Collocation Methods for Distillation Design III: Flexible Column Design

Robert Huss

EDRC 06-198-95

COLLOCATION METHODS FOR DISTILLATION DESIGN III: FLEXIBLE COLUMN DESIGN

Robert S. Huss and Arthur W. Westerberg

Abstract

In this third paper on collocation methods for distillation design, we explore the use of the collocation models for design of simple distillation columns as well as flexible columns. Solvent recovery plants must deal with a wide range of feeds and still return pure solvents. The design problem we address is a single flexible column within the overall solvent recovery plant.

We have developed the models and algorithms in the ASCEND system. We discuss the attributes and use of the ASCEND system. With ASCEND we can create complex models with simple building blocks and interactively learn to solve them.

We found the collocation model an excellent tool for distillation design, allowing us to develop new concepts in design strategies. We designed a single column as would exist in a flexible solvent recovery plant for an azeotropic system. It was designed to handle three possible feeds, each with a distinct separation task. For each possible feed to a column, we approximate the operation of the column for that feed by creating a quadratic approximation of the reflux ratio as a function of the number of trays and feed location. We optimize the cost of the column over the approximation range, and reapproximate if the minimum is on a bound. We move the approximation range until the local optimum occurs

This work has been partially supported by the Engineering Design Research Center, a NSF Engineering Research Center.

Introduction

Several researchers have explored and developed collocation for distillation column modeling. In the first paper in this series, we presented a collocation model which expands on prior models, addressing the problems specific to steady-state, continuous columns [Huss and Westerberg, 1995a]. In the second paper, we discussed the use of the collocation method for minimum reflux determinations [Huss and Westerberg, 1995b]. In this paper, we present algorithms for designing distillations columns with the collocation model, including methods for designing flexible distillation columns. We also discuss the benefits of the ASCEND system for developing and using mathematical models.

Learning to Solve Models

A mathematical model begins as a set of equations and variables. However, formulation is only a small part of modeling and design. For complex models, solving is much harder than creating. We do not always know what parts of the model will be known and what parts need to be solved for. Also, we need to learn the best path to solution. The ASCEND system supports these needs. ASCEND (Advanced System for Computations in ENgineering Design) is a rapid model development tool, which has a strongly typed declarative modeling language and incorporates object-oriented concepts [Piela et al., 1993]. It also has an interactive model building and solving environment, which creates a great deal of flexibility in model use.

With ASCEND, we can build a simple flash model from stream models and a vapor-liquid equilibrium (VLE) model. We can then model a distillation column with an array of flash models. This hierarchical building of models aids in creating and organizing complex constructs. ASCEND also allows us to refine existing models to include more complexity.

We typically solve a normal stage-by-stage distillation model with the following steps. The initial model has no thermodynamics, simple stream models with molar flowrates and mole fractions, constant relative volatility for the VLE, and constant molar overflow. This is generally a simple type of model to solve, but if it is difficult, we can solve each tray individually first. Any part of an ASCEND model is accessible from the user interface. Once we solve the simple column, we can "refine" the simple stream models to liquid and vapor stream models and include enthalpy and Gibbs free energy calculations. To refine the model we use the deferred binding capability of ASCEND. We can interactively locate any part of a compiled (and perhaps solved) model and direct the type of the part be changed to a more refined type compatible with its current type. The compiler reinterprets and propagates the changes that result throughout the model. Values of variables already solved for become initial values for the modified model. Holding the temperatures constant and still using constant relative volatility for the equilibrium, we calculate initial values for the thermodynamic properties. Then the VLE can be refined to a thermodynamic equilibrium model, requiring the partial molar Gibbs free energies of the components in the liquid and vapor phase to be equal. The last step is to release the constant molar overflow specification and include heat balances instead. Using this incremental process, we can get a converged solution of a complex thermodynamic distillation model without trying to solve the whole thing from scratch.

By formulating the collocation model in ASCEND, we achieved a great deal of flexibility, as well as the ability to incrementally refine the model to include complex thermodynamics. We describe the details of this model in the first paper in this series.[Huss and Westerberg, 1995a] When creating a test model, we refine a generic collocation model and decide the number of collocation sections, the number of trays in each section, and the number of components. Interactively, we can change the degrees of freedom for the column, the direction for number staged (up or down) of each collocation section, and the spacing of collocation sections and trays.

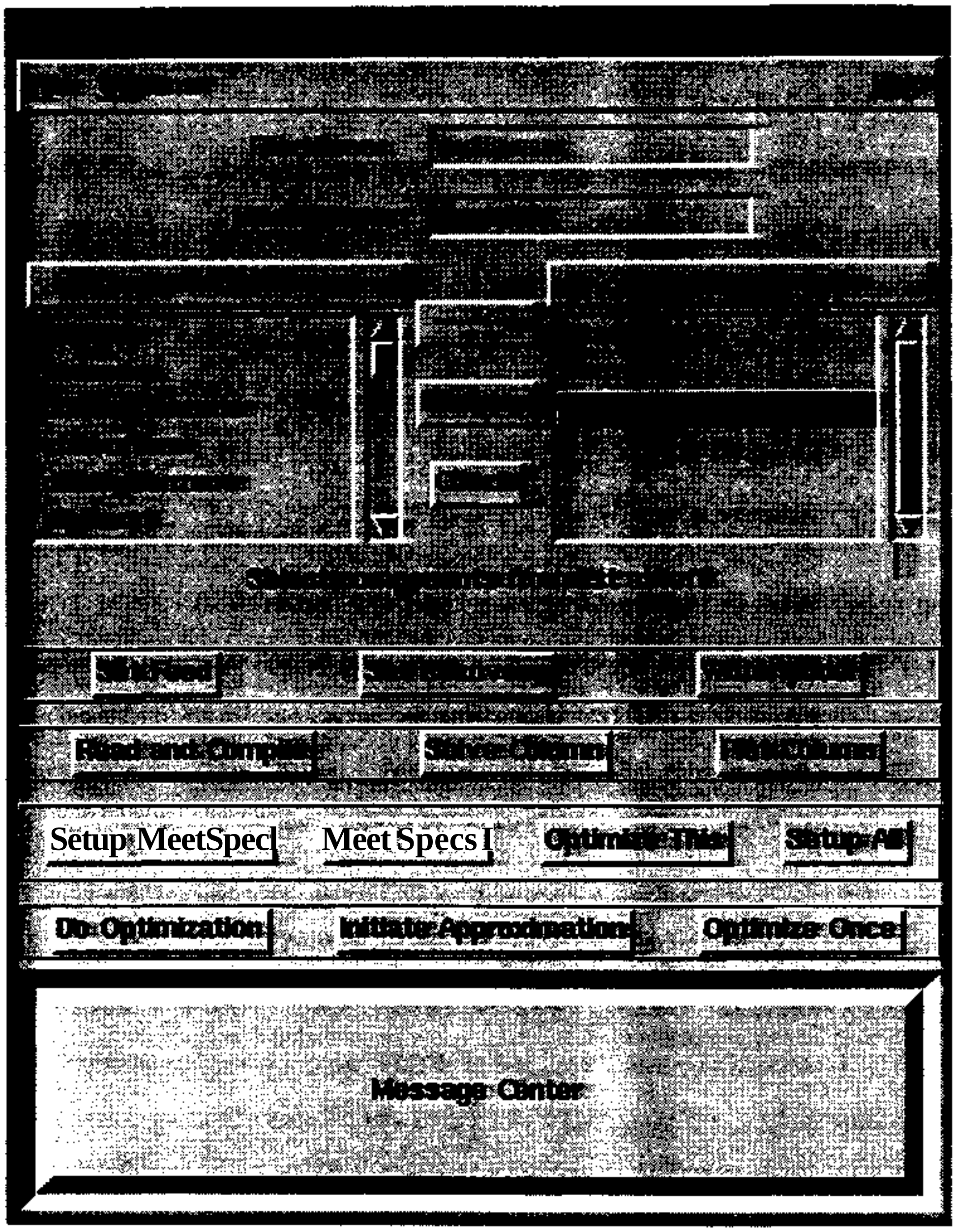

Figure 1. Column design application

ASCEND also has a scripting language with simple commands which reproduce interactive actions as well as the capability for developing complex procedures to automate the simulation process. We have used this scripting language to develop solution and design algorithms. This language also allows us to add to the graphical interface, creating our own applications which can run on top of the ASCEND system. Figure 1 is a picture of the interactive window for the application we created to design columns.

Design Methodology

We learned the following general strategy for designing a column given the feed and a required separation. We present the algorithm in Figures 2 and 3. Figure 2 shows the algorithm for getting to a rigorous column model for a specified feed, making no demands on the separation. Figure 3 shows the algorithm for getting from this starting point to a column which meets the design specifications. The steps in this algorithm are executed by the first two rows of buttons on the interactive window shown in Figure 1.

The algorithm in Figure 2 begins by getting information about the problem. We need to decide what components will be used, what the feed flowrate and composition will be, and what the recovery specifications will be. We then check on the nonideality of the components involved. The check is based on a database of infinite dilution K values for all the components in our library. From these values, we can guess if binary azeotropes are expected and if heterogeneous behavior is expected. If the components in the system will exhibit heterogeneous behavior, we stop because we currently cannot accurately model this behavior. If azeotropes are expected without heterogeneous behavior, we can model the system, but we need to be aware of distillationi>oundaries. After checking the components, we generate a standard collocation model.

For a single feed column, this model has four collocation sections, two above and two below the feed, with two trays in each collocation section. [Huss and Westerberg, 1995a] We start with only two trays in each half of the column and a reflux ratio of 0.5, so the initial models will solve even for systems with high relative volatilities. We also start with the variable transformations on stage number and mole fraction (described in Huss and Westerberg, 1995a) turned on,

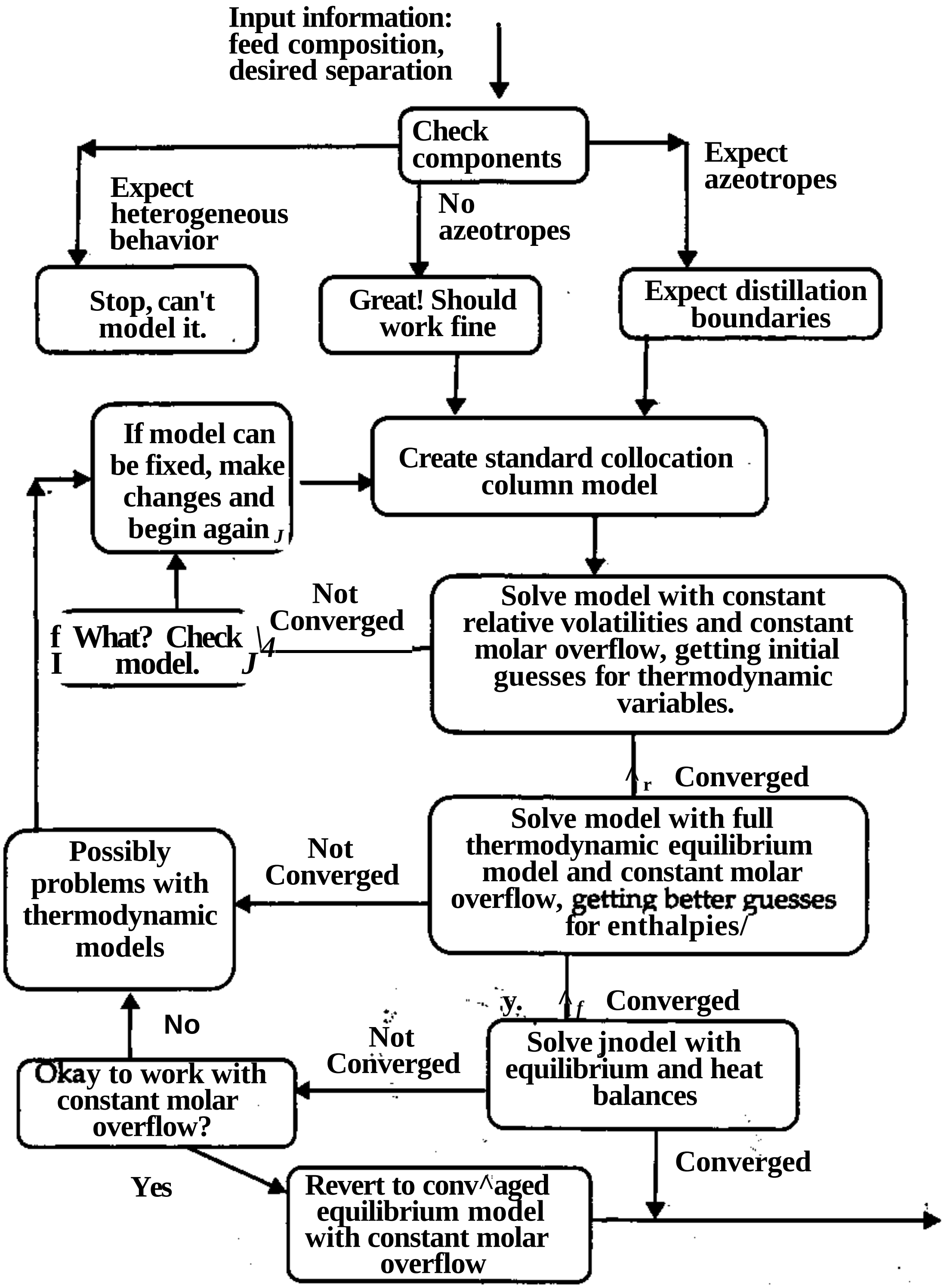
Input information:
feed composition,
desired separation
Check components
Expect heterogeneous behavior
No azeotropes
Expect azeotropes
Stop, can't model it.
Great! Should work fine
Expect distillation boundaries
If model can be fixed, make changes and begin again
Create standard collocation column model
Not Converged
What? Check model.
Solve model with constant relative volatilities and constant molar overflow, getting initial guesses for thermodynamic variables.
Converged
Possibly problems with thermodynamic models
Not Converged
Solve model with full thermodynamic equilibrium model and constant molar overflow, getting better guesses for enthalpies
Converged
No
Not Converged
Okay to work with constant molar overflow?
Solve model with equilibrium and heat balances
Converged
Yes
Revert to converged equilibrium model with constant molar overflow

Figure *2.* Algorithm for creating and converging a collocation model since they may be needed and have no negative effects.

Once we create the model, we compile it and solve it with constant relative volatilities and constant molar overflow, fixing the reflux ratio, distillate flowrate, and number of trays top and bottom. From this solution we get initial guesses for the thermodynamic variables at standard conditions. This model has converged for all the systems we have tested. We next solve the model with the equilibrium equations included but still holding constant molar overflow. Then we turn on the heat balance and solve the fully rigorous model. If the heat balanced column does not converge, we can either return to the equilibrium model and gradually approach the heat balanced column by incrementally reducing the heat duties on each tray to zero vising an algebraic continuation method, or we can decide to go ahead with the constant molar overflow column.

If any of the models with thermodynamics do not converge, it is possible that the thermodynamic models themselves are at fault. The collocation model and design algorithms are not dependent on what particular thermodynamics we use. If the thermodynamic models cause failure, we suggest changing them, but that is not the focus of this paper.

Once we have converged a sufficiently rigorous model of the system, we turn our attention to meeting the product specifications. Figure 3 shows that the first step is to adjust the degrees of freedom, fixing the recoveries of the keys and freeing the reflux ratio and distillate flowrate. Also, we define average slopes over the collocation sections, which can be used to maintain a reasonable number of trays for a given reflux ratio. Figure 4 shows how we define the average slope. In this case, the light component changes from 0.95 at the distillate to 0.35 at the feed tray, over a range of 50 trays. The average slope is therefore 0.012. If we fix the average slope for one component and free up the number of trays, the number of trays will change to maintain that slope if we change other variables. This allows us to avoid the problem of determining how many trays the column might

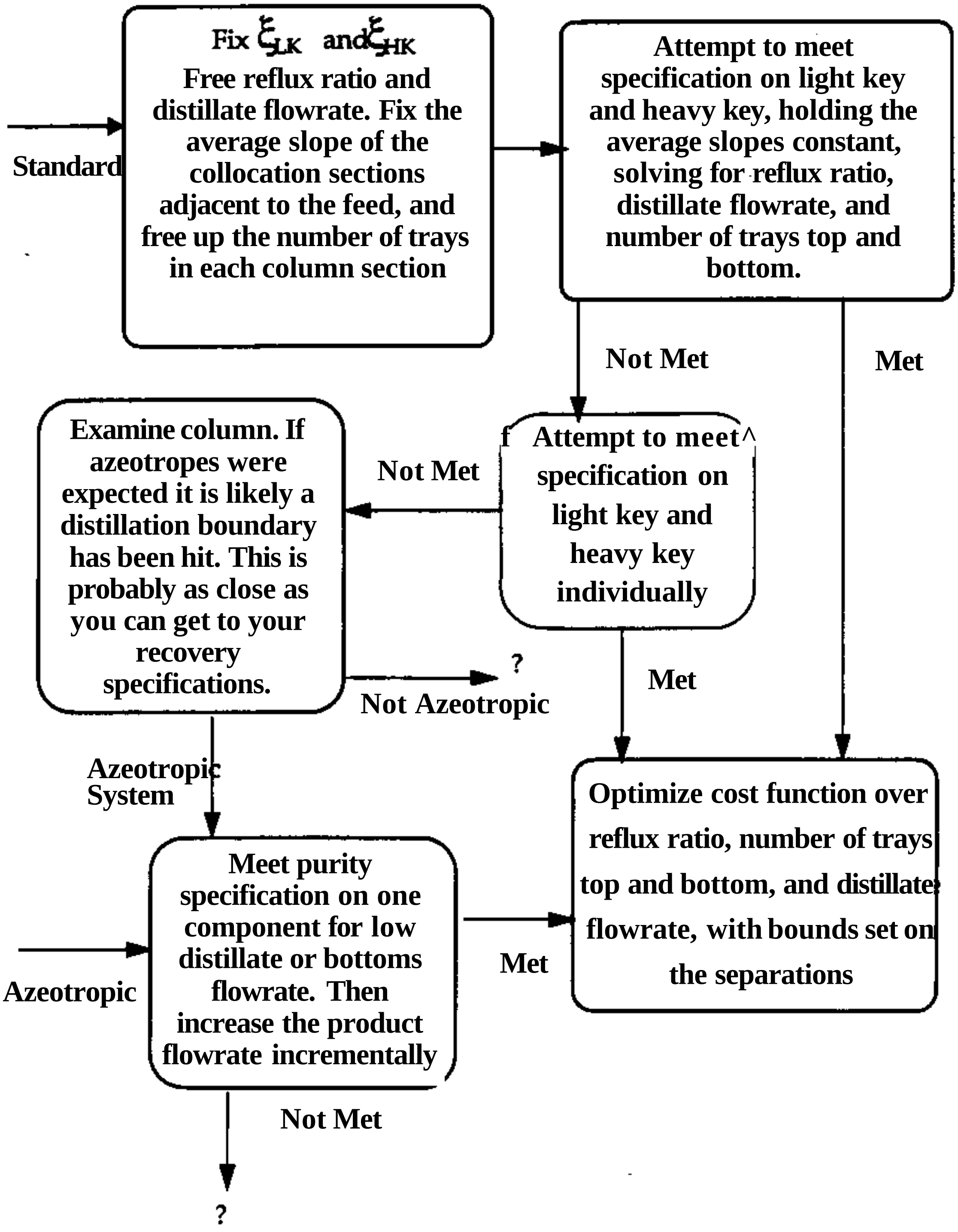
Fix ξ_{LK} and ξ_{HK}
Free reflux ratio and distillate flowrate. Fix the average slope of the collocation sections adjacent to the feed, and free up the number of trays in each column section
Standard
Attempt to meet specification on light key and heavy key, holding the average slopes constant, solving for reflux ratio, distillate flowrate, and number of trays top and bottom.
Not Met
Met
Examine column. If azeotropes were expected it is likely a distillation boundary has been hit. This is probably as close as you can get to your recovery specifications.
Not Met
f Attempt to meet^ specification on light key and heavy key individually
?
Not Azeotropic
Met
Azeotropic System
Optimize cost function over reflux ratio, number of trays top and bottom, and distillate flowrate, with bounds set on the separations
Meet purity specification on one component for low distillate or bottoms flowrate. Then increase the product flowrate incrementally
Azeotropic
Met
Not Met
?

Figure 3. Algorithm for designing a column to meet specifications

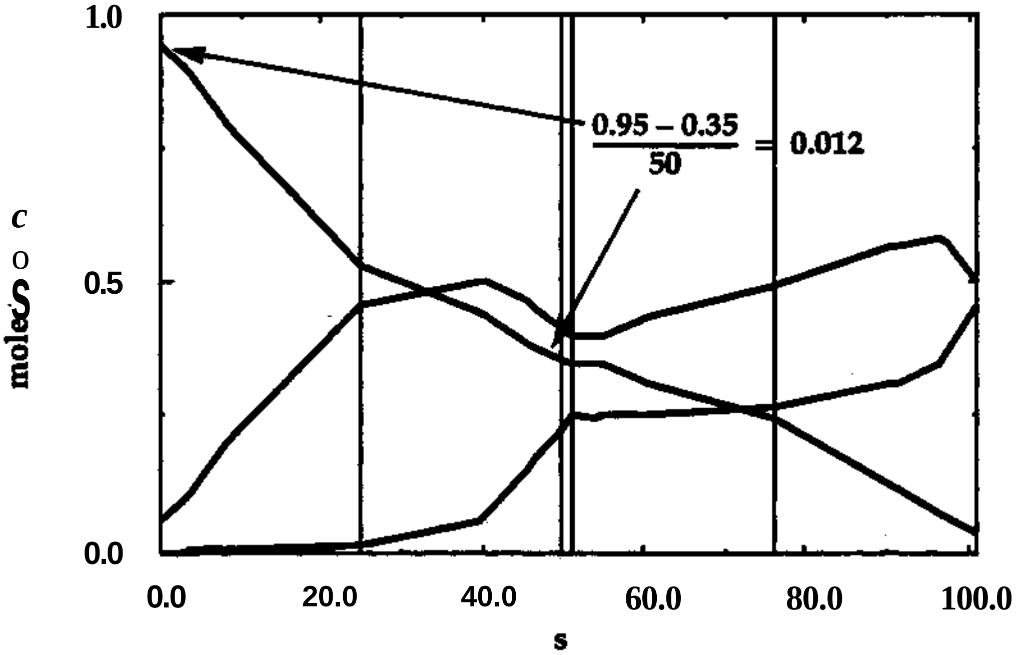

Figure 4. Average slope definition

need for a particular separation. For any fixed number of trays, it may be possible to solve for the reflux ratio and distillate flowrate for a given specification, but if there are not enough trays it will be impossible. By fixing the average slope of one component over each column section, the number of trays will increase as we approach the recovery specification.

In the next step, we attempt to meet the specifications on both key components while also meeting the requirement that the average slopes are 0.01/ tray, which is a reasonable value from our experience. We approach the specifications with an algebraic continuation method, taking a fractional step towards the desired settings. If the step succeeds, we take another step of the same size. If the step fails, we take a smaller step. If the four specifications (two key components and two slopes) cannot be met simultaneously, but the number of trays has increased, we fix the number of trays and just try to meet the two recovery specifications. Another alternative is to hold the slopes constant rather than trying to move them to 0.01/tray. If the specifications cannot be met simultaneously, we then try to meet them each individually. If the specifications still cannot be met, we check to see if azeotropes were expected. If they were, it is likely that a distillation boundary has been hit.

We have an alternative for azeotropic systems where we expect to hit distillation boundaries. When we know that we want a certain purity of key components but cannot get high recoveries or do not know what the recoveries will be, we can meet purity specifications instead. If, for example, we want 99% purity of the light component in the distillate, we can first meet this purity specification with a very low distillate flowrate, solving for the reflux ratio. We will still use the slope criteria for setting the number of trays in each column section. Once we have met the purity specification, we can incrementally increase the distillate flowrate while holding the purity constant.

If the specifications can be met, we can attempt to optimize the column for cost. We have a cost function, accounting for the capital cost of the column, condenser, and reboiler, and the heat duties of both exchangers without heat integration [Douglas, 1988]. Currently, we have failed to get MINOS attached to ASCEND to solve the optimization problem for a fully thermodynamic column, due perhaps to the nonlinearity of the thermodynamic models and possibly poor variable bounding, but we can approach an optimal cost for a specified separation by doing a simple gradient based optimization routine using an ASCEND script. By specifying the recovery of both the key components and fixing the number of trays top and bottom, we can solve for the reflux ratio and distillate flowrate and the cost. By perturbing the number of trays and the feed location and resolving, we determine a slope on the cost function and take a step to reduce it. We have developed a simple routine in the script to minimize the cost over the number of trays and the feed tray location.

Design Essentials and Tricks

While developing this design algorithm, we discovered certain essential techniques and discovered some tricks that significantly improve our algorithm. The first essential lesson is that we cannot simply specify a product purity and solve for an appropriate reflux ratio. Even when there is a solution, this is a difficult problem to converge and must be approached gradually. With the standard degrees of freedom: feed, reflux ratio, distillate flowrate, number of trays, it is difficult to make large changes in any of these variables. It is even more difficult to switch the degrees of freedom for product purities or recoveries and make large changes. We have also discovered that it is better to make both key component specifications on the recoveries and release the reflux ratio and distillate flowrate than to just make one purity specification and release either the reflux ratio or the distillate flowrate. It is very easy to make purity specifications that are impossible with either the reflux ratio or distillate flowrate fixed. When we approach both key component recoveries simultaneously and solving for reflux and distillate flowrate, we give the model more room to maneuver.

The main trick we discovered is using the average slope over a column section to determine the number of trays in that section. This allowed us to solve simultaneously for the reflux ratio, distillate flowrate, number of trays, and feed location for a given recovery specification. This is not an essential technique because it is not always possible to get a given slope for a given specification, but it is very helpful for increasing the trays while approaching a product specification. In the algorithm, we first use the slope criteria to get us as far as possible towards the desired recoveries, which increases the number of trays. If it fails to go the whole way, we can generally fix the trays at this point and continue to go for the separation, solving for reflux and distillate flowrate.

Design Examples

We used the design algorithm discussed above to generate column designs for different ideal and nonideal systems. For an ideal system with a simplified cost function we used MINOS attached to ASCEND to minimize the cost with constraints on the separation of the key components. The feed was 3 components, 3 mole/s of each component. The relative volatilities were 1.5,1.2, and 1.0. A nominal solution requiring 95% purity of the key components had 30 trays in each column section and a reflux ratio of 9.8, with a diameter of 1.3 meters, height of 42 meters, and an investment cost of 157,000 dollars per year. The optimal solution had 19.5 trays top, 18.8 trays bottom, a reflux ratio of 14.5, with a diameter of 1.6 meters, height of 27 meters, and an investment cost of 130,000 dollars per year.

We used the algorithm shown in Figures 2 and 3 to design a column for an equimolar feed of propanol, isobutanol and butanol, with 3 mole/s of each component. The separation specification was 99% of the propanol in the distillate and 1% of the isobutanol in the distillate. Using the algorithm it took 8 minutes to create a model, solve it up to full thermodynamics and heat balance, and incrementally meet the purity specifications, representing 9 solutions of the full column model. All that was required was to enter the components, the feed composition, and the purity specifications through the interface we placed on top of ASCEND, as shown in Figure 1. The nominal solution had 59.25 trays, with the feed tray 60.2% down the column, a reflux ratio of 6.12, and an annualized cost of 178,000 dollars per year. We optimized this column using a simple gradient search implemented from an ASCEND script, varying the total number of trays and the feed tray location with the separation of the key components fixed. The "optimal" solution had 52.28 trays, with the feed tray 52.7% down the column, a reflux ratio of 6.021and a cost of 174,000 dollars per year. To test the optimization, we increased the utilities cost by a factor of 10. As expected, the number of trays increased (to 75.1). The feed tray location moved to 50.2% down the column, and the reflux ratio decreased to 5.71. When we decreased the utilities cost by a factor of 10, the number of trays decreased as expected to 49.66 while the reflux ratio increased to 6.15. Each optimization took about 10 minutes to perform, representing 30 column solutions. We performed these calculations on an HP700.

These design tests demonstrate that it is possible with this collocation model to solve for the number of trays and the feed tray location, while minimizing a cost function and holding a given specification. The time needed to execute the algorithm will decrease as we increase our understanding of the solution procedure.

Background on Flexible Distillation Design

Chemical plants use a wide variety of solvents which must be recovered for reuse. However, the feed to a solvent recovery plant will vary as the chemical plant demands change. Figure 5 shows a general picture of this type of problem. Any number of plants will send their waste mixed solvents to the recovery plant. Therefore, the feed to the recovery plant will change as the operation of the plants change. Azeotropic systems are particularly difficult because a change in the feed composition can move the system into a different distillation region. A simple, but expensive, solution to this problem is to require a constant feed composition to the recovery plant, using mixing to maintain the composition. A flexible solvent recovery plant, capable of separating a range of feeds with the same equipment, would be very useful to the chemical industry.

Very little work has been done on the specific problem of flexible distillation design. A significant amount of work addresses the general problem of process design flexibility. Halemane and Grossmann[l983] developed a formulation for determining design flexibility and ensuring design feasibility. For a convex constraint set, they showed that, by guaranteeing feasibility at critical vertices, the design will be feasible over the entire parameter space. They developed an approach to solving the two-stage programming formulation of the flexible design problem. Grossmann, Halemane, and Swaney[1983] presented an overview of optimization strategies for flexible chemical processes.

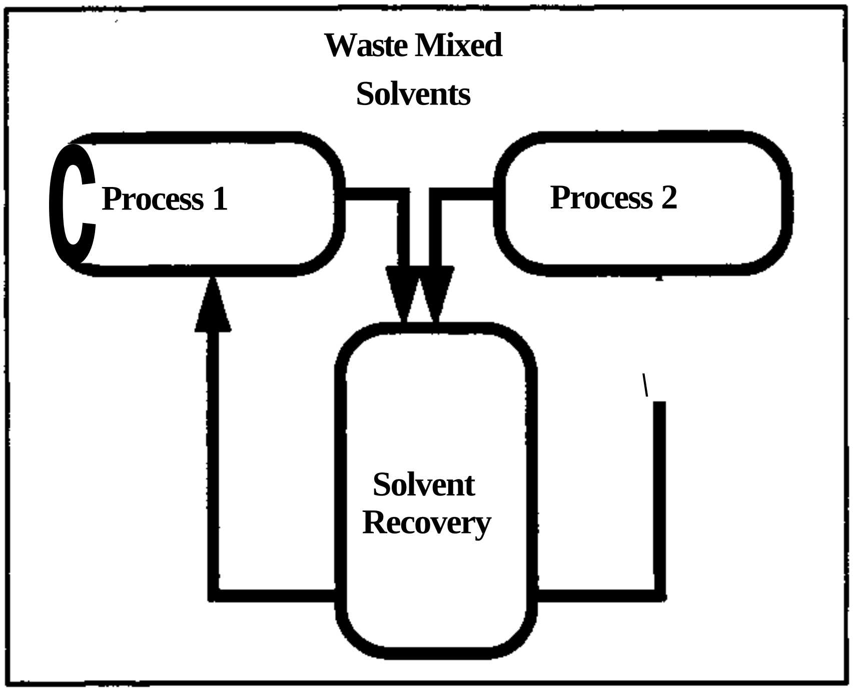

Figure 5. Solvent Recovery Plant Problem

Wagler and Douglas[1988] have addressed the specific problem of flexible distillation design. They used the concept of critical vertices also and simplified the problem further by combining vertices into "near critical points/' They determined these near critical points by having a working knowledge of characteristics of the system. Specifically for distillation, they showed that, for the five constraints on product purity, flooding limit, weeping limit, feasible heat exchanger operation, and feasible accumulator operation, two near critical points could replace the five constraints: However; their results showed that the near critical points did not guarantee feasibility for the heat exchangers. Also, they assumed that the system they investigated had constant relative volatilities, and they used the Fenske-Underwood-Gilliland shortcut method to design the columns.

Problem Statement:

Design a separation system that is feasible over a specified range of feeds, meeting product specifications, while minimizing the cost of the system.

We can characterize the variability of the feed in a number of ways. It could be gradually but continually changing, changing infrequently but abruptly, or changing frequently and abruptly. We chose to address the problem of an infrequently changing feed which changes abruptly to known points.

Figure 6 shows an example of this type of problem, using acetone, chloroform, and benzene. There is a maximum boiling azeotrope between acetone and chloroform, creating a distillation boundary between the azeotrope and benzene, as shown on the ternary diagram. The three points represent the three possible feed compositions. If designing a separation system individually for these feeds, we might come up with the flowsheets shown. The solvent recovery plant designed to handle any of these feeds could have only three columns, where the same columns are used in different configurations for different feeds. For any given feed, the columns may be operating close to flooding or weeping limits rather than at more conventional flows.

Single Column Problem

For this paper, we only consider the problem of designing a single flexible column. Flexible columns will be required by the type of problem shown above. Even if different feeds to the solvent recovery plant require different flowsheets as shown in Figure 6, a single column can be used in multiple flowsheets. Therefore, a subproblem to the overall synthesis problem could involve designing a column that is able to handle a set of different feeds, performing a specific separation task for each.

Assuming such a problem, we allow the reflux ratio, distillate flowrate, and *feed tray position* to be control variables. We must enforce flooding and weeping limits on the column's operation for each feed and design heat exchangers for the condenser and reboiler that are large enough to handle the operation of the column for all feeds. The objective is to design the column meeting these specifications while minimizing the cost of the column. The

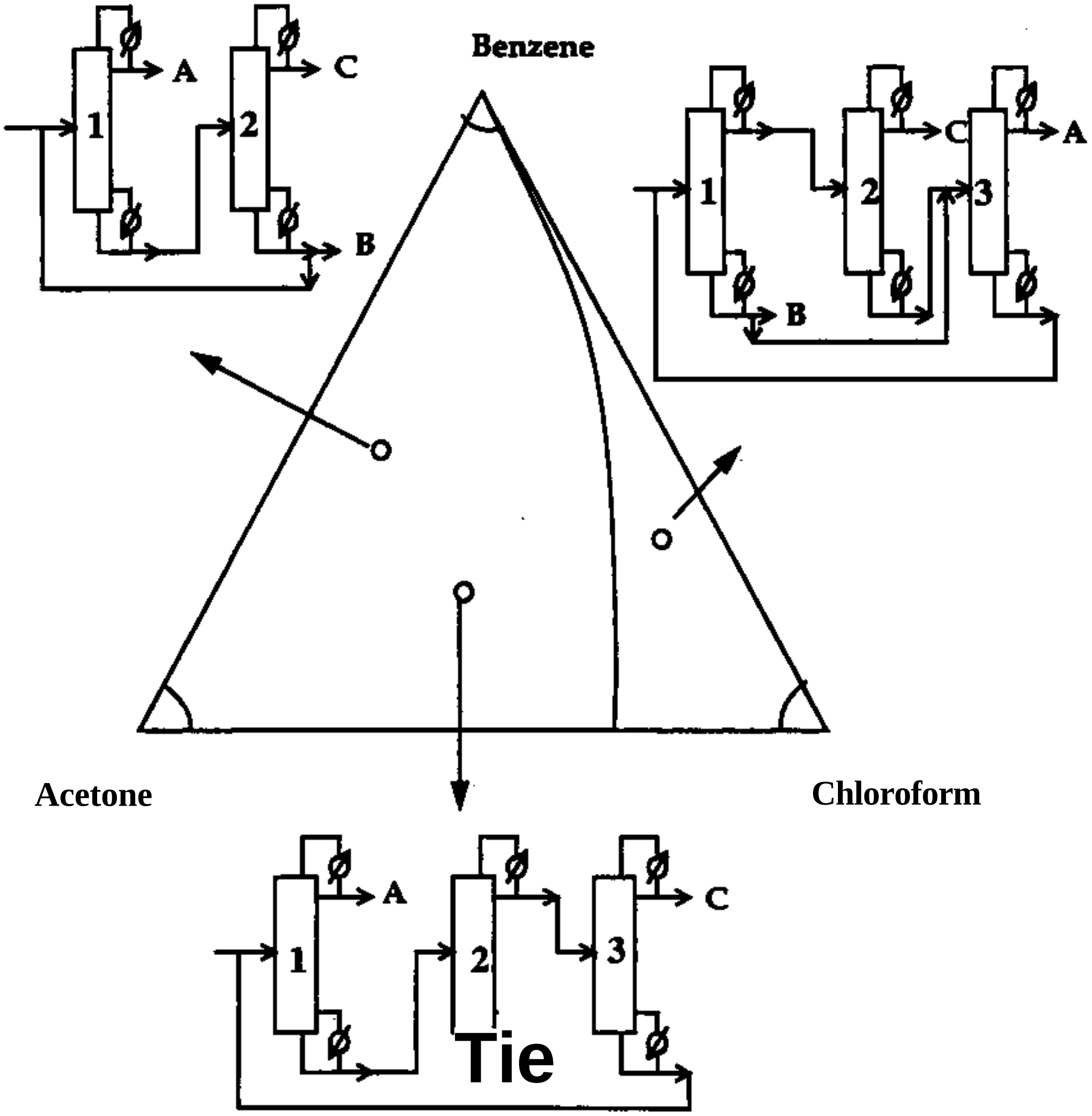

Figure 6. Solvent Recovery Plant Example

operating costs are averaged over the possible feeds weighting the cost by the estimated probability of each feed over time.

Flexible Design Algorithm

Figure 7 shows the algorithm used to design a single flexible column. The input information for this problem is the composition and flowrate of each feed

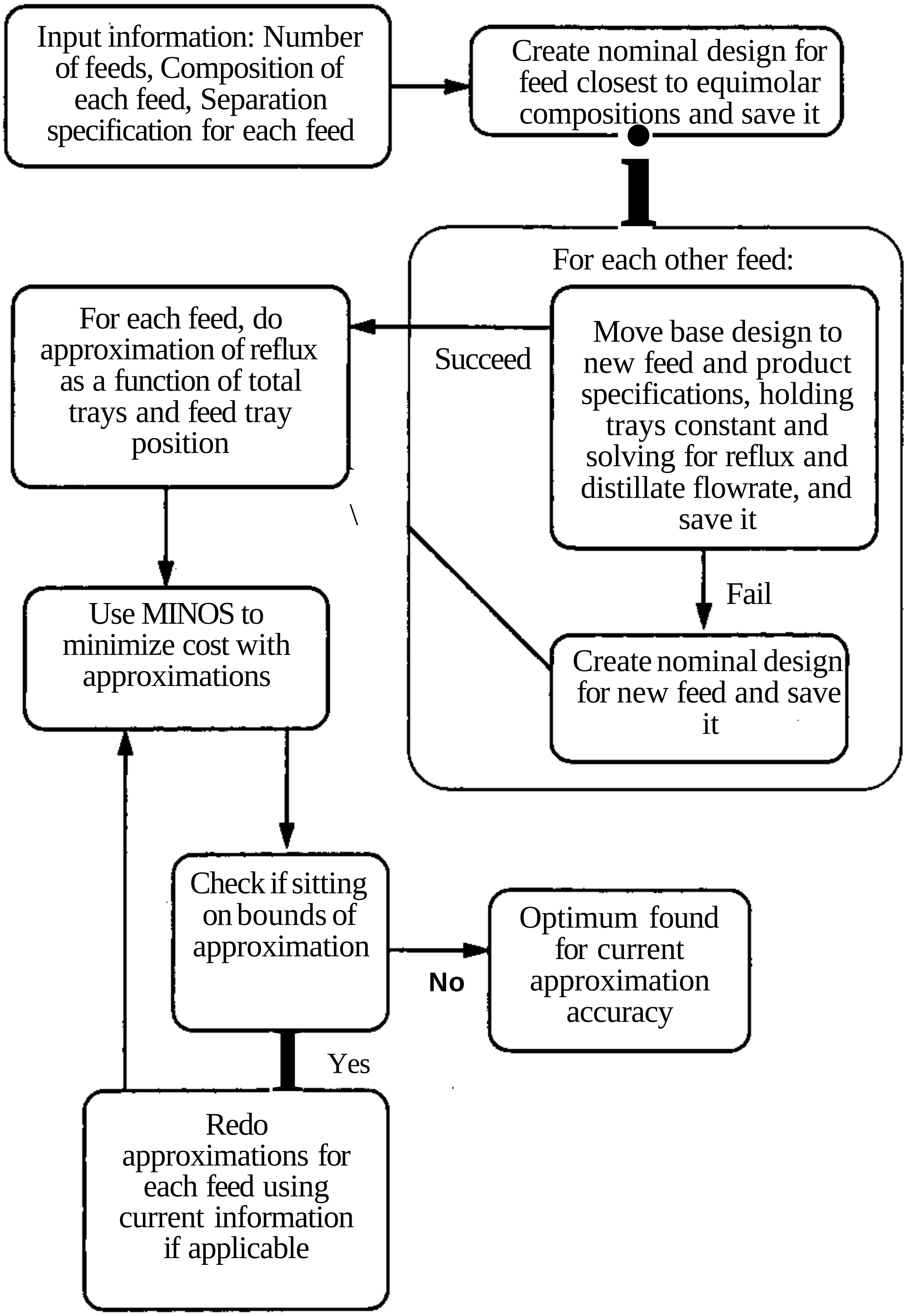
Input information: Number of feeds, Composition of each feed, Separation specification for each feed
Create nominal design for feed closest to equimolar compositions and save it
For each other feed:
Move base design to new feed and product specifications, holding trays constant and solving for reflux and distillate flowrate, and save it
Succeed
Fail
Create nominal design for new feed and save it
For each feed, do approximation of reflux as a function of total trays and feed tray position
Use MINOS to minimize cost with approximations
Check if sitting on bounds of approximation
No
Optimum found for current approximation accuracy
Yes
Redo approximations for each feed using current information if applicable

Figure 7. **Flexible Design Algorithm**

and the separation specification for each feed. With this information, we go through the column design algorithm presented earlier in this paper to create a nominal design for one of the feeds and save that solution. We choose the feed closest to the average composition of the feeds, which makes it more likely that the first nominal design will work for the other feeds.

For each other feed, we start with the nominal design and move its feed to the new feed, holding the number of trays constant. We create a nominal design for the new feed from scratch if we fail, getting a different number of trays for it. We then move the number of trays for each nominal design to the average number of trays.

We now have a nominal design for each possible feed. We save this information in a file for each feed, as well as saving it within the current ASCEND process. Within ASCEND we can make multiple saves of the values of existing models in RAM. Reading and writing these "virtual" saves takes a tenth of a second, while reading and writing to a file takes 30 seconds or more. We can read the values for these nominal designs into the one column model depending on the feed in which we are interested. Throughout the rest of the algorithm, we update the saved values whenever we change the column design.

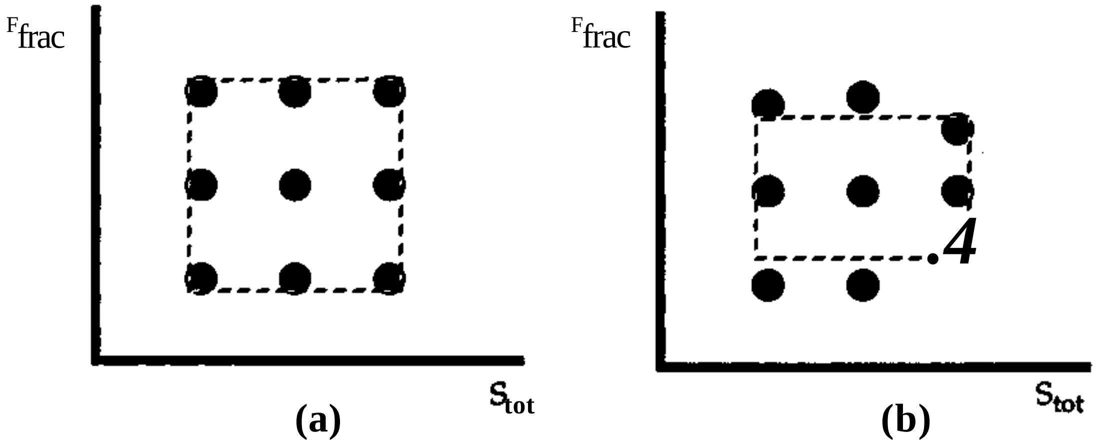

Figure 8. Data Point Placement

At this point we create an approximate model for each column to be used for cost minimization. We record the reflux ratio over a range of total trays and feed tray location. Figure 8a shows the normal placement of the data points. Generally we will move 10% away from the base point. From these nine data points we can fit reflux ratio as a quadratic function of the total number of stages, *Stot,* and the location of the feed tray, *Frac. Frac* is the fraction of the column above the feed tray. The approximation of the reflux ratio takes the following form.

$$^{R} = \overset{2}{\underset{fc=0}{X}} \hat{S}_k Stot^k \qquad \textbf{(1)}$$

$$h = \overset{2}{\underset{;\,=\,0}{X}} {}^{p}\!\cdot\! jk^{Frac}? \qquad <2)$$

Note that column diameter is not a concern here. For a given composition, the column trajectories will be the same, regardless of the total flowrate of the feed. The diameter can be determined once the reflux ratio, feed flowrate, and flooding factor are known.

We use the full model as a basis for the approximate model's cost calculation based on the number of trays, reflux ratio, and total feed flowrate. We set bounds for each approximation on *Stot* and *Frac* based on the range of the approximation. Figure 8b shows the points we might get if the column could not be moved fully 10% away from the base point. The dotted lines in both a and b show the bounds for that approximation.

Using these approximations, and requiring that the total number of trays and diameter is the same for each feed, we minimize the cost of the column. MINOS is attached to ASCEND for optimization problems. The optimization is formulated below:

tnin ColCost + RebCost + ConCost +WCost + SCost (3)

s.t. *ColCost ~~-~~ ColCost(H,D)* (4)

ConCosti -ConCostiiFeedfrRfrDisti), i =1..nfeeds (5)

RebCosti = RebCostfFeedjfRirDisti), i = L.nfeeds (6)

WCost_{t}- = *WCostiiFeedpRpDisti), i = h.nfeeds* (7)

SCosti = SCostiiFeed^R^Disti), i = L.nfeeds (8)

ConCost >_ ConCosty i = L.nfeeds (9)

RebCosf ≥ *RebCostj, i = h.nfeeds* (10)

$$wcost = \mathop{y}_{i \overset{\wedge *}{=} 1}^{nfeeds} \frac{P.(WCost.\,)_{x \quad J}}{nfeeds} \qquad (ID$$

$$SCost = \mathop{>}_{\overset{\bullet\wedge\bullet\mathrm{rf}}{i = 1}}^{nfeeds} \frac{p.({}_{S}Cost._{t})_{J}}{nfeeds} \qquad (12)$$

$$R_{t} \geq \mathop{\wedge}_{k=0}^{2} \$_{ik}Stot.\,{}^{k}_{t}\; i = L.nfeeds \qquad (13)$$

$$\hat{S}_{ik} = \mathop{\wedge}_{;=0}^{2} \hat{F}\wedge FraJ\text{-}\;{}^{\cdot}_{t},\; i = L.nfeeds \qquad (14)$$

H = 2.15Stot (15)

F_{t}- = *FiiFeedirRiPi), i m 1..nfeeds* (16)

$F_{\{} \leq$ *2.4, i = l..nfeeds* (17)

F,- ^ *0.8, i = l..nfeeds* (18)

Stot^StotU (19)

Sfo* £ SfofL (20)

$$Frac_{\{} \leq FracUj, i = L.nfeeds \quad (21)$$

$$Frac_{\{} \geq FracL_{\{}, i = L.ttfeeds \quad (22)$$

The cost functions in equations 4-8 are Guthrie cost calculations and are functions of the total feed flowrate, the reflux ratio, distillate flowrate, and the flooding factor [Douglas, 88]. The flooding factor, F, is a dimensionless quantity, representing how close the column is to the flooding limit. For tray spacing of 2 feet, a flooding factor of 2.51 represents flooding [Douglas, 88]. Normally, we would design at 60% of flooding, but, for the flexible design problem, the column may need to run close to flooding for some feeds and close to weeping for others.

We are designing the condenser and the reboiler in equations 9 and 10 to be large enough for the largest demand. Since the cost of the exchangers is based on area, an exchanger with the largest area will have the highest cost and we assume can be operated to handle the other feeds. The utility costs in equations 11 and 12 are a weighted average over all the feeds, where P; is the probability of feed *i* occurring over the time period of interest. Equations 13 and 14 are the reflux approximations for each feed. Note that equation 13 requires the reflux ratio to be greater than or equal to the reflux required for the specified separation. This allows the column to over-separate when the weeping limit is encountered. Equation 15 relates the height of the column to the number of trays. F_{Z} is the flooding factor for column *i* and is a function of the total feed flowrate, the reflux ratio, and the diameter. We determine the bounds used in equations 19-22 based on the data points used for the approximation of each column.

If the result from MINOS is on the bounds for the number of stages or the feed tray location, *Slot* or *Frac,* we perform the approximation again about the new point. If the point is in a corner, we need to generate 5 new points. If it is on an edge, we only generate 3 new points. Figure 9 shows two steps from an initial approximation. The first hits only the bound on *Stot,* so we use 6 existing

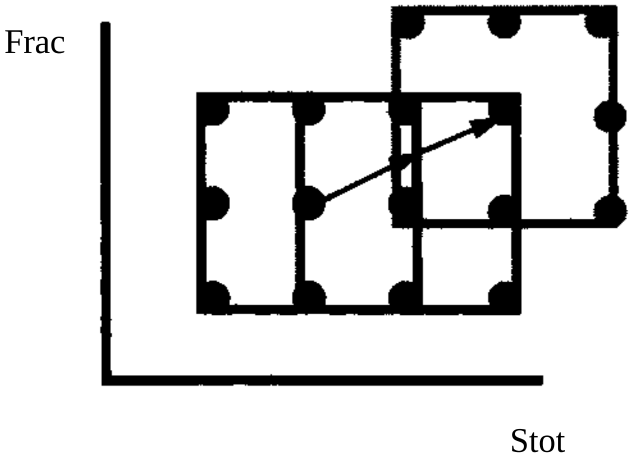

Figure 9. Reapproximation example

data points and create 3 more for the new approximation. The second optimization hits the upper bound on both *Stot* and *Frac,* so we can use 4 existing data points and must create 5. We keep looping through reapproximations and optimizations until the optimum is not on the bound for trays or feed tray location. This is a local minimum of cost based on the current approximation range. At this point, we could reduce the approximation range and re-optimize to get a more accurate solution.

Flexible Design Examples

We used the algorithm described above to design a flexible column for a propanol, isobutanol, butanol system. There were four possible feeds, each equally likely, and we required that 99% of the propanol and 1% of the isobutanol would be in the distillate. The component flowrates of each feed are listed in Table 1. Table 1 also shows the reflux ratios for the initial solution of 86 trays. After optimizing, the solution had 64.15 trays, and a cost of $210,000/year. The reflux ratios, flooding factors, and feed locations for the final solution are also Table 1.

Table 1. Flexible Design Example 1

	Feedl	Feed 2	Feed 3	Feed 4
Propanol feed	3 mole/s	1 mole/s	7 mole/s	3 mole/s
Isobutanol feed	3 mole/s	5 mole/s	3 mole/s	9 mole/s
Butanol Feed	3 mole/s	5 mole/s	3 mole/s	3 mole/s
Reflux ratio (86 trays)	5.73	17.61	3.69	5.73
Reflux ratio	6.18	20.91	3.84	5.71
Flooding factor	1.52	1.67	2.35	1.43
Feed location	0.51	0.47	0.64	0.47

Our second example is an acetone, chloroform, benzene system. We used approximately the three feeds shown on Figure 6. We desire a column that can perform the first separation task in each flowsheet. The two feeds to the left of the distillation boundary are separated to 99% pure acetone and the distillation boundary. The feed to the right of the boundary is separated to 99% pure benzene and a mixture of acetone and chloroform. We uses the alternative in Figure 3 for azeotropic systems where we first meet the purity specification of one of the product streams at a very low flowrate and then increase the flowrate of the product. For example, we want the first two feeds to have 99% pure acetone from the top, but the recovery of acetone will not be 99%. We set the distillate flowrate to a small number and met the purity specification on acetone while solving for the reflux ratio and the number of trays. Then we incrementally increased the distillate flowrate as far as it would go while maintaining the purity on acetone.

Table 2 shows the information for this problem. The initial solution had 24.2 trays. The final solution had 27.7 trays, and a cost of $150,000/year. This example demonstrates how a column can be designed for completely different separation tasks.

Table 2. Flexible Design Example 2

	Feedl	Feed 2	Feed 3
Acetone feed	3 mole/s	3 mole/s	1 mole/s
Chloroform feed	3 mole/s	1 mole/s	6 mole/s
Benzene feed	2 mole/s	3 mole/s	3 mole/s
Reflux ratio (24 trays)	12.4	6.8	3.6
Reflux ratio	11.5	4.8	2.5
Flooding factor	2.06	1.13	2.4
Feed location	0.78	0.85	0.5

Conclusions

This collocation method is an excellent tool for distillation design. By having the number of trays as a continuous variable, we are able to develop new concepts and techniques for distillation design and optimize the size of the distillation column. While developing the algorithms, we discovered some techniques that significantly improve the performance and enabled us to automate the design process. By modeling in the ASCEND system, we were able to learn how to solve these problems and create the tools for anyone to use our algorithms.

We demonstrated how a single column can be designed to deal with different separation tasks, providing the building block for more complex flexible design problems. This collocation method and these basic design algorithms should enable development of an algorithm for design of solvent recovery plants. However, we should note that this tool only finds the best design assuming the column must perform all the separation tasks. It may be more economical to have extra columns or to store mixtures for a time. The overall flexible design problem needs to address these issues.

Acknowledgment

Support from Eastman Chemicals and DOE contract number DE FG02-85ER13396 provide support for this research. Facilities support is from NSF grant number EEC-8942146 which supports the EDRC, an NSF funded Engineering Research Center.

Nomenclature

R	Reflux ratio
Stot	Total number of stages in a column
$\hat{S}$	Coefficient of reflux ratio fit.
$\hat{F}$	Coefficient of reflux ratio fit.
Frac	Fractional location of the feed tray down the column.
ColCost	Material cost of column.
RebCost	Material cost of reboiler.
ConCost	Material cost of condenser.
Wcost	Operating cost of cooling water.
Scost	Operating cost of steam.
Feed	Feed stream to column.
Dist	Distillate product from column
nfeeds	Number of possible feed streams.
H	Height of column.
F	Flooding factor.
StotU	Upper bound on number of stages.
StotL	Lower bound on number of stages.
Fracll	Upper bound on fractional location of feed tray for column.
FracL	Lower bound on fractional location of feed tray.

References

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

Grossmann, I. E, K. P. Halemane, and R.E. Swaney. 1983 "Optimization Strategies for Flexible Chemical Processes." *Contp. Chem Eng.* 7,439-462

Halemane, K. P. and I. E. Grossmann. 1983. "Optimal PRocess Design Under Uncertainty." *AIChE J.* 29,425-433

Huss, R. S. and A. W. Westerberg. 1995. "Collocation Methods for Distillation Design I: Model Description and Testing" to be submitted.

Huss, R. S. and A. W. Westerberg. 1995. "Collocation Methods for Distillation Design II: Minimum Reflux" to be submitted.

Piela, P., McKelvey, R., and Westerberg, A. 1993, "An Introduction to the ASCEND Modeling System: Its Language and Interactive Environment," /. *Management Information Systems,* 9,91-121

Wagler, R. M. and P. L Douglas. 1988 "A Method for the Design of FLexible Distillation Sequences." *Canadian f. of Chem. Eng.* 66,565-581