the parameter $sign(\lambda_j^k)$ 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.

W_i is 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 take the value of 1 for lowest existing tray (all trays below it do not exist and all trays above it exist). Last constraints explicitly state that in each section only one of those binary variables can take the value one.

Finally mention than in the M-P1 problem all the linear constraints are accumulative In order to avoid repeated solutions a binary cut is added in each iteration.

2.4 Example.

Let us illustrate the methodology with the extractive distillation of ethanol using ethylene glycol as entrainer. The first column is formed initially by 80 trays. The entrainer is fed in tray 20 and the feed in the 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 must exist. In the middle section if a given tray exists all trays above it must exist, and in the lower section if a given tray exists all trays above it must exist. In The second column 15 trays were postulated, the feed tray is located in tray 7. If the feed is completely specified, for the NLPs we have five degrees of freedom. As

independent variables we have chosen for column 1: ratio entrainer – feed, reflux ratio, molar fraction ratio water-ethanol in the bottom stream, and for the second column: reflux ratio and molar fraction ratio ethylene glycol-water in bottoms. molar fraction of each component higher than 0.999 is added through external constrains. Optimal results are shown in Figure (3). The problem converges in 5 major iterations with around 5 min. of CPU time.

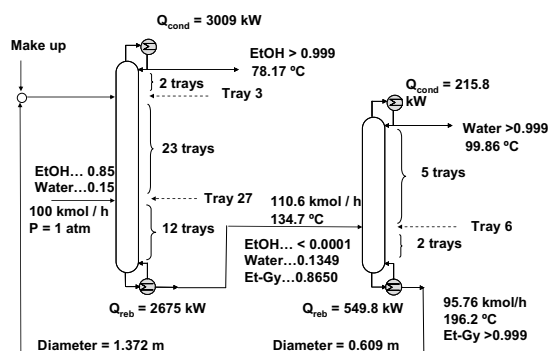


Figure 3. Optimal solution for extractive distillation of ethanol.

References

- Kister, H.Z. 1992. *Distillation Design* McGraw-Hill. New York
- Grossmann, I.E.; Aguirre, P.A. & Bartfeld, M. 2004. Optimal Synthesis of Complex Distillation Columns Using Rigorous Models. In *Computer Aided Chemical Engineering-18*. A. Barbosa-Pavoa & H. Matos (Eds.) ESCAPE-14. Elsevier, pp 53-74.
- Viswanathan, J. & Grossmann, I.E. 1990. A Combined Penalty Function and Outer Approximation Method for MINLP Optimization. *Computers & Chemical Engineering*, 14(7), 769.
- Ciric, A.R. & Gu, D. 1994. Synthesis of Nonequilibrium Reactive Distillation Processes by MINLP Optimization. *AIChE Journal*, 40(9), 1479-1487.
- Bauer, M.H. & Stichlmair, J. 1998. Design and Economic Optimization of Azeotropic Distillation Processes Using Mixed Integer Nonlinear Programming. *Computers & Chemical Engineering*, 22(9), 1271-1286.
- Dünnebier, G. & Pantelides C.C. 1999. Optimal Design of Thermally Coupled Distillation Columns. *Ind. Eng. Chem. Res.* 38, 162-176.
- Yeomans, H. & Grossmann, I.E. 2000. Disjunctive Programming Models for the Optimal Design of Distillation Columns and Separation Sequences. *Ind. Eng. Chem. Res.* 39, 1637-1648.
- Bartfeld, M.; Aguirre, A.P. & Grossmann, I.E. 2003. Alternative Representations and Formulations for the Economic Optimization of Multicomponent Distillation Columns. *Computers and Chemical Engineering*. 27, 363-383.

Acknowledgments

Authors gratefully acknowledge the financial support from the Spanish “*Ministerio de Ciencia y Tecnología*” under project PPQ – 2002-01734.