The Separation Network Design Space

D. W. Tedder
School of Chemical Engineering
Georgia Institute of Technology
Atlanta, GA 30332

M. Shoaei
CIBA-GEIGY Corp.
McIntosh, AL 36553

The design of chemical processes usually involves one or more chemical separation steps which may be analyzed using dynamic programming (DP). The number of alternative separation trains, which may be synthesized (or designed), increases dramatically with the number of final products produced by a separation network, and there are often several alternative unit operations which may be selected as DP "stages." Many such stage combinations result in feasible separation trains.

Dynamic programming (Bellman, 1957; Dreyfus and Law, 1977) has been used to address both optimization (Aris, 1961) and synthesis problems (Hendry, 1977; Hendry and Hughes, 1972; Shoaei, 1987; Shoaei et al., 1987). Synthesis applications have mostly dealt with the optimal design of process networks that can separate a single feed into n final products. Generally, investigators have only considered simple towers that split a single feed into two products. Relatively few have considered complex fractionators capable of two or more splits (Tedder and Rudd, 1978; Minderman and Tedder, 1982). Here, DP methodology is used to define the separation network design space in terms of the number of alternative stages that require suboptimization $D(S)$, and the number of structurally distinct separation networks T_n that may be generated from these stages.

Each alternative structure (or separation network) should be considered at its optimal operating state. This goal can be stated as:

$$\min_{q_i, p_i} \{\ell_i(q_i, p_i)\} \tag{1}$$

for $i = 1, 2, \ldots, T_n$ where $q_i \in Q$ and $p_i \in P$ are the decision and stream vectors, respectively. Each optimization is subject to the overall energy and material balance constraints, $p_i = h_i(p_0, q_i)$, and each element in these vectors has an upper and lower bound defined by the feasible regions Q and P. Nonlinear search methods may be used to find q_i^*, the optimal value for q_i, and each of the T_n networks should be compared at its respective q_i^* value while yielding the same final products, p_0.

Correspondence concerning this paper should be addressed to D. W. Tedder.

Species Allocation

The size of the species allocation space S and the number of separation stages and networks all depend upon the number of final products n that are required. While most synthesis studies have focused on the separation of a single feed, S can include the analysis of multiple feeds if pseudo final products are defined as members of p_0. That is, the final products must be partitioned to reflect their feed source as well as their final blended compositions and amounts. (If a final product originates from multiple feeds, it must be divided into that portion coming from each. The number of final products defined in the species allocation is then greater than the actual number produced by the network. The pseudoproducts in p_0, however, can be blended to produce the actual final products. With this definition, all intermediate streams that may occur in a multiple-feed problem are contained in S.) With this provision for multiple feeds, $p_i \subset S \; \forall i = 1, 2, \ldots, T_n$ (i.e., for all possible networks) as required for synthesis. The final partition, therefore, with any separation train is the set:

$$S_1 = \{A_{1,1}, A_{1,2}, \ldots, A_{1,n}\} = p_0 \tag{2}$$

Similarly, the partitions of binary, ternary, etc. stream mixtures can be represented as:

$$S_2 = \{A_{2,1}, A_{2,2}, \ldots, A_{2,n-1}\} \tag{3}$$

$$S_3 = \{A_{3,1}, A_{3,2}, \ldots, A_{3,n-2}\} \tag{4}$$

$$\vdots$$

$$S_i = \{A_{i,1}, A_{i,2}, \ldots, A_{i,j}, \ldots, A_{i,n-i+1}\} \tag{5}$$

$$\vdots$$

$$S_{n-2} = \{A_{n-2,1}, A_{n-2,2}, A_{n-2,3}\} \tag{6}$$

$$S_{n-1} = \{A_{n-1,1}, A_{n-1,2}\} \tag{7}$$

$$S_n = \{A_{n,1}\} = p_{n-1} \tag{8}$$

and the set:

$$S = \{S_1 \cup S_2 \cup \cdots \cup S_i \cup \cdots \cup S_n\} = P \tag{9}$$

contains all possible streams that may connect stages in a network as well as the original feeds and the final products. In the case of multiple feeds, they are members of S_i where $1 < i < n$ and S_n is a composite feed. Otherwise, S_n is the single original network feed. Thus, S includes the complete species allocation database needed to solve any synthesis separation problem and $\dim S = n(n+1)/2$ is the total number of required streams in this database.

The direct sequence in Figure 1, for example, is a particular case where each stage produces at least one member of S_1. The bottom products are members of sets S_1 through S_{n-1} and $p_i \subset S$. Other sequences utilize different members of subsets S_2 through S_n while still producing the final product set S_1.

The nature of chemical separations imposes an additional constraint. The members of the partition set S can usually be ordered in only one way. (An exception occurs if the property rank order changes for $q_i \in Q$. For example, the relative volatilities of some species in nonideal mixtures may change with stream compositions. They may also change with temperature in certain instances.) This order is determined by that physical property which is exploited to achieve the stage separation. For example, if distillation is used, then the members of S are ordered by decreasing volatility. If liquid/liquid extraction is the stage operation, then the species are ordered by decreasing distribution coefficients, and similar constraints exist for all separation operations.

Network Decomposition and Synthesis

The network problem is analyzed by dividing it into stages (e.g., distillation towers). In the case of optimization alone, the structure of these stages is fixed. In the case of simultaneous optimization and synthesis, the stage structure is variable.

An effective decomposition exploits those network locations where information recycle can be minimized. For example, distillation generally involves the countercurrent flow of vapor and liquid streams. So a decomposition within such a countercurrent cascade is not attractive. It is more useful to choose the feeds entering the cascade and the products leaving it when defining $p_i \in S$ because the flow at such points is more generally acyclic.

With a reasonable topological strategy, however, virtually all separation networks can be decomposed into acyclic stages and all feasible trains can be represented as a combined diverging and converging DP problem. The optimal network will only use a subset of those stages defined by S, and the feed to a stage is not necessarily the output from the previous stage or *vice versa*. In any event, a minimum cost path is chosen during synthesis.

The simplest case consists of a series of acyclic stages such as the direct sequence shown in Figure 1. The kth stage has an input vector p_k, and output vector p_{k-1}, and a decision vector q_k. The input vector is simply the feed, and the output vector consists of the separation products produced by that tower (or stage). As shown, each stage produces only two products, but a single stage could produce up to n products.

The decision vector is also the state or optimization vector for the stage. In simple distillation towers producing only two products, q_k might consist of variables such as the tower operating pressure, the reflux ratio, the feed enthalpy, and the degrees of reflux subcooling. More complex tower designs usually include additional degrees of freedom.

The output of the kth stage is a function of p_k and q_k: $p_{k-1} = h_k(p_k, q_k)$ due to the energy and material balance requirements. A gain (profit) or loss (cost) function also exists for the stage where either $g_k = g_k(p_{k,pk-1}, q_k)$ for gain or for costs $\ell_k = \ell_k(p_k, p_{k-1}, q_k)$.

A stage input is the output from the previous stage. Using DP, one first optimizes the last stage, then the next to the last, and so on until the entire process is optimized. For the last stage, the cost minimization objective becomes:

$$f_1(S_1) = \min_{q_1, p_1} \{\ell_1(p_1, p_0, q_1) + f_0(p_0)\} \tag{10}$$

with $p_1 \in P$, $q_1 \in Q$, $p_0 = h_1(p_1, q_1)$ and the boundary conditions $f_0(p_0) = 0$. The recursive DP loss function defining the optimal structure becomes:

$$f_k(S_k) = \min_{q_k, p_k} \{\ell_k(p_k, p_{k-1}, q_k) + f_{k-1}(S_{k-1})\} \tag{11}$$

where $\ell_k(p_k, p_{k-1}, q_k)$ must be evaluated for all stages and $f_{k-1}(S_{k-1})$ varies with the design.

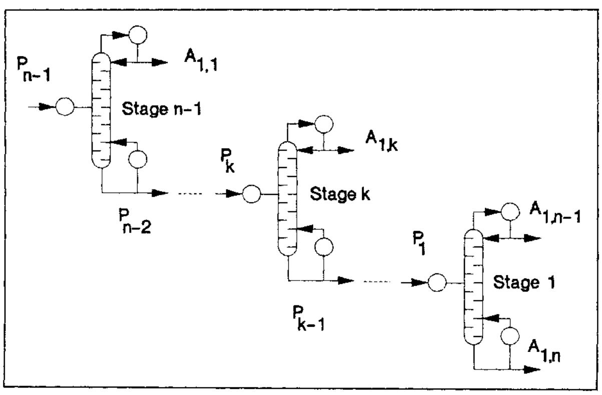

Figure 1. DP stage representation of the direct sequence of simple distillation towers.

Single-Stage Space

Eventually $n-1$ separations will be required somewhere in the train. These separations may be carried out entirely by an individual DP stage or by a series of stages that are connected by streams which are defined by an appropriate partition space S. If each stage completes only one separation, then $n-1$ stages are required. The direct sequence in Figure 1 is an example of this case. If a stage produces all n products, then $p_i = \{S_1 \cup S_n\}$ and only one stage is required. Design (h) in Figure 2 is an example of this case for $n = 7$.

Thus, a single stage may produce from two to n products. There are $(n-1)/1!$ ways of dividing a feed containing n final products into two intermediate streams, $(n-1)(n-2)/2!$ ways of dividing it into three intermediate streams, and $(n-1)(n-2)(n-3)/3!$ ways of dividing it into four streams, etc. Hence, the number of ways a single stage can be operated to

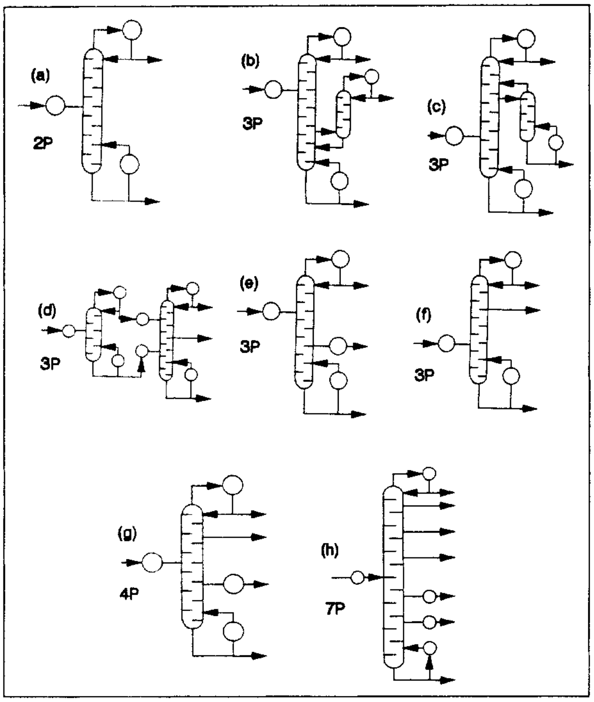

Figure 2. Alternative DP stage designs for use in network separations.

yield from two to n products is given by:

$$\sum_{i=1}^{n-1} \frac{1}{i!} \prod_{j=1}^{i} (n - j) \tag{12}$$

However, multiple design "types" usually exist for a single stage even if only one separation technology is permitted. Figure 2, for example, indicates several design types for a single DP stage that may occur in distillation trains. Design (a) produces only two products (a 2P design). Design types (b), (c), (d), (e), and (f) define stages where a single feed is split into three products (all 3P designs). Similarly, Designs (g) and (h) are designated as 4P and 7P designs because they produce four and seven products, respectively, in a single DP stage.

The number of multiple product (2P, 3P, 4P, etc.) design types that are feasible increases with the number of products produced by a single stage and the technology that one is willing to consider. Even if the technology is limited to simple distillation, however, additional 2P design types exist other than design (a) in Figure 2. Simple towers, for example, can include midreboilers or midcondensers. In this case, p_k and p_{k-1} are the same as for design (a) in Figure 2, but q_k has additional degrees of freedom (i.e., the midreboiler or midcondenser location in the tower and its duty). Thus, simple towers with such modifications constitute alternative design types with different degrees of freedom.

If $E(n)$ designates the number of design types that are permitted for a single stage producing n products, then the size of the single stage design space, D_n, is:

$$D_n = \sum_{i=1}^{n-1} \frac{E(i + 1)}{i!} \prod_{j=1}^{i} (n - j) \tag{13}$$

If other chemical separation techniques are allowed, and the product order in S is different for these alternatives, then the partition and design spaces for a single stage are increased further. In general, D_n must be computed for each separation technology in which products are ranked differently in S.

Table 1 lists values for D_n for $n = 2, 3, \ldots, 19$. These D_n values only consider one design type so $E(i) = 1$, for $i = 2, 3, \ldots, n$. Limiting the stages to simple towers (i.e., one 2P design only, with $E(2) = 1$ and $E(i) = 0$, for $i = 3, 4, \ldots, n$) severely limits the number of possibilities for stage operation. Table 1 shows, for example, that <1% of the single-stage possibilities are considered if only design (a) is allowed and $n = 8$.

Total Stage Space

For an n product problem, the total number of single-stage designs that must be evaluated can be estimated as a sum involving $D_2, D_3, \ldots, D_n$. As indicated in Eqs. 3–5, there are $(n - 1)$ binary mixtures, $(n - 2)$ ternary mixtures, etc. that must be resolved into final products. Hence, the total number of single-stage designs that must be evaluated to solve the general network synthesis problem is given by:

$$D(S) = \sum_{i=1}^{n-1} (n - i)\, D_{i+1} \tag{14}$$

Network Space

The variable T_n is the number of structurally distinct separation sequences that may be generated using the partition S and after evaluating $D(S)$ single-stage designs. The number of sequences resulting from only one binary design (i.e., 1–2P design, with $E(2) = 1$ and $E(i) = 0$, for $i = 3, 4, \ldots, n$) may be calculated from:

$$T_n = \frac{[2(n - 1)]!}{(n - 1)!n!} \tag{15}$$

which was found by Heaven (1969) and Thompson and King (1972). This number can also be calculated from the Catalan numbers series (Shoaei and Sommerfeld, 1986). Values for T_n where $n = 2, 3, \ldots, n$ are shown as Case 1 in Table 2.

Case 1 in Table 2 may also be computed recursively (Tedder, 1975). The subscript in the recursive variable R_i indicates the number of final products that can be generated from each product generated by a single stage. The number of separation sequences is computed by adding the permutations of the appropriate R_i product terms. The number of R_i variables in

Table 1. Design Space for a Single Stage As a Function of the Number of Intermediate Streams, *n*, Produced by That Stage

n	D_n	n	D_n	n	D_n
2	1	8	127	14	8,191
3	3	9	255	15	16,383
4	7	10	511	16	32,767
5	15	11	1,023	17	65,535
6	31	12	2,047	18	131,071
7	63	13	4,095	19	262,143

Table 2. Number of Possible Sequences When Only One Simple Binary (1–2P) Stage and Several Combinations Producing Three, Four and Five Products Are Allowed

Total Final Products	Case 1 1–2P Only	Case 2 1–2P, 5–3P Only	Case 3 1–2P, 1–3P Only	Case 4 1–2P, 1–3P, 1–4P	Case 5 1–2P, 1–3P, 1–4P, 1–5P
2	1	1	1	1	1
3	2	7	3	3	3
4	5	30	10	11	11
5	14	194	38	44	45
6	42	1,162	154	189	196
7	132	7,782	654	850	894
8	429	52,239	2,871	3,951	4,215
9	1,430	364,705	12,925	18,832	20,377
10	4,862	2,579,577	59,345	91,542	100,463
11	16,796	18,592,431	276,835	452,075	503,191
12	58,786	135,564,936	1,308,320	2,261,753	2,553,291
13	208,012	999,905,592	6,250,832	11,439,372	13,097,469
14	742,900	7,441,209,400	30,142,360	58,394,014	67,808,104
15	2,674,440	55,825,343,240	146,510,216	300,455,892	353,851,124

each product term equals the number of products generated by a single stage, and R_i is the number of downstream sequence possibilities at that point. The sum of the R_i subscript values in a product must always equal n.

A simple stage [e.g., design (a) in Figure 2], produces only two products. The sequence permutations may be calculated by adding the binary products R_iR_j that can be formed subject to the constraint $i + j = n$. Similarly, those sequence permutations which include stages that produce three products are included by adding product terms with the form $R_iR_jR_k$ and the constraint $i + j + k = n$. Stages producing four products are enumerated by adding product terms with the form $R_iR_jR_kR_l$, etc. until a single stage yielding n final products is included. If this procedure is carried out recursively so that $R_i = T_i$, for $i = 3, 4, \ldots, n$ and the correct number of design types, $E(i)$, premultiplies the sum for each type of stage operation (for each product term, 2P, 3P designs, etc.), then T_n will indicate the total number of network sequences that can be synthesized.

The T_n values shown as Case 2 in Table 2, for example, were computed recursively using two product terms:

$$T_n = E(2) \sum_{i=1}^{n-1} R_iR_{n-i} + E(3) \sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} R_iR_{j-i}R_{n-j} \tag{16}$$

where $\{R_1, R_2\} = \{1, E(2)\}$, $E(2) = 1$, and $E(3) = 5$. Similarly, Case 1 in Table 2 may be computed using either Eq. 15 or the first product term on the righthand side of Eq. 16 with $\{R_1, R_2\} = \{1, E(2)\}$ and $E(2) = 1$.

If single stages generating four products are allowed, then the righthand side of Eq. 16 should include the additional product term:

$$\cdots + E(4) \sum_{k=1}^{n-3} \sum_{i=k+1}^{n-2} \sum_{j=i+1}^{n-1} R_kR_{i-k}R_{j-i}R_{n-j} + \cdots \tag{17}$$

and so forth until the design space is covered when a single stage is allowed to produce all n final product streams. In this case, the righthand side of Eq. 16 includes term 17 and the sum of all analogous intermediate terms together with the final product term:

$$\cdots + E(n) \sum_{k_1=1}^{1} \sum_{k_2=k_1+1}^{2} \cdots \sum_{k_{n-1}=1+k_{n-2}}^{n-1} \cdot R_{k_1} \left(\prod_{j=2}^{n-1} R_{k_j - k_{j-1}} \right) R_{n-k_{n-1}} \tag{18}$$

which also equals $E(n)R_1^n$ or just $E(n)$ since $R_1 = 1$.

Cases 3–5 in Table 2 were generated using this approach and various combinations of complex stages yielding up to five products. In Case 5, for example, $E(2) = E(3) = E(4) = E(5) = 1$. For $n = 15$, Case 1 in Table 2 includes <1% of those structurally distinct sequences that result from Case 5.

Most of the synthesis literature focuses on networks using a single distillation tower to produce only two products (only design (a) in Figure 2 is allowed). Table 2 shows, however, that this approach only considers a very small fraction of the design space. Any claim of optimality from such a limited viewpoint is purely conjecture at best.

Example

Consider a network producing five final products (A–E) from a single feed. Suppose that 2-2P, 4-3P, 6-4P, and 10-5P stage designs are allowed. If the products are separated only by distillation and species A is the most volatile with species E being the least volatile, then:

$$S_1 = \{A, B, C, D, E\} \tag{19}$$

$$S_2 = \{AB, BC, CD, DE\} \tag{20}$$

$$S_3 = \{ABC, BCD, CDE\} \tag{21}$$

$$S_4 = \{ABCD, BCDE\} \tag{22}$$

$$S_5 = \{ABCDE\} \tag{23}$$

and $5(6)/2 = 15$ stream rates and compositions must be defined to provide a species allocation database to solve the synthesis problem.

From the problem statement, $E(2) = 2, E(3) = 4, E(4) = 6$, and $E(5) = 10$. From Eq. 13:

$$\begin{aligned} D_5 &= \frac{2}{1!}(5-1) + \frac{4}{2!}(5-1)(5-2) \\ &\quad + \frac{6}{3!}(5-1)(5-2)(5-3) \\ &\quad + \frac{10}{4!}(5-1)(5-2)(5-3)(5-4) \\ &= 2(4) + 4(6) + 6(4) + 10(1) \\ &= 66 \end{aligned} \quad (24)$$

Similarly, $D_2 = 8$, $D_3 = 32$, and $D_4 = 56$. Hence, the total number of stages that must be evaluated to synthesize all permitted sequences is:

$$\begin{aligned} D(S) &= (5-1)\,D_2 + (5-2)\,D_3 + (5-3)\,D_4 + (5-4)\,D_5 \\ &= 306 \end{aligned} \quad (25)$$

The recursive boundary conditions are $\{R_1, R_2\} = \{1, 2\}$. The total number of networks that can be synthesized from this database is computed from:

$$\begin{aligned} T_3 &= E(2)[R_1R_2 + R_2R_1] + E(3)[R_1R_1R_1] \\ &= 2[2+2] + 4[1] = 12 \end{aligned} \quad (26)$$

$$\begin{aligned} T_4 &= E(2)[R_1R_3 + R_2R_2 + R_3R_1] \\ &\quad + E(3)[R_1R_1R_2 + R_1R_2R_1 + R_2R_1R_1] + E(4)[R_1R_1R_1R_1] \\ &= 2[12+4+12] + 4[2+2+2] + 6[1] = 86 \end{aligned} \quad (27)$$

$$\begin{aligned} T_5 &= E(2)[R_1R_4 + R_2R_3 + R_3R_2 + R_4R_1] \\ &\quad + E(3)[R_1R_1R_3 + R_1R_2R_2 + R_1R_3R_1 + R_2R_1R_2 \\ &\quad + R_2R_2R_1 + R_3R_1R_1] \\ &\quad + E(4)[R_1R_1R_1R_2 + R_1R_1R_2R_1 + R_1R_2R_1R_1 \\ &\quad + R_2R_1R_1R_1] + E(5)[R_1R_1R_1R_1R_1] \\ &= 2[86 + 24 + 24 + 86] \\ &\quad + 4[12 + 4 + 12 + 4 + 4 + 12] \\ &\quad + 6[2 + 2 + 2 + 2] + 10[1] \\ &= 690 \end{aligned} \quad (28)$$

Thus, the evaluation of 306 stages enables one to synthesize 690 networks in this case.

Notation

$A_{i,k}$ = kth stream member in the species allocation partition set S_i; each member in S_i contains i final product streams

D_n = design space for a single stage producing n products

$D(S)$ = total number of stages that must be suboptimized to synthesize all possible networks defined by S

$E(n)$ = number of allowed design types yielding n product streams from a single DP stage

$f_k(S_k)$ = minimum cost of partitioning from stream set p_k to set p_{k-1}

$g_k(p_k, q_k)$ = gain (profit) at stage k

$h(p_0, q)$ = material and energy balance constraints for the train

$\ell_k(p_k, p_{k-1}, q_k)$ = loss (or cost) function for stage k

n = number of final products (actual or pseudo) generated by a separation network

p_i = species allocation vector for the ith network

p_k = input function (feed vector) to stage k

p_{k-1} = total output function (products) from stage k

p_0 = set of final products

p_{n-l} = original feed for a single-feed problem or the composite feed in a multiple-feed problem

P = feasible region for p

q_i^* = optimal state vector for the ith network

q_k = decision vector at stage k

Q = feasible region for q

R_j = recursion variable for estimating T_j

S_j = partition set of streams containing j final products

T_n = number of alternative trains producing n final products

Literature Cited

Aris, R., *The Optimal Design of Chemical Reactors, a Study in Dynamic Programming,* Academic Press, New York (1961).

Bellman, R. E., *Dynamic Programming,* Princeton Univ. Press, Princeton, NJ (1957).

Dreyfus, S. E. and A. M. Law, *The Art and Theory of Dynamic Programming,* Academic Press, New York (1977).

Heaven, D. L., "Optimum Sequencing of Distillation Columns in Multicomponent Fractionation," MS Thesis, Chemical Engineering Department, Univ. of California, Berkely (1969).

Hendry, J. E., "Computer Aided Synthesis of Optimal Multicomponent Separation Sequences," PhD Thesis, Chemical Engineering Department, Univ. of Wisconsin, Madison, WI (1977).

Hendry, J. E., and R. R. Hughes, "Generating Separation Process Flowsheets, *Chem. Eng. Prog.,* **68,** 69 (1972).

Minderman, Jr., P. A., and D. W. Tedder, "Comparisons of Distillation Networks: Extensively State Optimized vs. Extensively Energy Integrated," *AIChE Symp. Ser.,* **78**(214), 69 (1982).

Shoaei, M., "Process Synthesis of Optimal and Near Optimal Distillation Networks," PhD Thesis, School of Chemical Engineering, Georgia Institute of Technology, Atlanta (1987).

Shoaei, M., S. S. Lorensen, and D. W. Tedder, "A Tree Search Methodology Based on Sequential Dynamic Programming and Set Analysis," *Comp. Chem. Eng.,* **11**(5), 469 (1987).

Shoaei, M., and J. T. Sommerfeld, "Catalan Numbers in Process Synthesis," *AIChE J.,* **32**(11), 1931 (1986).

Tedder, D. W., "The Heuristic Synthesis of Topology of Optimal Distillation Networks," PhD Thesis, Chemical Engineering Department, Univ. of Wisconsin, Madison (1975).

Tedder, D. W., and D. F. Rudd, "Parametric Studies in Industrial Distillation: I. Design Comparisons," *AIChE J.,* **24**(2), 303 (1978).

Thompson, R. W., and C. J. King, "Systematic Synthesis of Separation Schemes," *AIChE J.,* **18,** 941 (1972).

Manuscript received Jan. 24, 1989, and revision received Jan. 26, 1990.