Disjunctive Optimization Design Models for Complex Liquid–Liquid Multistage Extractors

Juan A. Reyes-Labarta
Dept. of Chemical Engineering, University of Alicante, Ap. Correos 99, Alicante E-03080, Spain

Ignacio E. Grossmann
Dept. of Chemical Engineering, Carnegie Mellon University Pittsburgh, PA 15213

A new method for the optimal design of multicomponent liquid–liquid extraction processes using multistage countercurrent extractor systems is proposed. The method determines the optimum number of equilibrium stages and flow rates needed to obtain a specified product separation and recovery, and accounts for the possibility of side feed streams and product extractions. A superstructure is proposed that has embedded all potential configurations and interconnections. Based on this superstructure representation, the problem is formulated as an optimization problem using generalized disjunctive programming (GDP) to minimize the total cost of the process, subject to design specifications. The robustness and computational efficiency of the model is illustrated with different cases involving single and complex countercurrent cascades in a quaternary liquid–liquid system.

Introduction

Liquid–liquid extraction is a competitive separation technology in a number of important applications (Thornton, 1992). For example, distillation may be impractical for separating mixtures containing heat-sensitive materials, components with high or close boiling temperatures, very small relative volatilities, low concentrations, or that give rise to azeotropes. Such components can be separated by their differences in solubility with a suitable solvent through extraction. Due to its low energy consumption, solvent extraction also plays a major role in the treatment of effluent streams for the removal of toxic materials to comply with environmental requirements. Other applications of liquid–liquid extraction include petrochemical processes (such as removal of aromatics from kerosene or lubricating oils to improve their final properties, desulfurization, butadiene separation), pharmaceutical processes (such as antibiotic, vitamins, proteins), food processing (such as refining of fats and oils using propane, recovery of lactic acid, biopolymer extraction), and fertilizer industry and inorganic processes (such as recovery of metals such as copper, nickel, and cobalt).

The simulation of liquid–liquid extraction systems can be performed rigorously by the sum-rates method (SR) of Tsuboka and Katayama (1976), but classic methods for the design of extraction systems involve the use of graphical techniques, which combine material balances with liquid–liquid equilibrium relations (Treybal, 1968; Seader and Henley, 1998). However, these methods are restricted to ternary systems, such as the work by Barnes (1999, 2000) using Bancroft's coordinates. Minotti et al. (1996, 1998) developed a design method similar to the one for nonideal multicomponent distillation systems (Fidkowski et al., 1991), where the minimum solvent flow rate to the extractor is determined by a geometric analysis of the composition profiles and their fixed points. Recently, Reyes-Labarta (1998) and Marcilla et al. (1999) suggested a stage-by-stage method of a multistage extractor using an analytical extension of the Ponchon and Savarit method for the design of distillation columns (Marcilla et al., 1997). This method solves alternately the mass balances and the equilibrium equations to determine the optimal number of stages and solvent flow.

Since designs for liquid–liquid extraction systems involve not only the selection of the number of stages, but also the configuration of extractors and distribution of streams, the

---

Correspondence concerning this article should be addressed to I. E. Grossmann.

problem of optimally designing these systems requires the optimization of a superstructure that has embedded all alternatives of interest. Treating the existence or nonexistence (inactivity) of the different stages with binary variables and taking into account the nonlinear equations for the equilibrium and total flow rates/composition relations, the design problem can be formulated as a mixed-integer nonlinear programming (MINLP) problem (Grossmann, 1996).

In this article, instead of using MINLP techniques directly, we address the synthesis and design of multistage extractors for multicomponent liquid–liquid systems with logic-based optimization techniques. Specifically, a nonlinear model based on generalized disjunctive programming (GDP) (Raman and Grossmann, 1994) is proposed that relies on the identification and application of mass balances and equilibrium equations for conditional stages to reduce the size of the nonlinear programming (NLP) subproblems that have fixed values of the binary variables. When entering streams are at the same temperature and heat of mixing is negligible, the operation can be considered as isothermal, and therefore the energy balances can be omitted. A variant of the logic-based Outer Approximation algorithm (Turkay and Grossmann, 1996) is proposed to solve the problem. This algorithm consists of iterating between reduced NLP subproblems and a mixed-integer linear programming (MILP) master problem, which uses the big-M formulation to model the disjunctions. The combination of the disjunctive model and the appropriate logic-based solution algorithm greatly improves the robustness of the design procedure. Although the proposed method can handle standard thermodynamic methods for liquid–liquid equilibrium (NRTL, UNIFAC, UNIQUAC), empirical correlations (Reyes et al., 1999) of the solubility surface and the tie-lines data are used in order to more accurately calculate the equilibrium of the water–acetone–chloroform–acetic acid mixture considered in this article. Several examples for single and complex countercurrent cascades using a quaternary liquid–liquid system are presented to illustrate the effectiveness of the proposed GDP model.

Problem Statement

Figure 1 shows a countercurrent multistage extraction cascade. In this type of system the *extract* stream (normally organic phase) and the *raffinate* stream (normally aqueous phase) have countercurrent flow throughout the various stages. The final products are represented by E_1 and R_n, respectively. These phases should have a viscosity that is as low as possible to favor the mass transfer. The stages are considered to be in equilibrium, which means that the incoming streams are perfectly mixed and the outcoming streams are in equilibrium, though experimental efficiency factors could be easily included in order to be more realistic. Different arrangements of equilibrium stages can be used to perform extraction operations, such as single contact, repeated contact with fresh solvent in each stage (multistage crossflow), countercurrent contact, and countercurrent contact with solvent reflux (Wankat, 1988).

The design problem can be stated as follows. Given is a set of feed streams with known composition and specified separation of solutes in the product streams. The problem consists of determining the optimal number of stages, the feed stream locations, the solvent flow rate, and the existence and locations of intermediate solvent streams, side feed/product streams, and fractional bypasses of the initial feed stream given a specified set of countercurrent extraction stages. The objective is to minimize the total annualized cost of equipment and utilities.

Multistage Countercurrent Extractor

In order to achieve a specified separation an upper bound for the number of stages in the cascade is specified. The value of ten should be sufficiently large in most of the liquid–liquid multistage extractors. If a solution is obtained with ten stages, one can increase the upper bound to a higher number until the number of stages that is calculated is strictly less than the upper bound. The selection of the stages for the optimal extraction cascade will then be performed among this set of stages.

General countercurrent extractor superstructure

Figure 2 shows the superstructure for a complex cascade with up to n potential stages, an initial feed stream (R_0) to be treated, and a solvent feed stream (E_0). This superstructure considers the possibility of a bypass ($R_{0,byp}$) of the initial feed stream, side solvent feed streams EL_j, and side raffinate feed/product streams (RL_k, PL_q, respectively), bearing in mind that all the side feeds (raffinate and solvent) can have different compositions. The side streams also have the possibility of being divided into the different stages (streams $RL_{k,j}$ and $PL_{q,j}$, respectively).

A straightforward approach to optimize the superstructure would be to allow the stages to disappear in the cascade by modeling the design problem as an MINLP using big-M constraints in which equations and inequalities are relaxed for

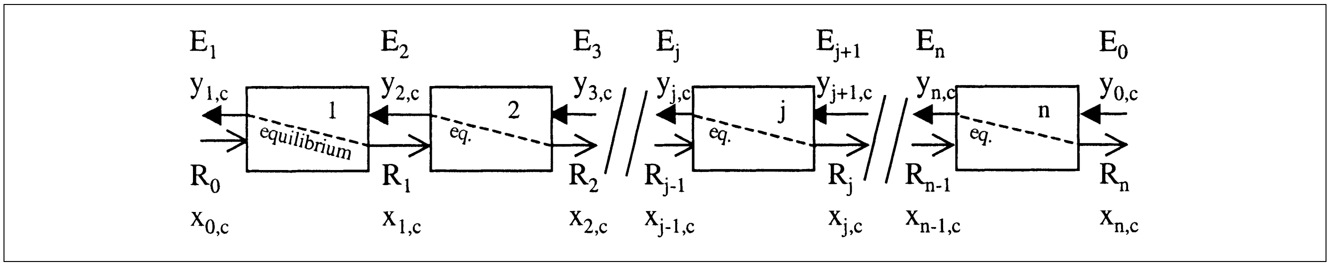

Figure 1. Countercurrent extractor.

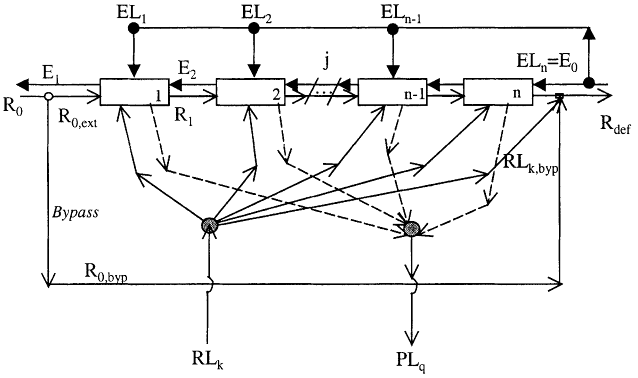

Figure 2. Superstructure of a multistage countercurrent extractors.

inactive stages. However, this approach has poor numerical performance, since it leads to singularities that can cause infeasible or suboptimal designs. This is due to the nonconvexities introduced by the bilinear terms in the mass balances and the equilibrium relations. Moreover, the resulting MINLP model that has to be solved is very large, because all equations must be converged whether or not the corresponding stage is selected (such as phase equilibrium). Therefore, we use a GDP model and algorithm to circumvent these problems.

The approach proposed in this article handles through disjunctions the equations that take place in active and inactive stages. For existing stages the following equations are considered:

1. Total and individual mass balances.
2. Equilibrium equations.
3. Summation of mass fractions in each phase equal to 1.
4. Relation between total and individual flow rates for all the components present in every stream through its mass fraction.

As shown in Figure 3 (see also Yeomans and Grossmann, 2000), for nonexisting or inactive stages, the equations considered are simply input−output relations in which no mass transfer takes place (inlet and outlet flows are the same for each phase). Because the mass balances include the trivial solution, the only difference between existing and inactive stages is the application of the equilibrium equations.

The advantage of the proposed modeling approach is that the nonconvex equilibrium relations do not have to be converged for inactive stages, making the convergence of the optimization procedure more reliable. Also, by using GDP as the modeling tool, the computational expense can be reduced, as can the likelihood of getting trapped into poor suboptimal solutions (Turkay and Grossmann, 1996).

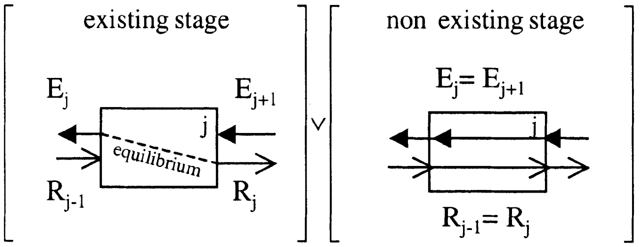

Figure 3. Disjunction for stage existence.

Generalized disjunctive programming model

In this section we present the detailed GDP model for the superstructure in Figure 2. Consider the following definitions for the model: *COMP* is the set of components c present in the system (including those in either the feed and solvent); *NT* represents the set of permanent stages j in the cascade, where the first stage ($j = 1$) corresponds to the initial feed stage, and the last stage ($j = n$) is the solvent feed stage; *NINT* represents the set of intermediate stages ($1 < j < n$); K and Q are the set of side feed and products streams, respectively, and c is used to indicate a component c mass flow.

Let $\xi_{EI,kc}$ represent the recovery fraction of the key component kc in the final stream E_1, and $\tau_{R,\mathrm{def},kc}$ the purity of the key species kc in the final raffinate product R_{def}. Here $\xi_{PLq,kc}$ and $\tau_{PLq,kc}$ are the recovery fraction and purity, respectively, of the key species kc in the side raffinate product PL_q. The parameters that are assumed to be known and the positive continuous variables are given in the Notation section.

The overall constraints include the purity and recovery requirements, and the general cascade mass balance. These constraints are as in the following set of equations (Eqs. 1−15).

1. Purity and recovery requirements in the final products streams E_1 and R_{def}:

$$\left.\begin{array}{l} E_{1,kc'} \geq \xi_{E1,kc'} \cdot (R_{0,kc'}) + \sum\limits_{k=1}^{K} RL_{k,0,kc'} \\ x_{\boldsymbol{R}\mathrm{def},\mathrm{kc'}} \leq \tau_{R\mathrm{def},kc'} \end{array}\right\}$$

$$\text{if } kc' = \text{key component in the extraction stream} \tag{1}$$

$$\left.\begin{array}{l} E_{1,kc''} \leq \xi_{E1,kc''} \cdot \left(R_{0,kc''} + \sum\limits_{k=1}^{K} RL_{k,0,kc''} \right) \\ x_{R\mathrm{def},kc''} \geq \tau_{R\mathrm{def},kc''} \end{array}\right\}$$

$$\text{if } kc'' = \text{key component in the raffinate stream.} \tag{2}$$

2. Purity and recovery requirements in the side products streams PL_q:

$$\left.\begin{array}{l} PL_{q,kc''} \geq \xi_{PLq,kc''} \left(R_{0,kc''} + \sum\limits_{k=1}^{K} RL_{k,0,kc''} \right) \\ x_{PLq,kc''} \geq \tau_{PLq,kc''} \end{array}\right\}$$

$$\text{if } kc'' = \text{key component in the raffinate stream.} \tag{3}$$

3. Global mass balance:

$$E_{1,c} + R_{\mathrm{def},c} + \sum_{q=1}^{Q} PL_{q,c} = R_{0,c} + \sum_{k=1}^{K} RL_{k,c} + \sum_{j=1}^{n} EL_{j,c} \qquad \forall c \in COMP. \quad (4)$$

4. First stage ($j = 1$):

$$R_{1,c} + E_{1,c} + \sum_{q=1}^{Q} PL_{q,1,c} = R_{0,\mathrm{ext},c} + E_{2,c} + \sum_{k=1}^{K} RL_{k,1,c} + EL_{1,c} \qquad \forall c \in COMP. \quad (5)$$

5. Last stage ($j = n$):

$$R_{n,c} + E_{n,c} + \sum_{q=1}^{Q} PL_{q,n,c} = R_{n-1,c} + \sum_{k=1}^{K} RL_{k,n,c} + EL_{n,c} \qquad \forall c \in COMP. \quad (6)$$

6. Intermediate stage:

$$R_{j,c} + E_{j,c} + \sum_{q=1}^{Q} PL_{q,j,c} = R_{j-1,c} + E_{j+1,c} + \sum_{k=1}^{K} RL_{k,j,c} + EL_{j,c} \; \forall j \in \mathrm{NINT}, \qquad \forall c \in \mathrm{COMP}. \quad (7)$$

7. Bypass mass balance:

$$R_{\mathrm{def},c} = R_{n,c} + R_{0,byp,c} + \sum_{k=1}^{K} RL_{k,byp,c} \qquad \forall c \in \mathrm{COMP} \quad (8)$$

$$R_{0,c} = R_{0,\mathrm{ext},c} + R_{0,byp,c} \qquad \forall c \in \mathrm{COMP}. \quad (9)$$

8. Component and total flow-rate relations (bilinear terms):

$$\left.\begin{array}{ll} R_{j,c} - R_j \cdot x_{j,c} = 0 & E_{j,c} - E_j \cdot y_{j,c} = 0 \\ EL_{j,c} - EL_j \cdot y_{EL,j,c} = 0 & RL_{k,j,c} - RL_{k,j} \cdot x_{RL,k,c} = 0 \\ PL_{q,j,c} - PL_{q,j} \cdot x_{j,c} = 0 & \end{array}\right\} \qquad \begin{array}{l} \forall j \in NT, \quad \forall c \in COMP \\ \forall k \in K, \quad \forall q \in Q \end{array} \quad (10)$$

$$\left.\begin{array}{ll} R_{\mathrm{def},c} - R_{\mathrm{def}} \cdot x_{\mathrm{def},c} = 0 & R_{0,byp,c} - R_{0,byp} \cdot x_{0,c} = 0 \\ R_{0,\mathrm{ext},c} - R_{0,\mathrm{ext}} \cdot x_{0,c} = 0 & RL_{k,byp,c} - RL_{k,byp} \cdot x_{RL,k,c} = 0 \\ PL_{q,c} - PL_q \cdot x_{PL,q,c} = 0 & \end{array}\right\} \qquad \begin{array}{l} \forall c \in COMP \\ \forall k \in K, \quad \forall q \in Q. \end{array} \quad (11)$$

9. Side stream balances:

$$RL_{k,c} = \sum_{j=1}^{n} RL_{k,j,c} + RL_{k,byp,c} \qquad \forall k \in K, \qquad \forall c \in COMP \quad (12)$$

$$PL_{q,c} = \sum_{j=1}^{n} PL_{q,j,c} \qquad \forall q \in Q, \qquad \forall c \in COMP. \quad (13)$$

10. Mass fractions:

$$\sum_{c=1}^{\mathrm{COMP}} x_{j,c} = 1 \qquad \sum_{c=1}^{\mathrm{COMP}} y_{j,c} = 1 \qquad \forall j \in NT \quad (14)$$

$$\sum_{c=1}^{\mathrm{COMP}} x_{\mathrm{def},c} = 1 \qquad \sum_{c=1}^{\mathrm{COMP}} x_{PL,q,c} = 1 \qquad \forall q \in Q. \quad (15)$$

The constraints in Eqs. 1–15 involve only continuous variables, and are valid for any cascade configuration. The following disjunctions in Eq. 16 are the ones associated with the discrete choice of enforcing the equilibrium relations in existing stages. This is accomplished with the Boolean variable Z_j, which can be true or false depending on whether stage j is selected or not. The disjunction is as follows:

$$\begin{bmatrix} Z_j \\ \text{Equilibrium: Eqs. 22, 23} \\ \text{Bilinear terms: Eqs. 10} \end{bmatrix} \vee \begin{bmatrix} & \neg\, Z_j \\ x_{j,c} = x_{j-1,c}; & y_{j,c} = y_{j+1,c} \\ E_j = E_{j+1}; & E_{j,c} = E_{j+1,c} \\ R_j = R_{j-1}; & R_{j,c} = R_{j-1,c} \\ RL_{k,j} = 0; & RL_{k,j,c}, = 0 \\ PL_{q,j} = 0; & PL_{q,j,c} = 0 \\ EL_j = 0; & EL_{j,c} = 0 \end{bmatrix} \qquad \begin{array}{l} \forall j \in NT \\ \forall c \in COMP \\ \forall k \in K \\ \forall q \in Q, \end{array} \quad (16)$$

where the nonlinear equilibrium relations are given as Eqs. A1 and A2 in the Appendix (Reyes et al., 1999).

Because there is the possibility of deleting or deactivating different intermediate stages for the same total number of stages, it is possible to obtain multiple solutions with the same objective function value. To avoid this situation and ensure a unique representation, the following logic constraints are added:

$$Z_j \Rightarrow Z_{j-1} \qquad \forall j \in NINT. \quad (17)$$

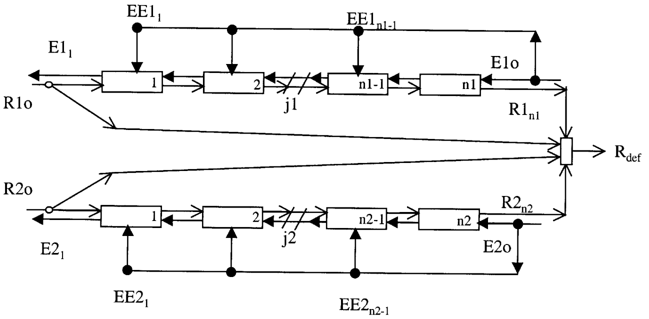

Figure 4. Superstructure of two multistage countercurrent extractors.

The objective function involves the minimization of the total annualized cost of equipment and utilities. For simplicity, the total capital cost is considered proportional to the sum of stages, and the operating cost of the process a function of the flow rate of the solvent feed stream. The objective function of the problem is given by

$$\min OF = \sum_{j=1}^{n} EL_j \cdot C_E + \sum_{j=1}^{n} Z_j \cdot C_n. \tag{18}$$

In order to avoid infeasibilities in the solution of the NLP subproblems with fixed values of the Boolean variables, the design specifications are relaxed using limited slack variables, which are introduced as an additional set of terms in the objective function to be minimized. The weight chosen for the slacks are generally small.

The proposed model can be easily extended to the case of several countercurrent extractors. As an example, Figure 4 shows the superstructure corresponding to two independent countercurrent extractors, where the possibility of fractional bypasses of the initial feed streams and side solvent feed streams are considered.

Obviously, this superstructure can be extended to complex extractor cascades with multiple side feed/product streams, and also with streams that connect each stage of one cascade with all the stages in the other extractors.

Solution Algorithm

The proposed GDP model is solved with a modification of the Logic-Based Outer Approximation (OA) algorithm of Turkay and Grossmann (1996). This decomposition algorithm solves the problem by iterating between reduced NLP subproblems and an MILP master problem. This MILP master problem consists of linear approximations derived at all solution points given for all the NLP subproblems that were previously solved. The NLP subproblem contains only the equations for the terms in the disjunction that are true, and its solution provides an upper bound for the objective function. The master problem predicts a combination of discrete variables, which is optimal for a global linear approximation of the problem. The original MILP master problem, proposed by Turkay and Grossmann (1996), was originally constructed by applying the convex hull formulation (Balas, 1985) to the linearized constraints. But the disaggregation of the variables that is involved in that convex hull formulation greatly increases the number of variables. For this reason, we have used the big-M formulation for the disjunctions in the MILP master problem, although this yields a somewhat weaker lower bound for the LP relaxation.

In this way, the linear constraints in Eq. 16 are formulated using the big-M inequalities as following:

$$\begin{bmatrix} Z_j \\ G(t) = H(t') \end{bmatrix} \vee \begin{bmatrix} \neg Z_j \\ G(t) = S(t') \end{bmatrix}$$

$$\Rightarrow \begin{cases} -M_H \cdot (1 - Z_j) + H(t') \le G(t) \le M_H \cdot (1 - Z_j) + H(t') \\ -M_S \cdot Z_j + S(t') \le G(t) \le M_S \cdot Z_j + S(t'), \end{cases} \tag{19}$$

where $G(t)$, $H(t')$, and $S(t')$ are general linear functions. As an example, we can see the formulation of the raffinate and extract flow rates:

$$\left.\begin{aligned} R_{j-1} - M_R \cdot (1 - Z_j) \le R_j \le R_{j-1} + M_R \cdot (1 - Z_j) \\ E_{j+1} - M_E \cdot (1 - Z_j) \le E_j \le E_{j+1} + M_E \cdot (1 - Z_j) \end{aligned}\right\} \quad \forall j \in NT. \tag{20}$$

The nonlinear constraints in Eq. 16 have to be approximated to linear functions in the MILP master problem. If we define PNLP as the set of all the previously solved NLP subproblems and NLF as the set of all nonlinear functions (bilinear terms and equilibrium relations) in the model, these nonlinear functions can be linearized and therefore replaced by the following big-M linear inequalities in the MILP master problem:

$$\left.\begin{aligned} P_s(\bar{u}^{nl}) + \nabla P_s(\bar{u}^{nl}) \cdot (\bar{u} - \bar{u}^{nl}) \le M_{P_s} \cdot (1 - Z_j) + sv1(P_s^{nl}) \\ P_s(\bar{u}^{nl}) + \nabla P_s(\bar{u}^{nl}) \cdot (\bar{u} - \bar{u}^{nl}) + sv2(P_s^{nl}) \ge -M_{P_s} \cdot (1 - Z_j) \end{aligned}\right\}$$

$$\begin{matrix} \forall s \in NLF, & \forall nl \in \mathrm{PNLP} \\ \forall j \in NT, & \forall c \in \mathrm{COMP} \end{matrix}, \tag{21}$$

where $P_s(\bar{u}^{nl})$ represents the value of the nonlinear functions P_s in all the corresponding previous NLP solutions, and where the vector of variables $\bar{\mu}^{\mathrm{nl}}$ corresponds to $y_{j,c}^{nl}$, $x_{j,c}^{nl}$,$x_{\mathrm{def},c}^{\mathrm{nl}}$, R_j^{nl}, E_j^{nl}, R_{def}^{nl} and so on. Here, M_{P_s} is a sufficiently large number that renders the inequality redundant when $Z_j = 0$. Also, note that in each of the *nl* linearizations a slack variable $sv(P^{nl})$ is introduced to make mathematically feasible all the different linearizations of the same nonlinear equation P in the different points corresponding to the previous NLP solutions.

In order to avoid the *nl* linearizations of each bilinear terms in the MILP master problem, we also tried to replace these linearizations with convex envelopes (McCormick, 1976; Quesada and Grossmann, 1995). These results, however, have not been satisfactory due to the absence of good upper and lower bounds of the corresponding variables.

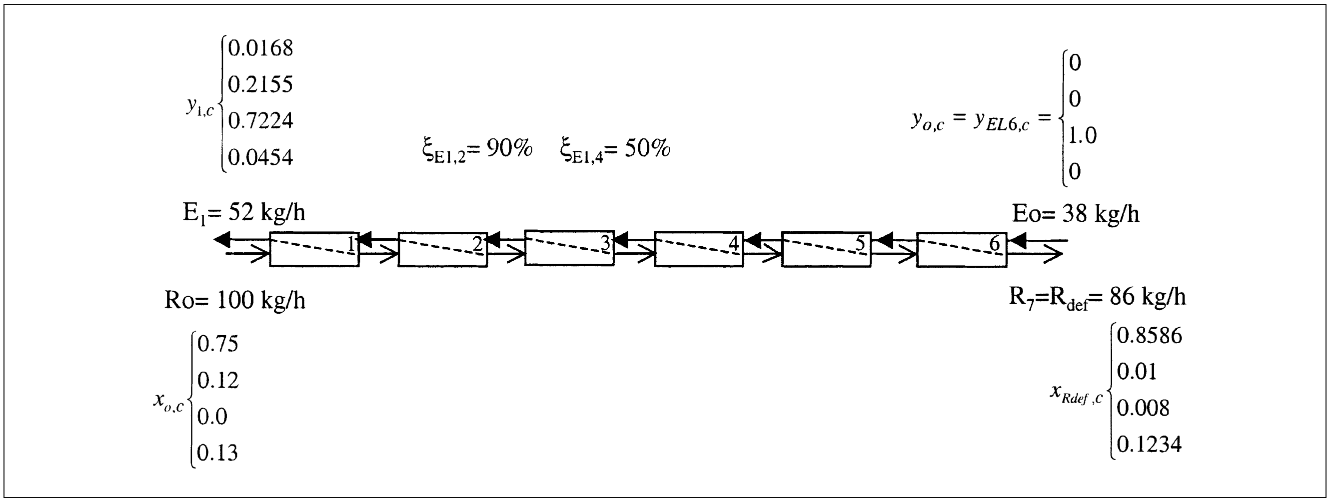

Figure 5. Simple countercurrent extractor cascade.

Numerical Example

The proposed design model was tested with several examples that involve the separation of a quaternary system —water(1)–acetone(2)–chloroform(3)–acetic acid(4)—in a countercurrent extractor. The empirical correlation developed by Reyes et al. (1999) was used to model the liquid–liquid equilibrium of this mixture. In all cases, a maximum number of ten potential stages was postulated for the extraction cascade. The problems were solved on a 667-MhZ Pentium III PC (Linux), using the GAMS modeling system (Brooke et al., 1997). The solver CONOPT was used for the NLP subproblems and the solver CPLEX for the MILP master problems.

Table 1 shows the main statistics of the GDP model and solution time for the different examples. We can see how the size of the model increases with the complexity of the example, and how much bigger the MILP master problem becomes vs. the NLP subproblems (note that the size reported is for 10 stages). Therefore, the major bottleneck is the solution of the MILP master problem. This is due to the presence of binary variables. The NLP subproblems do not contain any binary variables and are therefore not as difficult to solve as the MILP master problems.

Example 1: simple countercurrent extractor

In this case we consider only one aqueous feed stream R_0, and one solvent feed stream E_0, as shown in Figure 1. The feed for this example is a mixture of water (1), acetone (2), and acetic acid (4) in equilibrium with a flow rate of 100 kg/h, using pure chloroform (3) as the solvent. The feed compositions are shown in Figure 5. The required purities for the raffinate product are 0.01 and 0.1 mass fractions of acetone and acetic acid, respectively, for a minimum recovery of 90% of acetone in the organic stream, and a maximum recovery of 50% of acetic acid in the organic product stream E_1.

The results for the optimal design of this example are shown in Figure 5, where it can be seen that six stages were selected, with a flow rate of 38 kg/h for the initial solvent. The cost of this design is $363,579/yr. For comparison we present in Figure 6 the plot of the optimal cost and solvent flow rate for designs with a different number of stages. As can be seen, the optimum is indeed six stages. It is also interesting to see that there is a local optimum at eight stages ($386,385/yr). An important feature of the proposed algorithm is that we do not need to solve the NLP optimization for each number of stages. Other examples that involve complex configurations show more clearly the advantages of the proposed method.

Table 1. Characteristics of the Initial GDP Model for the Different Examples

Example	1	2	3	4	5
NLP subproblem					
No. of equations	503	547	547	953	2,116
No. cont. var.	300	350	350	586	866
CPU time (s)	1.3	2.5	8.3	38	49
MILP master problem					
No. of equations	1,130	1,317	1,317	4,113	6,124
No. cont. var.	2,950	3,800	3,800	11,000	19,196
No. Boolean var.	10	10	10	20	10
CPU Time (s)	69.6	69	80.3	1,503	443
Total CPU time (s)	71	71.5	88.6	1,541	492
Total loops	6	5	4	13	7

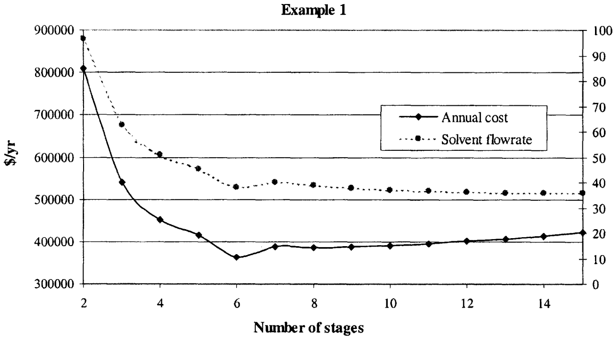

Figure 6. Evolution of the annual cost and solvent flow rate vs. number of stages for Example 1.

It is also worth noting that of all the examples shown in this article, this is the only case that can be solved by traditional graphical methods since it has neither feed/product side streams nor bypasses.

Example 2: countercurrent extractor with one side feed stream

In this case we consider the feed streams *Ro* and *Eo*, and one side raffinate feed stream RL_1 (see Figure 2). The two feed streams *Ro* and RL_1 have a flow rate of 100 kg/h. The specified purities for the raffinate product are mass fractions of acetone(2) and acetic acid(4) of 0.01 and 0.1, respectively, for a minimum recovery of 90% of acetone, and a maximum recovery of 40% of acetic acid in the organic product stream E_1.

The results for this example, which yields an optimal cost of $649,698/yr, are shown in Figure 7 where it can be seen that eight stages were selected, with two side streams that are fed into stages 4 and 5 and correspond to the side feed RL_1. It should be noted that if the possibility of the second feed stream as side stream is not considered and the total feed of 200 kg/h is considered as a single feed (*Ro*), the objective function increases 16% to a cost of $752,837/yr.

Example 3: countercurrent extractor with one side product stream

In this example we consider the feed streams *Ro* and *Eo*, and one side raffinate product extraction PL_1. In this case, the feed *Ro* has a flow rate of 275 kg/h, and the purity specifications for the raffinate product are 0.01 and 0.1 mass fractions of acetone(2) and acetic acid(4), respectively, for a minimum recovery of 90% of acetone and a maximum recovery of 40% of acetic acid in the organic product stream E_1. The specifications for the side raffinate product PL_1 are a flow rate of 100 kg/h, a purity $\tau_{PL_{1,2}} = 0.04$ and a recovery $\xi_{PL_{1,2}} = 5\%$.

The results of this problem, which yields an optimal cost of $954,309/yr, are shown in Figure 8 where it can be seen that six stages were selected, with two side extractions of the product in stages 5 and 6. If the possibility of the fractional extraction is excluded, the cost increases by 3% (that is, to $982,938/yr).

Example 4: two independent countercurrent extractors

In this example, two independent countercurrent extractors with their corresponding feed streams *Ro* and *Eo* are considered. The feed streams $R1o$ and $R2o$ have a flow rate of 189 and 100 kg/h, respectively, with the followings compositions (mass fraction):

$R1o$: water 0.59: acetone 0.29: acetic acid 0.12

$R2o$: water 0.70: acetone 0.17: acetic acid 0.13

The required purities for the raffinate product are mass fractions of acetone and acetic acid of 0.01 and 0.1, respectively, for a minimum recovery of 90% of acetone and a maximum recovery of 40% of acetic acid in the organic product stream E_1.

The optimal design shown in Figure 9 has a cost of $1,101,583/year and involves eight stages in both independent cascades.

Example 5: two dependent countercurrent extractors

In this example we consider two interrelated countercurrent extractors with their corresponding feed streams *Ro* and *Eo*. The feed streams (composition and flow rates) are the same as in the previous example. In this case, the required purities for the raffinate product are 0.01 and 0.1 mass fractions of acetone(2) and acetic acid(4), respectively, for a minimum recovery of 90% of acetone and a maximum recovery of 40% of acetic acid in the organic product stream E_1.

The results for this example are shown in Figure 10, where four stages were selected for the first extractor and five for the second one. Also, we can see that this configuration is

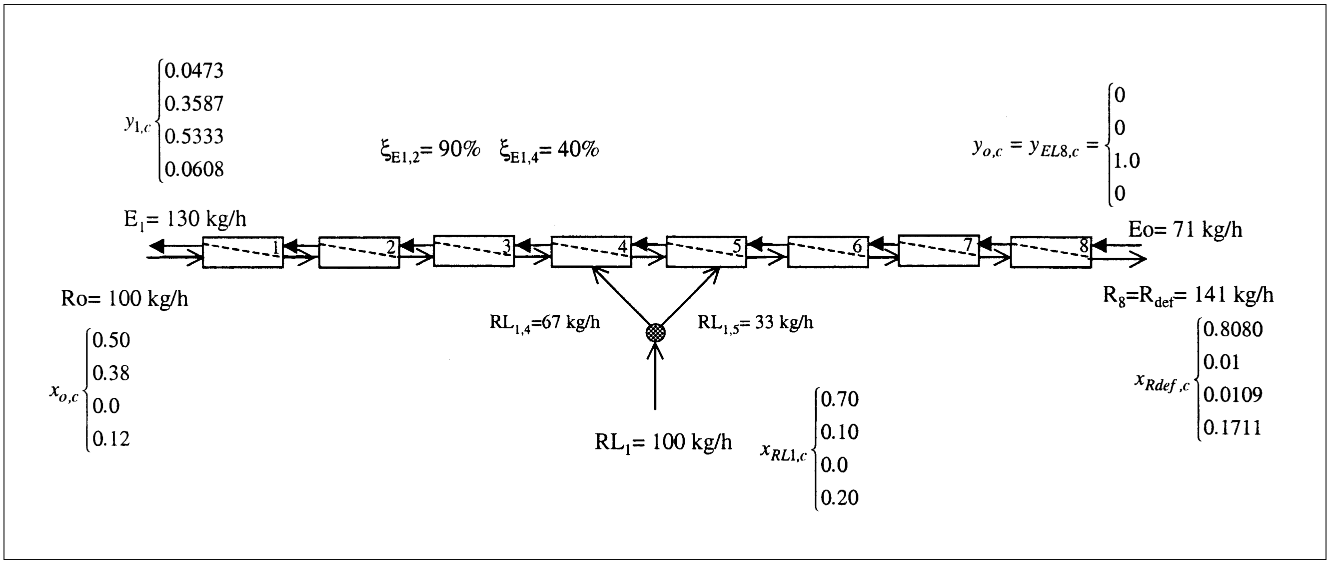

Figure 7. Countercurrent extractor cascade with one side of feed stream.

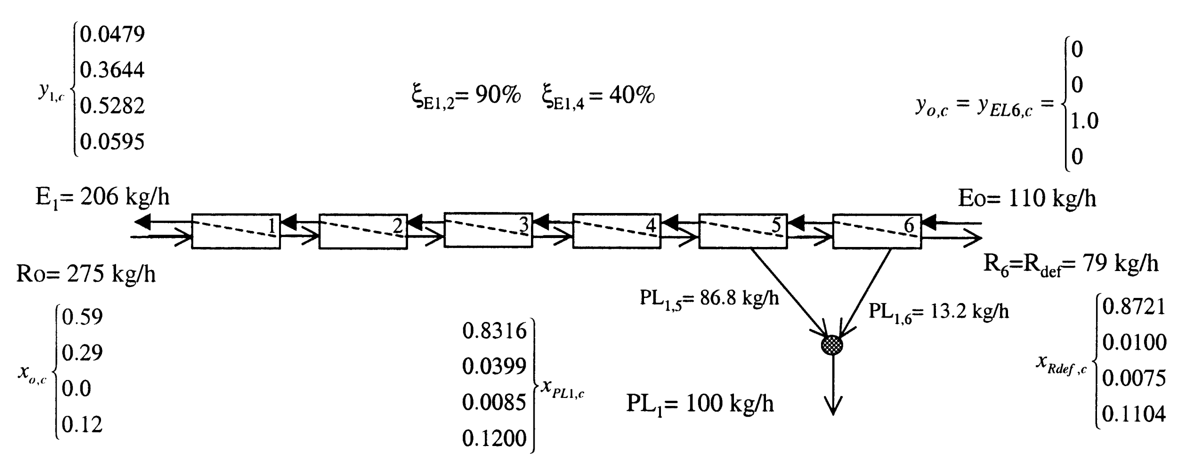

Figure 8. Countercurrent extractor cascade with one side of product stream.

equivalent to having only one extractor with nine stages where the second feed is introduced in stage 3. As opposed to the design in Figure 9, the optimal cost of this design is \$962,129/yr, showing that significant improvement can be attained by taking into account interactions between the two extractors.

Conclusions

This article has presented a new optimization model for the synthesis of complex countercurrent extractors and separation sequences that simultaneously optimizes the design and operating parameters of the cascade for quaternary liquid–liquid systems. Two major superstructures were proposed and formulated as generalized disjunctive programming (GDP) models. A modified Logic-Based Outer Approximation (OA) algorithm was used to solve these models.

The application of the proposed model was illustrated with five examples, which have shown that significant economic savings can be achieved. These examples also have been used to evaluate the robustness and performance of the proposed method for simple, complex, and multiple extraction cascades. The results showed that the method can produce systematically optimal designs, and is computationally efficient and robust, especially when compared to standard MINLP models. This robustness is a result of the disjunctive-stage

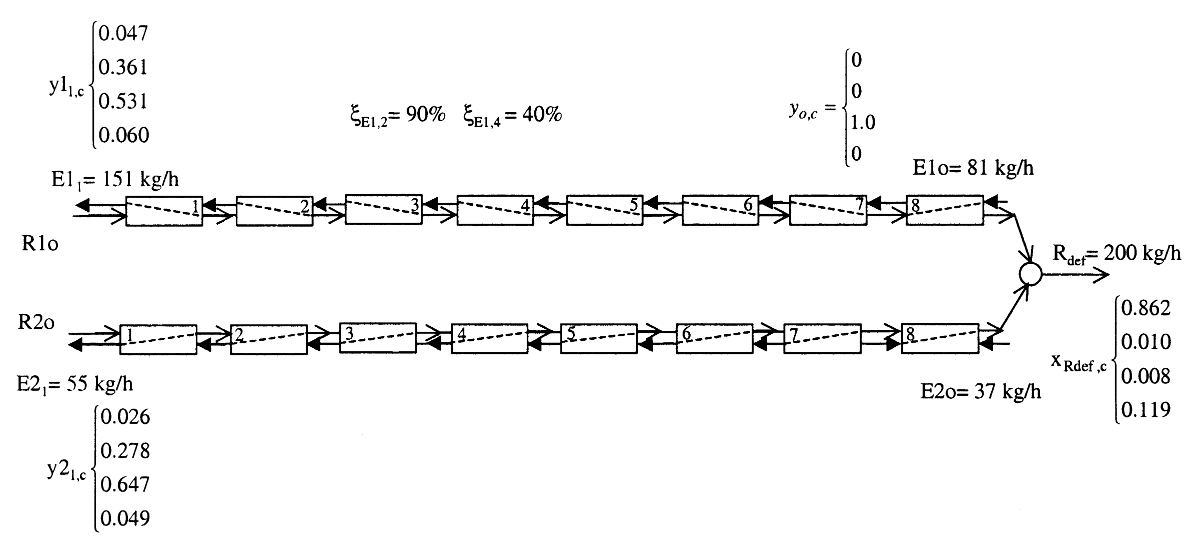

Figure 9. Two independent countercurrent extractor cascades.

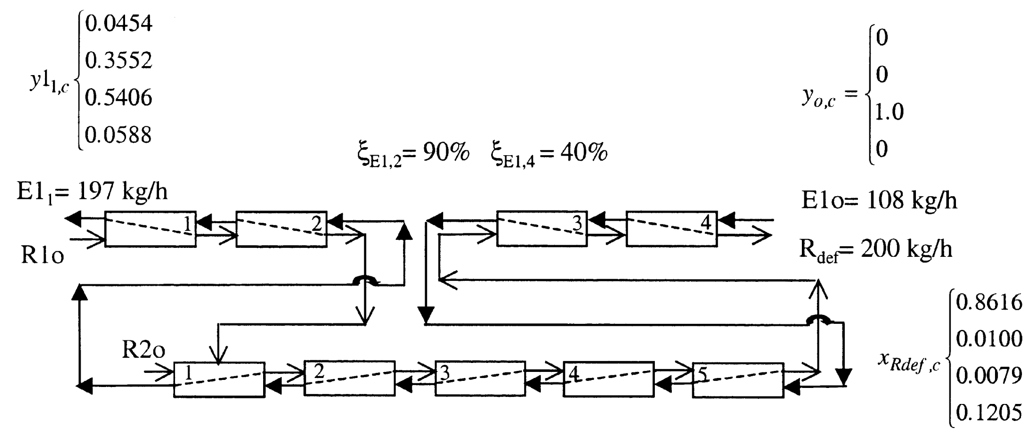

Figure 10. Two interrelated countercurrent extractor cascades.

representation, as well as of the size of the NLP subproblems that are solved when using the Logic-Based OA algorithm.

Acknowledgments

The authors acknowledge financial support from the Conselleria d'Educacio-Generalitat Valencia of Spain (POST00-11-89) and from the Center of Advanced Process Decision-making (CAPD) at Carnegie Mellon University (USA).

Notation

Parameters

PL_q = mass flow of the side raffinate product stream *q*
R_0 = mass flow of the main feed stream introduced at the extractor
RL_k = mass flow of the initial side feed stream *k* introduced into the extractor
$x_{0,c}$ = mass fraction of the component *c* in feed stream R_0
$x_{RL,k,c}$ = mass fraction of the component *c* in side feed stream RL_k
$y_{EL,j,c}$ = mass fraction of the component *c* in solvent side feed stream EL_j

Continuous variables

E_j = mass flow of the extract phase that leaves stage *j*
$E_{j,c}$ = component *c* mass flow of the extract phase that leaves stage *j*
EL_j = mass flow of the side solvent feed introduced at the stage *j*[EL_n is equivalent to E_0 (initial mass flow of the solvent feed)]
$PL_{q,j}$ = mass flow of the side (raffinate) product stream *q* that leaves stage *j*
R_j = mass flow of the raffinate phase that leaves stage *j*
R_{def} = mass flow of the final raffinate phase that leaves the extractor
$RL_{k,byp}$ = mass flow of the side feed stream *k* that bypasses the extractor
$RL_{k,j}$ = mass flow of the side feed stream *k* introduced into the extractor at stage *j*
$R_{0,\text{ext}}$ = mass flow of the main feed stream introduced at the first stage
$R_{0,byp}$ = mass flow of main feed stream that bypasses the extractor
$x_{j,c}$ = mass fractions of the component *c* in the raffinate phase that leaves stage *j*
$x_{def,c}$ = mass fraction of the component *c* in the final raffinate phase that leaves the extractor
$x_{PL,q,c}$ = mass fraction of the component *c* in the side raffinate product stream PL_q
$y_{j,c}$ = mass fractions of the component *c*, in the extract phase that leaves stage *j*

Literature Cited

Balas, E., "Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems," *SIAM J. Algebraic Discrete Methods*, **6**, 466 (1985).

Barnes, N. G., M. B. Gramajo de Doz, and H. N. Solimo, "Liquid-Liquid Extraction of trans-Aconitic Acid from Aqueous Solutions with Tributyl Phosphate and a Mixed Solvent at 303.15 K," *Ind. Eng. Chem. Res.*, **39**, 3364 (2000).

Barnes, N. G., M. B. Gramajo de Doz, and H. N. Solimo, "Liquid-Liquid Extraction of Oxalic Acid from Aqueous Solutions with Tributyl Phosphate and a Mixed Solvent at 303.15 K," *J. Chem. Eng. Data*, **44**, 430 (1999).

Brooke, A., D. Kendrick, A. Meerhaus, and R. Raman, *GAMS Release 2.25 Language Guide*, Version 92, Gams Development Corporation, Washington, DC (1997).

Fidkowski, Z. T., M. F. Malone, and M. F. Doherty, "Nonideal Multicomponent Distillation: Use of Bifurcation Theory for Design," *AIChE J.*, **37**, 1761 (1991).

Grossmann, I. E., "Mixed-Integer Optimization Techniques for Algorithmic Process Synthesis," *Adv. Chem. Eng. Process Synth.*, **23**, 171 (1996).

Marcilla, A., A. Gómez, and J. Reyes, "New Method for Designing Distillation Columns of Multicomponent Mixtures," *Lat. Amer. Appl. Res.*, **27** (1-2), 51 (1997).

Marcilla, A., A. Gómez, J. A. Reyes, and M. M. Olaya, "New Method for Quaternary Systems Liquid-Liquid Extraction Tray to Tray Design," *Ind. Eng. Chem. Res.*, **38**, 3083 (1999).

McCormick, G. P., "Computability of Global Solutions to Factorable Nonconvex Programs, Part I. Convex Underestimating Problems," *Math. Prog.*, **10**, 146 (1976).

Minottim, M., M. F. Doherty, and M. F. Malone, "A Geometric Method for the Design of Liquid Extractors," *Ind. Eng. Chem. Res.*, **35**, 2672 (1996).

Minotti, M., M. F. Doherty, and M. F. Malone, "Economic Tradeoffs for Extraction Systems," *Trans. Inst. Chem. IChem. Eng.*, **76**, 361 (1998).

Quesada, I., and I. E. Grossmann, "Global Optimization of Bilinear Process Networks with Multicomponent Flows,"*Comp. Chem. Eng.*, **12**, 1219 (1995).

Raman, R., and I. E. Grossmann, "Modeling and Computational Techniques for Logic Based Integer Programming,"*Comput. Chem. Eng.*, **18**, 563 (1994).

Reyes, J. A., M. M. Olaya, A. Gómez, and A. Marcilla, "Calculation of Liquid-Vapour and Liquid-Liquid Equilibrium in Multicomponent Systems Using Correlations of Equilibrium Data," *Equifase'99, V Iberoamerican Meeting. Phase Equilibrium for Process Design*, Vigo, Spain (1999).

Reyes-Labarta, J. A., "Design of Multicomponent Rectification and Extraction Column. Minimum Reflux Calculation," PhD Diss., Univ. of Alicante, Alicante, Spain (1998).

Seader, J. D., and E. J. Henley, *Separation Process Principles*, Wiley, New York (1998).

Thornton, J. D., *Science and Practice of Liquid-Liquid Extraction*, Oxford Univ. Press, New York (1992).

Treybal, R. E., *Liquid Extraction*, McGraw-Hill, New York (1963).

Tsuboka, T., and T. Katayama, "General Design Algorithm Based on Pseudoequilibrium Concept for Multistage Multicomponent Liquid-Liquid Separation Processes," *J. Chem. Eng. Jpn.*, **9**, 40 (1976).

Turkay, M., and I. E. Grossmann, "Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Networks," *Comput. Chem. Eng.*, **20**, 959 (1996).

Wankat, P. C., *Equilibrium Stage Separation*, Elsevier, New York (1988).

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

Appendix: Equilibrium Correlations

The correlation of the tie-lines and solubility surface used in this article to calculate the liquid–liquid equilibrium is the one proposed by Reyes et al. (1999) for quaternary systems, whose parameters are obtained by optimization from a set of experimental equilibrium data. The corresponding equations for each stage j, are as follow.

1. Tie-line correlation:

$$\log\left(\frac{y'_{j,k}}{y'_{j,p}}\right)-\left\{\left[a_{k,p}+b_{k,p}\cdot\left(\frac{x'_{j,4}}{x'_{j,2}}\right)+c_{k,p}\cdot\left(\frac{x'_{j,4}}{x'_{j,2}}\right)^2\right]+\left[d_{k,p}+e_{k,p}\cdot\left(\frac{x'_{j,4}}{x'_{j,2}}\right)+f_{k,p}\cdot\left(\frac{x'_{j,4}}{x'_{j,2}}\right)^2\right]\cdot\log\left(\frac{x'_{j,2}}{x'_{j,1}}\right)+\left[g_{k,p}+h_{k,p}\cdot\left(\frac{x'_{j,4}}{x'_{j,2}}\right)^2+i_{k,p}\cdot\left(\frac{x'_{j,4}}{x'_{j,2}}\right)^2\right]\cdot\left[\log\left(\frac{x'_{j,2}}{x'_{j,1}}\right)\right]^2\right\}=0. \tag{A1}$$

Table A1. Parameters of the Tie-Line Correlation

Parameters	$\log(y'_{j,2}/y'_{j,3})$	$\log(y'_{j,2}/y_{j,2}/y'_{j,1})$	$\log(y'_{j,3}/y'_{j,4})$
a	1.08418344	0.97697805	0.31539557
b	−1.69670841	−1.43680314	−0.50174614
c	0.54918114	0.47827271	0.23025919
d	10.6191185	6.36524047	−1.26380234
e	−17.4490159	−11.6820482	−0.47501422
f	6.64954207	4.31160926	0.74088285
g	24.810838	12.7737816	−5.00508651
h	−43.3874678	−24.3262345	3.58436122
I	15.4583387	8.06761518	0.92388234

Table A2. Parameters of the Solubility Surface Correlation (Aqueous Region)

$x'_{j,3}+x'_{j,4}$		$x'_{j,2}+x'_{j,4}$			
A'	−0.56166692	A	−0.0002179	G	−100.019097
B'	−0.7922085	B	−0.00181492	H	−74.7854131
C'	638.307294	C	0.00208734	I	352.362963
D'	202.120475	D	0.3966406	J	0.00814773
		E	0.80055132	K	−0.03665672
		F	−1.49226664	L	−27,822.0805

2. Solubility surface correlation:

$$x'_{j,3}+x'_{j,4}-\left\{\left(A'\cdot M_j+B'\right)\cdot\left(x'_{j,1}+x'_{j,4}\right)+\left(C'\cdot M_j+D'\right)\right\}=0 \tag{A2a}$$

$$x'_{j,2}+x'_{j,4}-\left\{\left(A\cdot M_j^2+B\cdot M_j+C\right)\cdot\left(x'_{j,1}+x'_{j,4}\right)^3+\left(D\cdot M_j^2+E\cdot M_j+F\right)\cdot\left(x'_{j,1}+x'_{j,4}\right)^2+\left(G\cdot M_j^2+H\cdot M_j+1\right)\cdot\left(x'_{j,1}+x'_{j,4}\right)+\left(J\cdot M_j^2+K\cdot M_j+L\right)\right\}=0 \tag{A2b}$$

$$M_j=\frac{x'_{j,4}}{x'_{j,2}+x'_{j,4}}, \tag{A2c}$$

where $x'_{j,c}=x_{j,c}+CC$: corrected mass fraction of the c component in raffinate phase that leaves stage j, with $\sum_{c=1}^{4}x'_{j,c}=1+4\cdot CC$; $y'_{j,c}=y_{j,c}+CC$: corrected mass fraction of the c component in extract phase that leaves stage j, with $\sum_{c=1}^{4}y'_{j,c}=1+4\cdot CC$; k and p correspond to two different components; $a_{k,p}$, $b_{k,p}$, $c_{k,p}$, $d_{k,p}$ $\cdots$ A', B', C', A, B, $C\cdots$: adjustable parameters of the correlation that depend on the components and temperature, but are independent of the composition; and CC: positive constant, to allow the use of the binary and ternary equilibrium data in the proposed correlation ($CC=100$).

The set of these equations represents the sufficient and necessary conditions that guarantee that $x_{j,c}$ and $y_{j,c}$ are compositions in equilibrium. This means that they are on the solubility surface and that they are extremes of a tie-line.

In order to avoid infeasibilities in the solution of the NLP subproblems that are largely due to the complex nature of the liquid–liquid equilibrium calculations, the solubility surface constraints were relaxed through the use of slack variables, which are minimized in the objective function.

Manuscript received Dec. 21, 2000, and revision received Apr. 2, 2001.