K = adsorption equilibrium constant, cm^3/g
L_1 = height of slurry, cm; region L_1 to L_2 is gas collecting space above slurry
m_n = moment component, defined by Equation (10)
m_s = mass of carbon per unit volume of bubble and solid free liquid, g/cm^3
n = amount of adsorbed nitric oxide, mole/g
Q = gas feed rate at 25°C and 1 atm, cm^3/s
r = radial coordinate in a spherical particle, measured from its center, cm
R_B = average radius of gas bubbles, cm
R = radius of spherical particle, cm
s = Laplace variable
S_{BET} = BET surface area, m^2/g
t = time, s
v_B = vertical bubble velocity, cm/s
V_B = bubble volume per unit volume of bubble and solid-free liquid
V_L = total volume of liquid in the vessel, cm^3
z = slurry height, measured from bottom of the vessel, cm

Greek Letters

α = bubble-to-liquid rate parameter defined by Equation (16), cm^{-1}
β = particle porosity
$\delta(0)$ = input pulse, Dirac delta function with an infinite value at $t = 0$
μ'_{1,L_1} = first absolute moment in the slurry, or for the liquid when no particles are present, s
$\mu'_{1,d.v.}$ = first absolute moment in the dead volumes, s
$\mu'_{1,D}$ = first absolute moment evaluated at the detector, s
μ_{2,L_1} = second central moment for the slurry; $\mu_{2,d.v.}$ and $\mu_{2,D}$ are second moments for the dead volumes and as measured at the detector, respectively, s^2
ρ_p = particle density, g/cm^3
ρ_t = solid-phase density, g/cm^3
γ = tortuosity factor

Subscripts

L_1, L_2 = value at $z = L_1$ and L_2 respectively
b, δ = responses to step and pulse (δ) inputs
o = inlet value
PF = gas bubbles in plug flow through slurry
ST = gas bubbles completely mixed in slurry

LITERATURE CITED

Calderbank, P. H., "Mixing," V. W. Uhl and J. B. Gray, ed., Vol. II, p. 65, Academic Press, New York (1967).

Furusawa, T., and J. M. Smith, "Fluid-Particle and Intraparticle Mass Transfer Rate in Slurries," *Ind. Eng. Chem. Fundamentals*, **12**, 197 (1973*a*).

———, "Mass Transfer Rate in Slurries by Chromatography," *ibid*, 360 (1973*b*).

Furusawa, T., and M. Suzuki, "Moment Analysis of Concentration Decay in a Batch Adsorption Vessel," *J. Chem. Eng. Japan*, **8**, 119 (1975).

Kenney, C. N., and W. Sedriks, "Effectiveness Factors in a Three-Phase Slurry Reactor," *Chem. Eng. Sci.*, **27**, 2029 (1972).

Juvekar, V. A., and M. M. Sharma, "Adsorption of CO_2 in a Suspension of Lime," *ibid.*, **28**, 825 (1973).

Komiyama, H., and J. M. Smith, "Sulfur Dioxide Oxidation in Slurries of Activated Carbon. Part I. Kinetics," *AIChE J.*, **21**, 664 (1975); "Part II. Mass Transfer Studies," *ibid.*, 670.

———, "Surface Diffusion in Liquid-Filled Pores," *ibid.*, **20**, 1110 (1974).

Linke, W. F., "Solubilities," American Chemical Society, Washington, D.C. (1965).

Misic, D. M., and J. M. Smith, "Adsorption of Benzene in Carbon Slurries," *Ind. Eng. Chem. Fundamentals*, **10**, 380 (1971).

Prasher, B. D., and G. B. Wills, "Mass Transfer in an Agitated Vessel," *Ind. Eng. Chem. Process Design Develop.*, **12**, 351 (1973).

Reid, R. G., and T. K. Sherwood, "The Properties of Gases and Liquid," 2 ed., McGraw-Hill, New York (1966).

Satterfield, C. N., "Mass Transfer in Heterogeneous Catalysis," pp. 38-39, 119-120, M.I.T. Press, Cambridge, Mass. (1970).

Sherwood, T. K., and E. J. Farkas, "Studies on the Slurry Reactor," *Chem. Eng. Sci.*, **21**, 573 (1966).

Siems, W., and W. Weiss, "Mixing of Liquid by Means of Gas Bubbles in Narrow Columns," *Chem. Eng. Tech.*, **29**, 727 (1957).

Solbakken, A., and L. H. Reyerson, "Sorption and Magnetic Susceptibility Studies on Nitric Oxide—Alumina Gel Systems at Several Temperatures," *J. Phys. Chem.*, **64**, 1903 (1960).

Valentin, F. H. H., "Absorption in Gas-Liquid Dispersions," E. and F. N. Spon, Ltd., London, England (1967).

Manuscript received February 19, 1976; revision received April 16, and accepted April 20, 1976.

Separation Sequence Synthesis by a Predictor Based Ordered Search

ALEJANDRO GOMEZ M.

and

J. D. SEADER

Department of Chemical Engineering
The University of Utah
Salt Lake City, Utah 84112

A predictor based ordered search procedure is used to scan the graph representing all possible separation sequences for a given multicomponent separation process. The algorithmic procedure is expedited by utilizing an heuristic cost function to obtain lower bound estimates of the cost of separators not yet designed. Considerable reduction in search space occurs. Optimal and near optimal sequences are readily generated.

SCOPE

Many chemical processes involve separation sequences, wherein multicomponent feed mixtures are separated into products (including recycle or intermediate streams) by two or more separators. As shown by Thompson and King (1972*a*), the number of possible processing schemes or sequences (arrangements of separators) increases exponentially with the number of products and with the number of different types of separators being considered.

Correspondence concerning this paper should be addressed to J. D. Seader.

If only ordinary distillation is to be employed, the number of possible sequences increases from two to five as the number of products increases from three to four. Rodrigo and Seader (1975) show an example involving one feed to be separated into three products by ordinary distillation and/or extractive distillation with a single solvent, wherein nine different sequences are possible.

Because computing time and cost for the analysis of multicomponent separators are not negligible, it is desirable to have procedures for determining the optimal and/or near optimal sequences without examining all possible sequences. Several such procedures have been developed. Among the heuristic techniques, the cheapest-first heuristic of Thompson and King (1972*b*) can produce a near optimal (and sometimes optimal) sequence with very minimal computing requirements. The method is particularly useful in preliminary design problems involving very large search spaces.

The algorithmic methods of Hendry and Hughes (1972) and Westerberg and Stephanopoulos (1975) optimize with respect to structure and a limited number of operating conditions. Although these two methods find the optimal solution without examining every sequence, the former method based on dynamic programming does require the analysis of all unique separation subproblems.

The ordered branch search procedure of Rodrigo and Seader (1975) optimizes only the arrangement of the sequence but does so algorithmically. Heuristics are utilized to set values of separator operating conditions. Like the branch and bound method of Westerberg and Stephanopoulos (1975), the ordered branch search procedure may not require the analysis of all separation subproblems, particularly when a very wide distribution of sequence costs exists. All three algorithmic procedures can synthesize near optimal as well as optimal sequences.

Despite the substantial progress made toward a rapid and efficient technique for synthesizing separation trains, the following desirable features are lacking in previous algorithmic methods: reduction of search space for separation problems involving a relatively narrow range of sequence costs, elimination of the need to generate a list of all possible separation subproblems (this list can become extensive for large problems), and ease of application to process simulation programs, which are widely used in practice and are described by Motard, Shacham, and Rosen (1975). In this paper, a new technique for the synthesis of separation sequencies is developed that has these features.

Among the intelligent structured search techniques (dynamic programming, integer programming, etc.) are the so-called blind search procedures. These can be classified into two types, breadth first and depth first as discussed by Nilsson (1971). In both types, nodes representing feeds or intermediate streams are expanded in terms of alternative separation subproblems to generate new stream nodes until product goal nodes are generated. A depth-first algorithm is one where the most recently generated nodes are expanded first. The ordered branch search is an example of this type. Goal nodes are reached early in the search. An example of a breadth-first algorithm (an algorithm that expands nodes in the order in which they are generated) is the so-called uniform cost method of Nilsson. When blind algorithms are enlightened by a heuristic, they are called heuristic search algorithms. In general, the technique presented here is a modified uniform cost method, which uses a heuristic cost evaluation function of the type described by Nilsson (1971), to order and reduce the search.

Unlike previous methods, which are essentially blind search techniques, the method reported here continuously estimates and uses an absolute minimum cost for that portion of the search space not yet designed. In this manner, lower (rather than upper) bounds to the total sequence cost are generated to direct development of the optimal sequence. Sequences are generated starting from the process feed stream and moving toward the product streams. The lower bounds are computed as the sum of the actual cost of the partially developed sequence and a prediction of the minimum cost for that part of the sequence remaining to be developed. This prediction employs the following separation cost heuristic. A separation is least expensive when conducted in the absence of nonkey components. Although this heuristic is well-known and is listed by King (1971) and by Rudd, Powers, and Siirola (1973), it has not previously been applied in the manner reported here. Four examples are presented to illustrate the application of the new search procedure, and comparisons are made to some of the other synthesis techniques. The synthesis problem addressed here involves the same component material balance simplification utilized by Rodrigo and Seader (1975). Also, only the system structure is optimized, and a mass separating agent need not be recovered in the separator following the one to which it is added.

CONCLUSIONS AND SIGNIFICANCE

The predictor based ordered search procedure for finding optimal separation sequences is found to be efficient with respect to computer memory requirements and computation time. By continually predicting lower bounds on the complete costs of sequences, the optimal sequence is rapidly generated by analyzing only a fraction of the separators in the search space. A minimum of additional computations is required to generate a specified number of near optimal sequences or the other sequences that are within a specified factor of the cost of the optimal sequence.

The heuristic used to estimate the costs of complete sequences appears to have reasonably general applicability. However, it may fail when the presence of nonkey components has a significant effect on the relative separability between the two key components. Nevertheless, in the four examples examined here, the optimal sequences were generated. The new procedure is considerably more efficient than the dynamic programming-list processing method. However, the latter method can also optimize operating conditions. The new procedure is also more efficient than the ordered branch search method, particularly where sequence cost distributions are relatively narrow.

The new procedure can be implemented with advanced process simulation programs, such as the FLOWTRAN program described by Seader, Seider, and Pauls (1974), by using FORTRAN insert statements to direct the procedure and special modules for calculating and accumulating costs.

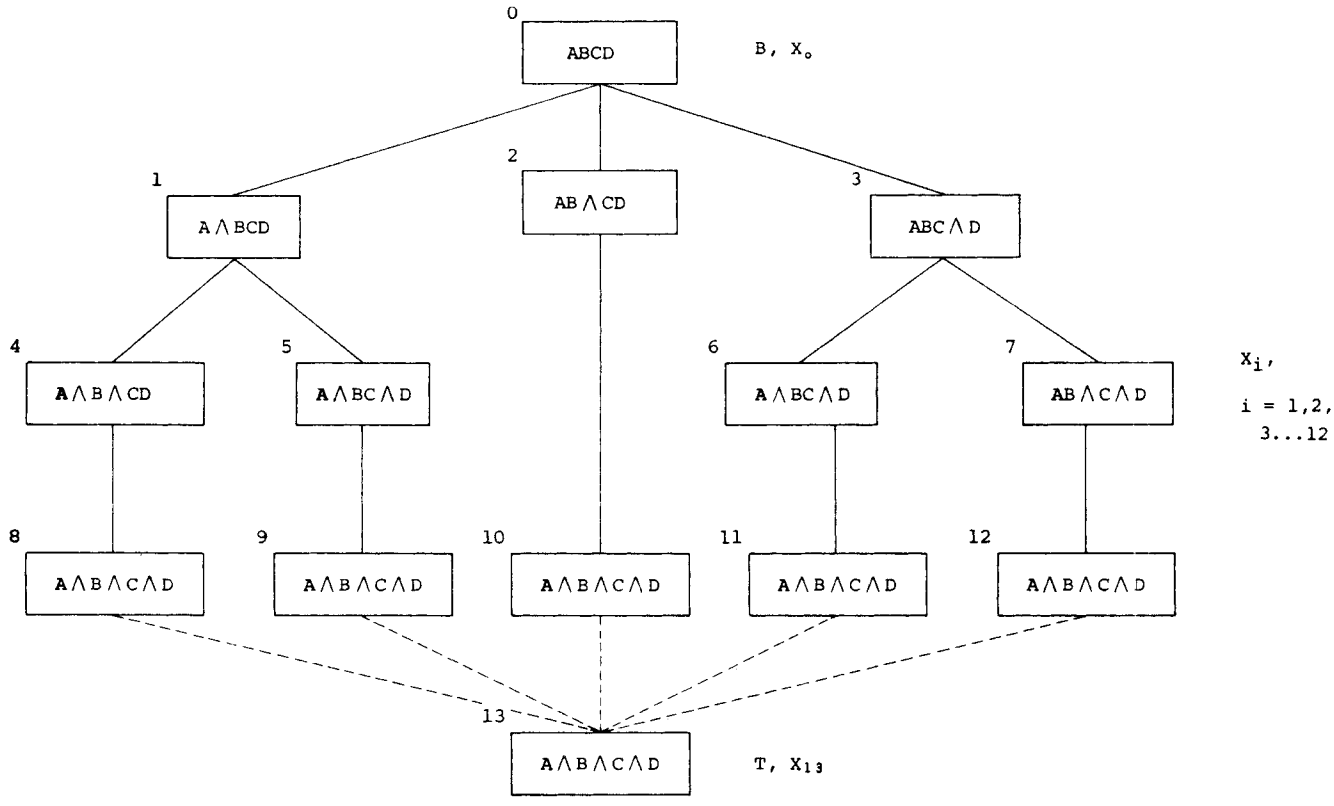


Fig. 1. Example of sequence graph for use of one type of separator.

PROBLEM STATEMENT

Consider the following hypothetical example. A mixture containing species A, B, C, and D (listed in the order of decreasing value of relative volatility) is to be separated by ordinary distillation into four essentially pure product streams. The search space of all possible separation subproblems and sequences may be represented by an and/or directed graph of the type illustrated by Rodrigo and Seader (1975). Alternatively, as shown in Figure 1, a sequence graph G may be constructed that shows the development of all sequences in a more compact fashion. In Figure 1, B is the beginning or start node, which represents the feed stream to the separation process. Nodes X_i represent steps in the development of complete sequences of separation operations leading to the terminal node T, which contains all products. Solid arcs represent separation operators $\Gamma_t^{m/n}(i, j)$ for expanding any node X_i to successor nodes X_j for a given separator type t and separation point m/n, where m and n are the light key and heavy key, respectively. Corresponding to each solid arc connecting nodes X_i and X_j is a cost $C(i, j)$. Dashed arcs connect the last node in a completed sequence to the identical goal node of desired product streams. Thus, no Γ or C is associated with a dashed arc. Also associated with each node X_i is the actual cost $g(B, i)$ of the partially completed sequence beginning from B and ending in that node, and the evaluation function equal to the estimated cost $g^*_i(B, T)$ of the complete sequence passing through that node as determined from

$$g^*_i(B, T) = g(B, i) + C(i, j) + C^*(j, T) \qquad (1)$$

where i refers to the predecessor of j, and $C^*(j, T)$ is a heuristic cost function equal to the estimated cost for completing a sequence that starts with intermediate node X_j.

It is assumed that a successor node X_j is reachable from a vertex node X_i if an arc exists from X_i to X_j. Such accessibility can be represented for each type of separator t by a reachability matrix. The connection array consists of entries $M(i, j)$, where the entry is 1 if the arc exists and 0 if it does not. In the synthesis method described here, it is not necessary to determine the complete reachability matrix for each type of separator prior to developing the sequence graph G. When the terminal

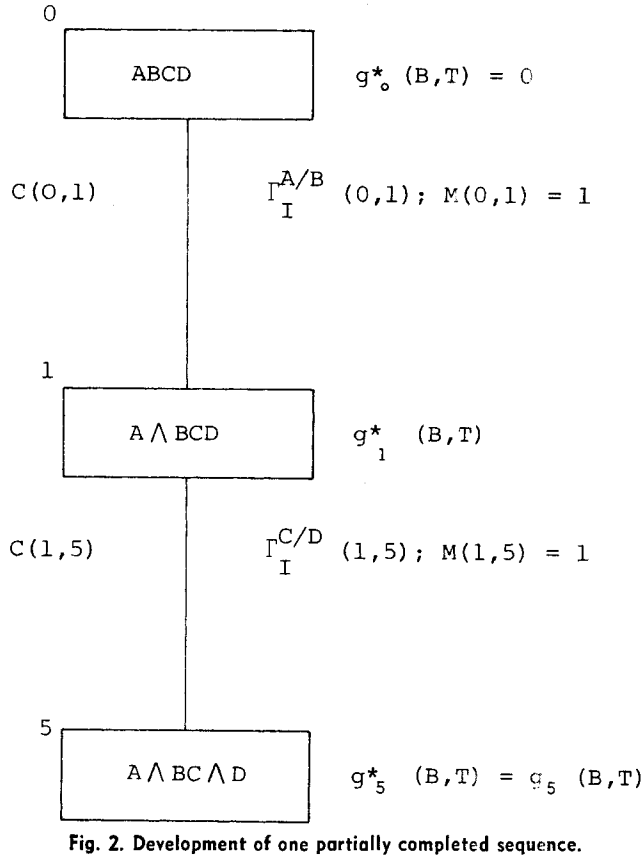
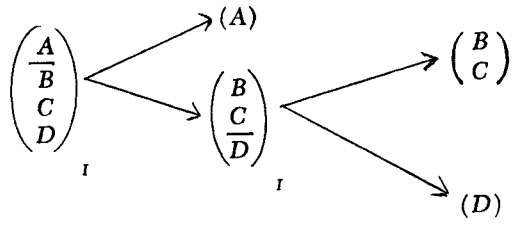


Fig. 2. Development of one partially completed sequence.

node is reached, elements of the matrix M act as pointers to back generate the complete sequence. Partially completed sequences can also be generated in this manner. For example, a partially completed sequence terminating in node 5 is readily generated to be



by the values of M shown in Figure 2, where I indicates the use of ordinary distillation.

The separation problem may now be formally stated as follows. Given B, T, and a means of determining feasible Γ and corresponding values of C, find the optimal and near optimal sequences contained in the set S by developing as little of the search space or sequence graph $G = [N, A, \phi]$ as necessary, where N is the complete set of nodes, A is the complete set of arcs, and the complete set of separation operators is

$$\phi = \{\Gamma_t^{m/n}(i, j)\} \qquad (2)$$

$\forall$ selected t, and

$\forall$ feasible pairs m/n

The effectiveness of the search procedure may be measured by search factors F_s, F_{sp}, and F_{usp}, as discussed by Rodrigo and Seader (1975).

MODIFIED UNIFORM COST ALGORITHM

The uniform cost algorithm is a blind search procedure, wherein a breadth-first search is made beginning from the start node. As nodes are expanded, they are ordered according to $g(B, i)$, the cost of the partially completed sequence. The node X_i with the smallest value of $g(B, i)$ is expanded next. Because values of $g(B, i)$ are continually increasing as nodes are expanded, the optimal sequence having the smallest value of $g(B, i)$ can be found only by searching along the path of every sequence until the value of $g(B, i)$ for that partially completed sequence exceeds an upper bound value. The upper bound value is initially the value of $g(B, i)$ for the first completed sequence or for a subsequently completed more optimal sequence. Although it proceeds in the opposite direction, the dynamic programming method is essentially of the uniform cost type. Most other methods are breadth-first techniques.

The uniform cost method can be modified by ordering nodes according to g^*, as given by Equation (1), provided that a method is available for predicting the minimum value of C accompanying the remaining development of a sequence. Assume that such a method exists, as will be described in the next section. To find the optimal sequence, the path of a sequence need only be searched

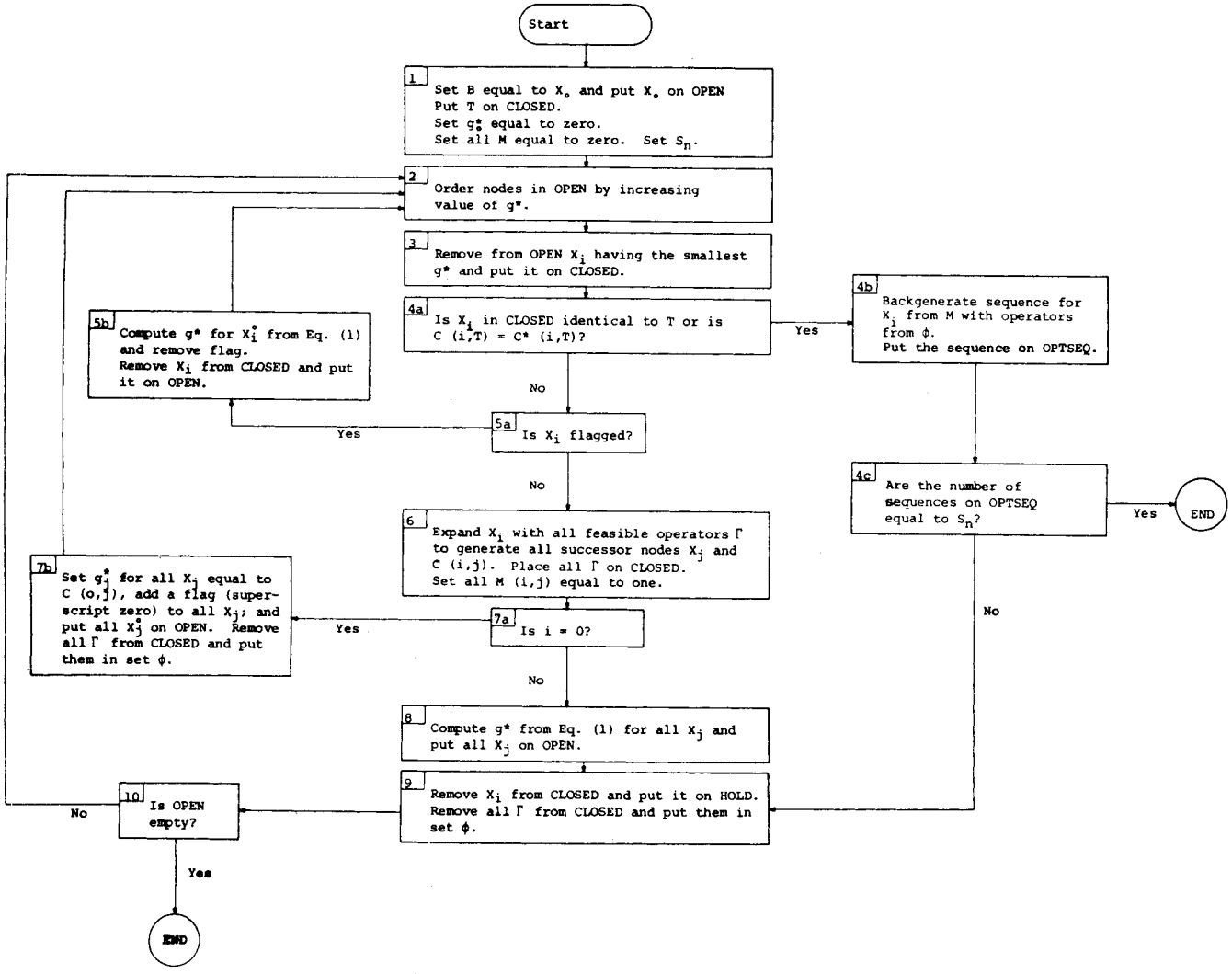


Fig. 3. Algorithm for predictor based ordered search.

if that path currently has the lowest value of g^*. As a sequence is developed, its value of g^* monotonically increases. Thus, in Figure 2, $g^*_5(B, T) > g^*_1(B, T) > g^*_0(S, T)$. The smallest value of g^* for all partially completed sequences is the lower bound. Node expansions are dictated by this lower bound (which may be increasing continually) and are not continued until some upper bound (that may be continually decreasing) is exceeded. With this predictor based ordered search, the total cost of any completed sequence will always equal or exceed the value of g^* that represents the current lower bound.

The algorithm for this procedure is outlined in Figure 3 and is as follows:

Step 1. Set the process feed or start node B equal to node X_o and put X_o on a list called OPEN. Put the process products or terminal node T on a list called CLOSED. Set the evaluation function g^*_o for X_o equal temporarily to zero. Zero out the reachability matrix, M. Set S_n equal to the desired number of near optimal sequences that are to be generated.

Step 2. Order the nodes X_i on OPEN according to increasing values of g^*. However, when coming from Step 1, only $B = X_o$ will be on OPEN.

Step 3. Remove from OPEN node X_i having the smallest value of g^* and put it on a list called CLOSED.

Step 4. a. Compare the node just put on CLOSED with the goal node T. If they are not identical or if the remaining separations have not already been designed and costed in computing $C^*(i, T)$, go to Step 5a. b. Otherwise, back generate the sequence for X_i from the pointers of matrix M using the operators in ϕ and put the sequence on the list OPTSEQ, which consists of the optimal and near optimal sequences. c. If the number of sequences in OPTSEQ is the desired number S_n, terminate the run. Otherwise, go to Step 9.

Step 5. a. Check to see if X_i is flagged. If not, transfer to Step 6. b. Otherwise, recompute g^* for X_i^o by adding $C^*(i, T)$ to the temporary value of $g^*_i = C(O, i)$ previously computed in Step 7b. Remove the flag from X_i^o to give X_i. Remove X_i from CLOSED and put it on OPEN. Go to Step 2.

Step 6. Expand node X_i on CLOSED to generate all successor nodes X_j. This is done by determining and utilizing all feasible operators Γ. Change corresponding $M(i, j)$ to one. Place all operators Γ on CLOSED. If a mass separating agent is added in the operation, the agent is included in the list of species in X_j, and a flow rate is associated with the agent. If a mass separating agent is recovered, its flow rate is subtracted from the value previously associated with the mass separating agent, and the agent is deleted from the list of species associated with X_j. As in the method of Hendry and Hughes (1972), overall process recoveries are assumed for the key components of each separation operation in order to design the separator. However, after the design calculation for that particular separator is completed, each species is reallocated to the separator products on the basis of a perfect separation between the key components. While this simplification is not essential, as discussed by Rodrigo and Seader (1975), it is desirable, particularly when mass separating agents are involved and when multiplicate separation subproblems can be identified as a result of the simplification.

Step 7. a. If this is not the first pass through Step 2, i will not be equal to zero, and a transfer is made to Step 8. b. Otherwise, with $i = 0$, compute all $C(o, j)$ and set all g^* equal temporarily to corresponding values of $C(o, j)$. Put all X_j on OPEN. Transfer to Step 2. Thus, on the initial expansion of the start node, only the costs of the first separators are computed. Estimates of complete sequence costs are not made in Step 7b. Instead, all X_j are flagged by adding a zero superscript to give X^0. Remove operators $\Gamma_t^{m/n}(i, j)$ from CLOSED and put them in a set ϕ.†

Step 8. Compute the evaluation function $g^*_j(B, T)$ for each successor node X_j from Equation (1), where $g(B, i)$ is the cost associated with the parent node X_i, $C(i, j)$ is the cost of the particular expansion operation $\Gamma(i, j)$, and $C^*(j, T)$ is the heuristic function or lower bound estimate of the remaining cost of the sequence calculated in the manner described in the next section. Thus, $g^*_j(B, T)$ is a lower bound estimate of the cost of a complete sequence passing through node X_j. Put all values of X_j on OPEN.

Step 9. Remove parent node X_i from CLOSED and put it on HOLD for future reference. Associated with X_i are costs $g(B, i)$ and $g^*_i(B, T)$. Remove operators $\Gamma_t^{m/n}(i, j)$ from CLOSED and put them in set ϕ. Associated with these operators are arc costs $C(i, j)$. The operators serve as pointers to generate paths of optimal and near optimal sequences. Set ϕ can also be checked in Step 6 for multiplicate operators so to avoid redesigning the same separator.

Step 10. Check OPEN to determine if it is empty. If so, terminate the run. This can occur if all sequences are desired ($S_n \geq$ total number of possible sequences. If OPEN is not empty, return to Step 2.

An algorithm is called admissible if it guarantees to find an optimal sequence from B to T for any graph G. In the following section, the heuristic cost function $C^*(j, T)$ is defined, and the admissibility of the algorithm just presented is proved in the manner of Hart, Nilsson, and Raphael (1968).

HEURISTIC COST FUNCTION FOR SEPARATORS

Uniform cost algorithms can be inefficient for searching graphs to find optimal and near optimal sequences, when the cost distribution of the sequences is narrow. This is because only costs of previous node expansions are accumulated and compared to an upper bound. Suppose, instead, that estimated (lower limit) costs g^* for complete sequences passing through any node X_j can readily be estimated by looking ahead to the goal node by means of C^* in Equation (1). If these estimates are continually made as successor nodes X_j are generated, and if all nodes on OPEN are ordered according to increasing estimated cost for the complete sequence, then expansion of the node with the lowest value of g^* will eventually lead to the optimal sequence:

Lemma: Let the heuristic cost function be

$$C^*(j, T) = \Sigma C(X, T) \tag{3}$$

where X represents an immediate predecessor of goal node T (inverse operation); that is, $X = [\Gamma_*^{m/n}]^{-1}(T)$ $\forall$ (m/n) of search space (X_j, T), and Γ_* represents the least expensive way to perform a final expansion on a graph. If $C^*(j, T) \leq C(j, T)$, $\forall$ X_j; then, from Equation (1) for an optimal sequence from start node B to goal node T, a node X_j exists on the list OPEN such that

$$g^*_j(B, T) \leq g_j(B, T) \tag{4}$$

Proof: Let B be the only element of the optimal sequence. If B is on OPEN, then $g^*_j(B, T) = g^*_o(B, B)$

† See further remarks concerning this under Step 9.

$= g_o(B, B) = 0$. This is the trivial case. Next let X_i be on CLOSED:

$$\Gamma_t^{m/n}(i, j)[X_i] = X_j \tag{5}$$

From Equation (1)

$$g^*_j(B, T) = g(B, j) + C^*(j, T) \tag{6}$$

Also

$$g_j(B, T) = g(B, j) + C(j, T) \tag{7}$$

Subtracting Equation (7) from Equation (6), we get

$$g^*_j(B, T) = g_j(B, T) + C^*(j, T) - C(j, T) \tag{8}$$

But, since $C^*(j, T) \leq C(j, T)$, Equation (4) holds.

Corollary I: The evaluation function $g^*_j(B, T)$ is a lower limit of the actual cost $g_j(B, T)$ if $C^*(j, T) \leq C(j, T)$ for all X_j.

Corollary II: If $X_j = T$, then $g^*_j(B, T) = g_j(B, T)$.

Proof: From Equation (8), if $C^*(j, T) \leq C(j, T)$, then $g^*_j(B, T)$ is the lower limit of $g_j(B, T)$. If $X_j = T$, $C(j, T) = C^*(j, T) = C(T, T) = 0$. Therefore, from Equation (8), $g^*_j(B, T) = g_j(B, T)$.

The following theorem can now be proved.

Theorem: If $C^*(j, T) \leq C(j, T), \forall\ X_j$ then the algorithm is admissible.

Proof: Assume the algorithm leads the search to the goal state without finding the optimal sequence and that node X_k was the last node expanded. Then

$$g^*_k(B, T) = g_k(B, T) > g_j(B, T)$$

where $g_j(B, T)$ is the actual cost of the optimal sequence. By Corollaries I and II of the Lemma, there is a node X_j on OPEN for which

$$g^*_j(B, T) \leq g_j(B, T) < g^*_k(B, T)$$

Therefore, X_j would be selected for expansion rather than X_k. But this contradicts the assumption that the search of the algorithm terminates without finding the optimal sequence. Thus, the theorem is proved.

The above proofs are easily extended to show that if $S_n > 1$, the algorithm will generate the next best $(S_n - 1)$ sequences in order of increasing cost.

It remains now to demonstrate that

$$C^*(j, T) \leq C(j, T), \ \forall\ X_j \tag{9}$$

where C^* is given by Equation (3). A formal mathematical proof of Equation (9) may not be possible for the general case because of the highly nonlinear nature of the equilibrium stage equations. However, consider the case of ordinary distillation under the assumptions of constant molal overflow and constant relative volatility. Assume further that column pressure and the feed phase condition for the separation of two key components are specified independently of the presence of any nonkey components. Separation cost will depend mainly on the required number of equilibrium stages and the required boil up rate. To determine the influence of nonkey components on stages and boil up rate, assume the applicability of the Fenske equation for minimum stages

$$N_{\min} = \frac{\log\left[\left(\frac{S_{f,LK}}{1 - S_{f,LK}}\right)\left(\frac{1 - S_{f,HK}}{S_{f,HK}}\right)\right]}{\log \alpha_{LK,HK}} \tag{10}$$

the approximate Gilliland equation for minimum reflux; and the Gilliland correlation for relating stages to minimum stages, minimum reflux, and actual reflux. These are all described by Robinson and Gilliland (1950).

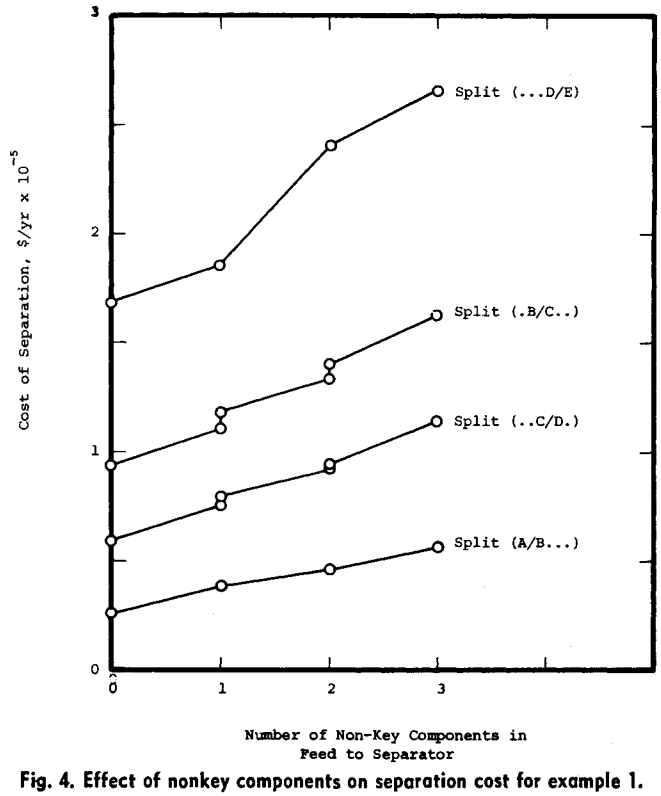


Fig. 4. Effect of nonkey components on separation cost for example 1.

Equation (10) shows that the number of minimum stages is not influenced by nonkey components. However, the boil up rate is influenced. Based on an actual reflux given by the Gilliland equation* for a saturated liquid feed, the boil up rate is

$$V = \sum_{i=1,LLK} \frac{\alpha_{i,HK} + 0.25\, n_{fi}}{\alpha_i - 1} + V_B + \sum_{j=HHK,N_c} \frac{1.25\, \alpha_{j,HK} n_{fj}}{\alpha_{LK} - \alpha_j} \tag{11}$$

where V_B is the boil up rate for a feed containing only the two key components. All relative volatilities in Equation (11) are referred to the heavy key component. Since all terms on the right-hand side of Equation (11) must be positive, $V \geq V_B$. Therefore, it is clear that the presence of nonkey components tends to increase the minimum reflux rate (or minimum boil up rate) and causes the required number of stages to be greater for a given ratio of actual reflux to minimum reflux. Thus, the separation cost is greater when nonkey components are present. Similar demonstrations can be made for other separation operations (for example, absorption, extractive distillation, liquid-liquid extraction), where a mass separating agent is distinguished from a nonkey component and may be added to increase the relative volatility between key components.

Figure 4 illustrates quantitatively the effect of nonkey components on separation costs. This plot shows the costs of all twenty separation subproblems arising from the five-product case involving paraffin hydrocarbons considered by Rathore et al. (1974), wherein separations are made only by ordinary distillation. Annual costs are plotted vs. the number of nonkey components in the feed to the separator. Successive sets of lines connect the data points for each of the four feasible pairs of key

* As written, Equation (11) assumes that a perfect separation is achieved between the key components and that key component mole ratios in the pinch zone and feed zone compositions are equal.

components. In all cases, costs are lowest for the binary-feed separations. Thus, for this case, the heuristic cost function given by Equation (3) is valid. Although not required for the method presented here, it is also interesting to note that costs in Figure 4 always increase with increasing number of nonkey components.

EXAMPLES AND COMPARISONS

Four examples were solved by the predictor based ordered search with the heuristic cost function used. One of these is presented in detail, followed by a comparison of the results of all four examples with the procedures of Rodrigo and Seader (1975). In all cases, cost data were taken from the previous studies cited.

Example 1: Separation of Light Paraffins

Consider the example used by Rathore et al. (1974) involving the separation by ordinary distillation (I) of propane (A), isobutane (B), n-butane (C), isopentane (D), and n-pentane (E). Fourteen different sequences are possible involving twenty unique separation subproblems. It is desired to determine the optimal sequence and the next best sequence. Although costs of all separation subproblems are given in Figure 4, the procedure outlined here requires that these costs be computed only when and if needed. By the algorithm of Figure 3, the desired sequences are obtained as follows:

Steps 1 and 2:

$$B = X_o = \{ABCDE\}$$

$$\text{OPEN} = \{X_o\}$$

$$g^* = 0$$

$$T = \{A\} \wedge \{B\} \wedge \{C\} \wedge \{D\} \wedge \{E\}$$

$$\text{CLOSED} = \{T\}$$

Steps 3, 4a, and 5a:

$$\text{CLOSED} = \{T, X_o\}$$

$$X_o \neq T$$

X_o is not flagged

Steps 6, 7a, and 7b: (Since $i = 0$)

i,j	m/n	$x_j = \Gamma_I^{m/n}(i,j)[X_i]$	$M(i,j)$	$g(B,j) = g^*_j(i,j) = C$, \$/yr $\times 10^{-5}$
0,1	A/B	$\{A\} \wedge \{BCDE\}$	1	0.5715
0.2	B/C	$\{AB\} \wedge \{CDE\}$	1	1.6500
0,3	C/D	$\{ABC\} \wedge \{DE\}$	1	1.1490
0,4	D/E	$\{ABCD\} \wedge \{E\}$	1	2.6600

Flag all X_j

Step 2: OPEN $= \{X_1^0, X_3^0\ X_2^0, X_4^0\}$

Steps 3, 4a, and 5a:

$$\text{CLOSED} = \{T, X_1^0\}$$

$$X_1^0 \neq T$$

X_1^0 is flagged

Step 5b:

$$g^*_1(B,T) = g(B,1) + C^*(1,T)$$

where $C^*(1, T)$ = costs of binary separations B/C, C/D, and D/E

Therefore, $g^*(B, T) = (0.5715 + 0.9493 + 0.5927 + 1.6920) \times 10^{-5}$

$$= 3.8055 \times 10^{-5} \text{ \$/yr}$$

Step 2: OPEN $= \{X_3^0, X_2^0, X_4^0, X_1\}$

In succeeding steps, flags are also sequentially removed from X_3, X_2, and X_4 to give

$$\text{OPEN} = \{X_1\ X_3\ X_2\ X_4\}$$

Corresponding

$$g^* \times 10^{-5} \text{ \$/yr: } 3.8055\ 4.0516\ 4.1960\ 4.4633$$

Although all flags are removed in this example, in some separation problems values of certain X_i^0 may be of such magnitude that they may never advance to the first position in OPEN.

Steps 3, 4a, and 5a:

$$\text{CLOSED} = \{T, X_1\}$$

$$X_1 \neq T$$

X_1 is not flagged

Step 6:

i,j	m/n	$X_j = \Gamma_I^{m/n}(i,j)[X_i]$	$M(i,j)$	$C(i,j)$, \$/hr $\times 10^{-5}$
1,5	B/C	$\{A\} \wedge \{B\} \wedge \{CDE\}$	1	1.3340
1,6	C/D	$\{A\} \wedge \{BC\} \wedge \{DE\}$	1	0.9443
1,7	D/E	$\{A\} \wedge \{BCD\} \wedge \{E\}$	1	2.4180

Steps 7a and 8:

Costs, \$/yr $\times 10^{-5}$

j	$g(B,1)$	$C^*(j,T)$	$g^*_j(B,T)$ from Equation (1)
5	0.5715	0.5927 + 1.6920 = 2.2847	4.1902
6	0.5715	0.9493 + 1.6920 = 2.6413	4.1571
7	0.5715	0.9493 + 0.5927 = 1.5420	4.5315

Step 9:

$$\text{CLOSED} = \{T\}$$

$$\text{HOLD} = \{X_1\}$$

Step 2:

$$\text{OPEN} = \{X_3\ X_6\ X_5\ X_2\ X_4\ X_7\}$$

corresponding

$g^* \times 10^{-5}$, \$/yr

4.0516 4.1571 4.1902 4.1960 4.4633 4.5315

The next cycle gives

i,j	m/n	X_j	$g^*_j(B,T) \times 10^{-5}$ \$/yr
3,8	A/B	$\{A\} \wedge \{BC\} \wedge \{DE\}$	4.1856
3,9	B/C	$\{AB\} \wedge \{C\} \wedge \{DE\}$	4.3003

$$\text{OPEN} = \{X_6\ X_8\ X_5\ X_2\ X_9\ X_4\ X_7\}$$

corresponding

$g^* \times 10^{-5}$, \$/yr

4.1571 4.1856 4.1902 4.1960 4.3003 4.4633 4.5315

Steps 3, 4a, 4b, and 4c: CLOSED $= \{T, X_6\}$

since $X_6 = \{A\} \wedge \{BC\} \wedge \{DE\}$, $C(6, T) = C^*(6, T)$

because the only separations remaining are B/C and D/E.

Therefore, $g(B, T) = g^*_6(B, T) = 4.1571 \times 10^5$ \$/yr

The optimal sequence generated is

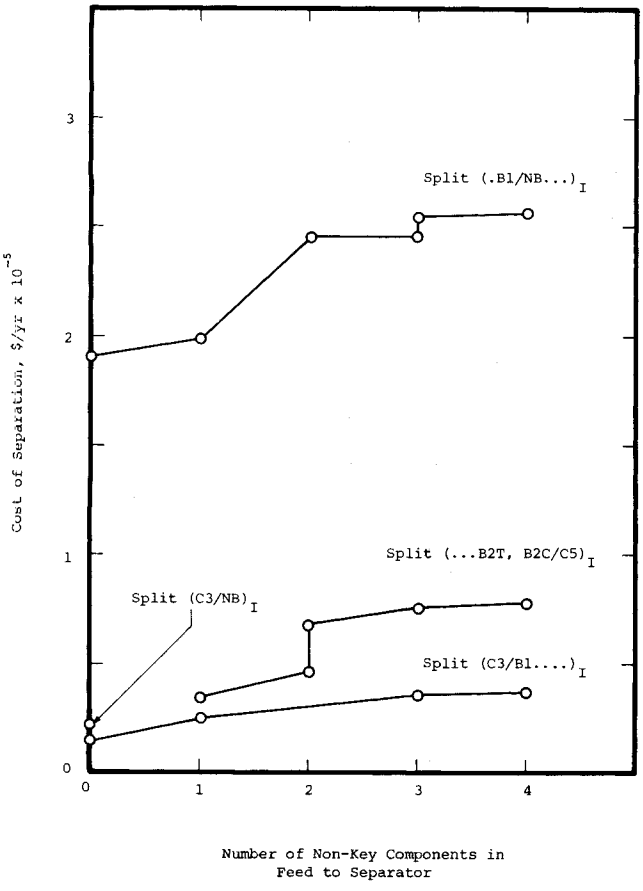
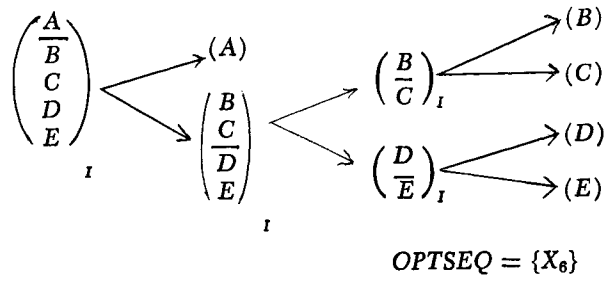


Fig. 5. Effect of nonkey components on separation cost for example 2—part 1.

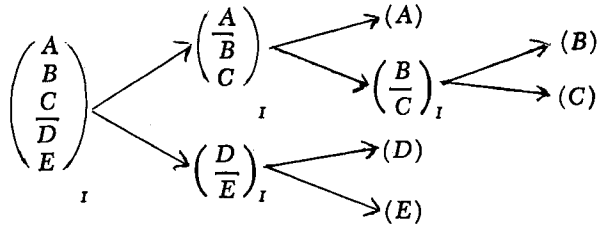


$$OPTSEQ = \{X_6\}$$

No. of optimal sequences generated $= 1 \neq S_n = 2$

In a similar manner, the next best sequence is found without further node expansions to be X_8, with $g(B, T) = g^*_8(B, T) = 4.1856 \times 10^5$ \$/yr.

The sequence generated is:



In order to generate these two sequences, only thirteen of twenty (or 65%) unique separation subproblems required analysis.

Example 2: Synthesis of a Butylenes Purification System

Hendry and Hughes (1972) presented a separation problem of industrial importance involving the separation of a stream containing propane, *n*-butane, butene-1, trans-butene-2, cis-butene-2, and pentane into four products (including one multicomponent product consisting of the three butene isomers) by the use of ordinary distillation (*I*) and/or extractive distillation with 96 wt% aqueous furfural (II). By specifying certain forbidden splits, the number of unique separation subproblems in the search space was reduced to 30 as discussed by Rodrigo and Seader (1975). The subproblem costs were derived from the tabulations of Hendry (1972) and are shown in Figures 5 and 6. For Type II separators, binary separations between the key components are made in the presence of the mass separating agent, and costs include subsequent recovery of the aqueous furfural for recycle. For both types of separators, costs for separating two given key components are seen to be lowest for separator feeds involving the least number of nonkey components. Application of the predictor based ordered search procedure leads to the optimal solution after analyzing only 70% (21 of 30) of the unique separation subproblems.

Example 3: Separation of Light Hydrocarbons

Thompson and King (1972) presented a six-component case involving ethane (1), propylene (2), propane (3), 1-butene (4), *n*-butane (5), and *n*-pentane (6). This problem was solved after it was modified to specify six (rather than four) relatively pure products to be obtained by ordinary distillation only. Forty-two different sequences were possible. Costs for the thirty-five unique subproblems are plotted in Figures 7 and 8. This time, the heuristic cost function is seen to fail in Figure 8 when the key components are 1-butene and butane. Nevertheless, the optimal sequence was generated by analyzing only 60% of the unique subproblems.

The failure of the heuristic cost function was found to be due to an increase in relative volatility between 1-butene and *n*-butane, which occurred when a paraffinic third compound was added. For example, the average relative volatility was 1.197 with only the two key com-

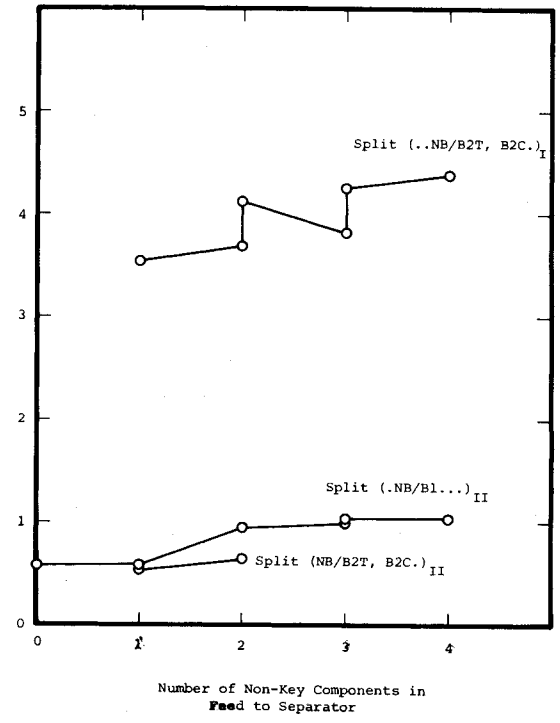


Fig. 6. Effect of nonkey components on separation costs for example 2—part 2.

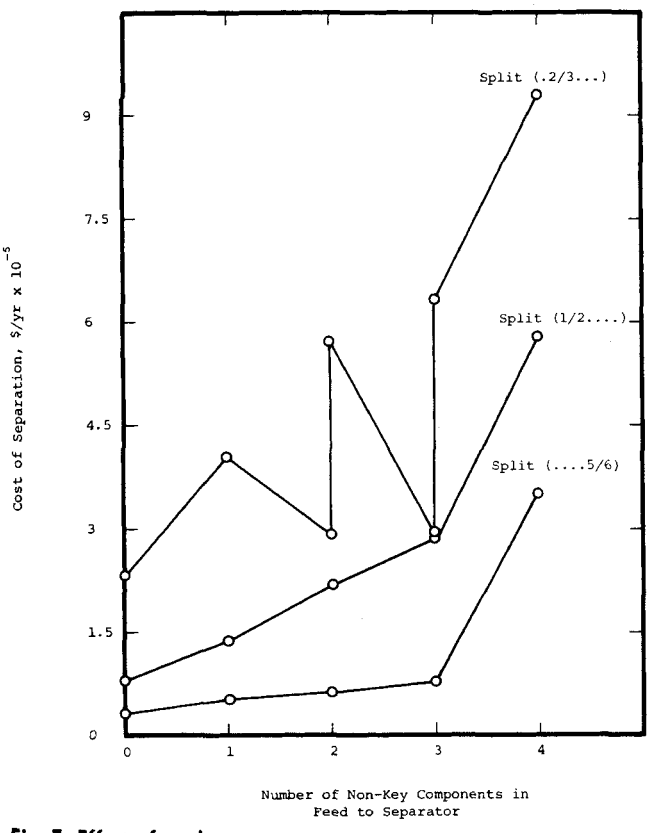


Fig. 7. Effect of nonkey components on separation costs for example 3—part 1.

ponents present. When n-pentane was added, the relative volatility increased to 1.226 owing to an increase in the activity coefficient for 1-butene and a decrease in the activity coefficient for n-butane according to regular solution theory. The increase in relative volatility resulted in a 12% decrease in the number of trays.

Example 4: Separation of Mixed Hydrocarbons

Rodrigo and Seader (1975) considered in detail a case involving the separation of n-hexane, benzene, and cyclohexane by ordinary distillation and/or extractive distillation with phenol. Using the predictor based ordered search procedure, the optimal solution was found by analyzing only eight of sixteen unique subproblems.

The results of these four examples are compared to the results obtained by Rodrigo and Seader (1975) in Table 1, where

$$F_{usp} = \frac{\text{number of unique subproblems analyzed}}{\text{number of unique subproblems}} = \frac{N_{uspa}}{N_{usp}}$$

TABLE 1. COMPARISON OF SEARCH ALGORITHMS

	F_{usp}, %	
Example	Ordered branch search	Predictor based ordered search
1	100.0	65.0
2	76.7	70.0
3	100.0	60.0
4	62.5	50.0

For all four examples, both methods found the optimal sequences. However, in all cases, the predictor based ordered search required less analysis than the ordered branch search. Significant reductions in analysis occurred in Examples 1 and 3, where sequence cost distributions were narrow.

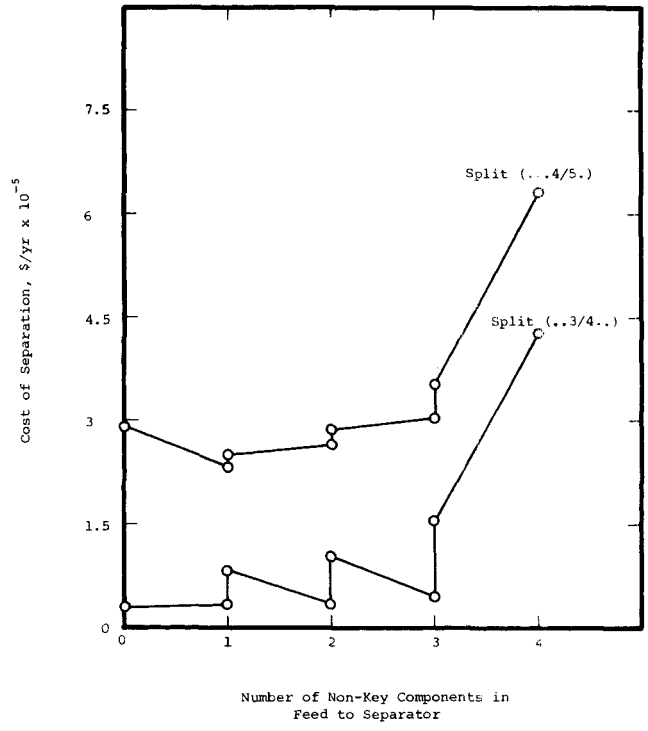


Fig. 8. Effect of nonkey components on separation costs for example 3—part 2.

NOTATION

A = complete set of arcs, $\{C\}$
B = beginning or start node, $B \epsilon N$; $B \neq T$
$C(i, j)$ = actual cost, associated with an arc, for expansion of node X_i to node X_j; an element of A
$C(j, T)$ = actual cost for a partial sequence extending from node X_j to goal node T
$C^*(j, T)$ = heuristic cost function or estimated cost for completing a sequence that starts with node X_j
F_s = sequence search factor, N_{sd}/N_s
F_{sp} = subproblem search factor, N_{spa}/N_{sp}
F_{usp} = unique subproblem search factor, N_{uspa}/N_{usp}
G = graph of search space, $[N, A, \phi]$
$g(B, i)$ = actual cost for developing a partially completed sequence ending with node X_i
$g_j(B, T)$ = actual cost for a complete sequence passing through node X_j
$g^*_j(B, T)$ = estimated cost of complete sequence passing through node X_j
$M(i, j)$ = entry in the reachability matrix equal to 1 if nodes X_i and X_j are connected by an arc; otherwise, 0
n_{fi} = flow rate of component i in separator feed
N = complete set of nodes, $\{B, X_i, T\}$
N_C = number of components
N_{min} = minimum number of equilibrium stages
N_s = number of possible sequences
N_{sd} = number of sequences developed
N_{sp} = number of possible subproblems
N_{spa} = number of subproblems analyzed
N_{usp} = number of unique subproblems
N_{uspa} = number of unique subproblems analyzed
S = set of optimal and near optimal sequences
$S_{f,i}$ = split fraction of component i

S_n = number of optimal and near optimal sequences desired
T = terminal or goal node; $T \epsilon N$; $T \neq B$
V = boil up vapor flow rate
X_i = node representing a partially completed sequence; an element of N
$\alpha_{m,n}$ = relative volatility between components m and n
$\Gamma_t^{m/n}(i,j)$ = an arc or separation operator for expanding node X_i into successor node X_j using separator type t, light key component m, and adjacent heavy key component n; an element of ϕ; acts as a pointer from node X_j back to node X_i
ϕ = complete set of feasible operators, $\{\Gamma\}$

Other Symbols

ϵ = element of
$\{\}$ = set
$\wedge$ = and
$\forall$ = for all

LITERATURE CITED

Hart, E. H., N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Systems Cybernetics*, 4, No. 2, 100 (1968).

Hendry, J. E., "Computer Aided Synthesis of Optimal Multicomponent Separation Sequences," Ph.D. thesis, Univ. Wisc., Madison (1972).

———, and R. R. Hughes, "Generating Separation Process Flowsheets," *Chem. Eng. Progr.*, 68, No. 6, 69 (1972).

King, C. J., *Separation Processes*, McGraw-Hill, New York (1971).

Motard, R. L., M. Shacham, and E. M. Rosen, "Steady State Chemical Process Simulation," *AIChE J.*, **21**, 417-436 (1975).

Nilsson, N. J., *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill, New York (1971).

Rathore, R. N. S., K. A. Van Wormer, and G. J. Powers, "Synthesis of Distillation Systems with Energy Integration," *AIChE J.*, **20**, 940 (1974).

Robinson, C. S., and E. R. Gilliland, *Elements of Fractional Distillation*, 4 ed., McGraw-Hill, New York (1950).

Rodrigo, F. R., and J. D. Seader, "Synthesis of Separation Sequences by Ordered Branch Search," *AIChE J.*, **21**, 885 (1975).

Rudd, D. R., J. J. Siirola, and G. J. Powers, *Process Synthesis*, Prentice-Hall, Englewood Cliffs, N.J. (1973).

Seader, J. D., W. D. Seider, and A. C. Pauls, "FLOWTRAN Simulation—An Introduction," CACHE, Cambridge, Mass. (1974).

Thompson, R. W., and C. J. King, "Synthesis of Separation Schemes," Technical Report No. LBL-614, Lawrence Berkeley Laboratory (July, 1972*a*).

———, "Systematic Synthesis of Separation Schemes," *AIChE J.*, **18**, 941 (1972b).

Westerberg, A. W., and G. Stephanopoulos, "Studies in Process Synthesis-I," *Chem. Eng. Sci.*, **30**, 963 (1975).

Manuscript received May 28, 1976; revision received August 6, and accepted August 9, 1976.

Application of the Limiting Current Method to Mass Transfer in Packed Beds at Very Low Reynolds Numbers

PETER W. APPEL and JOHN NEWMAN

Materials and Molecular Research Division, Lawrence Berkeley Laboratory, and Department of Chemical Engineering, University of California, Berkeley, California 94720

The limiting current technique is used to obtain mass transfer coefficients at very low Reynolds numbers ($Re < 0.1$) in a packed bed consisting of stainless steel spheres. The data show that for $Re < 0.015$, the Nusselt numbers are below those predicted by existing empirical correlations. The results are discussed in the light of some mathematical models for packed beds reported in the literature and show semiquantitative agreement with the values predicted by Sorensen and Stewart (1974*a*).

SCOPE

A large number of mass transfer processes require a liquid flowing at low velocities through a porous electrode or a packed bed. At these velocities ($Re << 1$) and in the absence of natural convection, mass transfer coefficients in packed beds are only dependent upon the Péclet number. Experimental results in the low Reynolds number region, low Péclet number region are scarce and scatter widely. Therefore, reliable data in said region are important; to obtain these data the limiting current technique was used in a packed bed consisting of stainless steel spheres. For our experiments we have chosen the cathodic reduction of ferricyanide to ferrocyanide in excess supporting electrolyte. Procedures to minimize possible sources of error and design criteria are given.

CONCLUSIONS AND SIGNIFICANCE

It has been shown that the limiting current method is suitable for obtaining mass transfer coefficients in packed beds, in particular at very low Reynolds numbers, provided proper design and experimental procedures are followed.

Peter W. Appel is with Unilever Research, Vlaardingen, The Netherlands.