



Optimal Design of Offshore Natural-Gas Pipeline Systems

B. Rothfarb; H. Frank; D. M. Rosenbaum; K. Steiglitz; D. J. Kleitman

Operations Research, Vol. 18, No. 6. (Nov. - Dec., 1970), pp. 992-1020.

Stable URL:

<http://links.jstor.org/sici?sici=0030-364X%28197011%2F12%2918%3A6%3C992%3AODOONP%3E2.0.CO%3B2-O>

Operations Research is currently published by INFORMS.

Your use of the JSTOR archive indicates your acceptance of JSTOR's Terms and Conditions of Use, available at <http://www.jstor.org/about/terms.html>. JSTOR's Terms and Conditions of Use provides, in part, that unless you have obtained prior permission, you may not download an entire issue of a journal or multiple copies of articles, and you may use content in the JSTOR archive only for your personal, non-commercial use.

Please contact the publisher regarding any further use of this work. Publisher contact information may be obtained at <http://www.jstor.org/journals/informs.html>.

Each copy of any part of a JSTOR transmission must contain the same copyright notice that appears on the screen or printed page of such transmission.

The JSTOR Archive is a trusted digital repository providing for long-term preservation and access to leading academic journals and scholarly literature from around the world. The Archive is supported by libraries, scholarly societies, publishers, and foundations. It is an initiative of JSTOR, a not-for-profit organization with a mission to help the scholarly community take advantage of advances in technology. For more information regarding JSTOR, please contact support@jstor.org.

OPTIMAL DESIGN OF OFFSHORE NATURAL-GAS PIPELINE SYSTEMS

B. Rothfarb, H. Frank, and D. M. Rosenbaum

Network Analysis Corporation, Glen Cove, New York

K. Steiglitz

Princeton University, Princeton, New Jersey

D. J. Kleitman

Massachusetts Institute of Technology, Cambridge, Massachusetts

(Received January 28, 1969)

The exploitation of offshore natural gas reserves involves several phases, including production from reservoirs, separation of byproducts, and transportation to markets. The gas, which may originate as far as 100 miles from land, must be transported through pipelines to onshore delivery points. This paper develops techniques for solving the following problems: (1) selection of pipe diameters in a specified pipeline network to minimize the sum of investment and operation costs; (2) selection of minimum-cost network structures, given gas-field location and flow requirements; (3) optimal expansion of existing pipeline networks to include newly discovered gas fields. The techniques incorporate procedures for globally optimizing pipeline diameters for fixed tree structures and heuristic procedures for generating low-cost structures.

THE PRODUCTION and distribution of natural gas is an important factor in our domestic economy. Over 38 million homes and industries, serviced by an 800,000 mile pipeline network, depend on gas for heating and other essential services. Enormous gas reserves located in the Gulf of Mexico and other offshore regions are presently undergoing development. The cost of installing underwater pipelines is a significant factor in the economics of offshore natural-gas exploitation. Proper design of these networks can facilitate the economical integration of newly discovered fields into the gathering system and result in lower prices to the consumer.

This study investigates three problems in the design and expansion of pipeline networks: (a) the selection of optimal diameters in a given pipeline network, (b) the design of an optimal pipeline system, given gas-field locations and delivery requirements, and (c) the optimal expansion of an existing pipeline network.

Natural gases contain both valuable by-products and impurities that

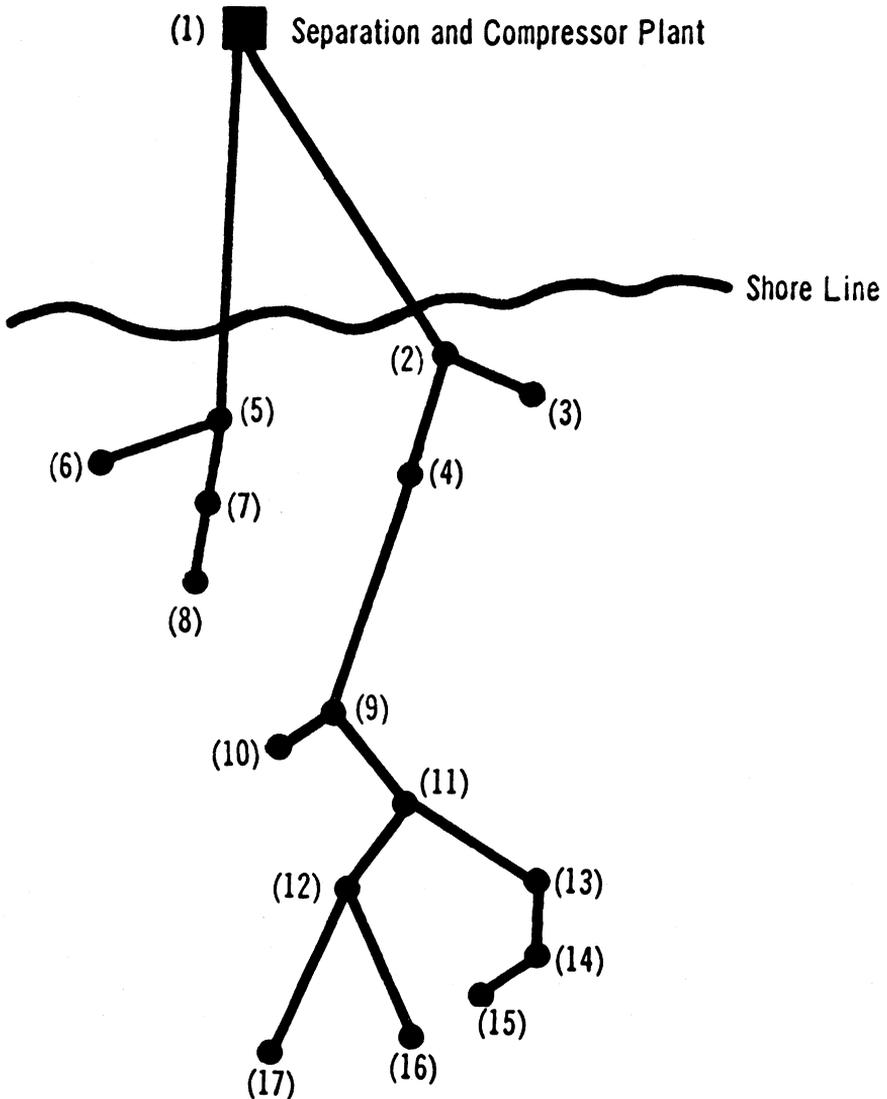


Fig. 1. Offshore pipeline network. Points labeled (2)–(17) are gas fields. Lines connecting these points are pipes.

must be removed before being transported appreciable distances. The necessary separation for gas produced from onshore fields is usually performed directly at the well. For offshore wells, however, this is too costly. Therefore, the gas must be transported through pipelines to be processed

at plants on land. Once processed, the gas must often be compressed so that it can be efficiently transported to its market. Figure 1 shows a typical offshore pipeline network. The essential elements of such a network are gas fields, pipelines, platforms, junctions (places where different pipelines are joined), separation plants, and compressors. The important characteristics of these elements are described below.

(a) It is assumed that these average daily production rates for the gas fields are specified.

(b) Whenever a gas field is tapped or several pipelines connected at a junction, a platform is usually installed to support the necessary construction. The cost of these platforms typically varies between 0.5 and 1.5 million dollars. A drilling platform at each gas field must be constructed

TABLE I
APPROXIMATE GULF-OF-MEXICO PIPELINE COSTS PER MILE

Nominal external pipe diameter, inches	Onshore-marsh	Water depth, feet	
		0-90	90-200
30	\$260,000	\$295,000	\$325,000
26	195,000	205,000	235,000
24		188,000	215,000
20		153,000	175,000
16		131,000	150,000
12 $\frac{3}{4}$		83,000	95,000
10 $\frac{3}{4}$		65,000	75,000

Data supplied by the Federal Power Commission.

in order to drill the wells. Once drilling has been completed, these platforms become available for pipeline junctions. Hence, for reasons of economy, pipelines will usually be connected between producer-constructed drilling platforms or from a producer-constructed platform to onshore facilities. (For further discussion about this restriction, see the section entitled "Extensions.")

(c) A pipeline is characterized by its internal and external diameters, type of material and fabrication, and its length. The flow of gas through a pipeline is a function of the internal diameter, the length, the pressures at the endpoints, the elevation profile of the pipeline, the flowing temperature, and the physical characteristics of the gas. A pipeline may carry both gas and liquid hydrocarbons. Pipelines that accommodate such liquid flow are called two-phase lines; the ones that carry no appreciable quantities of liquids are called single-phase lines. In this study, the stand-

ard 'Panhandle' equation is used to describe the steady-state (time-independent) flow of gas in single-phase pipelines^{11, 21}:

$$Q = K[(P_1^2 - P_2^2)/L]^{0.5394} D^{2.6182}, \quad (1)$$

where

- Q = flowing volume, cubic ft/day,
- P_1 = input pressure, psia,
- P_2 = output pressure, psia,
- D = internal pipe diameter, inches,
- L = length of pipe, miles,
- K = constant.

The design techniques discussed in this paper are independent of the exact form of the flow equation. For example, all exponents may be assumed variable. The essential characteristic of the flow equation is that, if P_1 and P_2 are the only unspecified variables, P_1 must be expressible as $P_1 = F(P_2)$, where F is invertible.

(d) Pipelines are available in a finite selection of standard diameters. For a particular diameter, the costs depend on the depth of the water at which the pipe is to be installed. The diameters of the pipes to be considered and typical costs per mile are given in Table I. These costs must be amortized over some time period to convert total capital cost to an annual cost. Any amortization scheme can be used in the design method.

(e) The key factor in the compressor model is the required horsepower as a function of the amount of gas that flows through the compressor and the input and output pressures. Any method of determining compressor cost, given input flow, input pressure and desired output pressure, is acceptable.

(f) The major pressure constraints on the system are: (i) The maximum allowable pressure in a pipeline cannot exceed P_{\max} , which is a specified constant; (ii) gas must be delivered into the onshore network at a pressure greater than or equal to P_{\min} , which is another specified constant with $P_{\min} < P_{\max}$; and (iii) the pressure available at each well is at least P_{\max} .

TREE ANALYSIS AND OPTIMIZATION

THE KEY TO the solution of the design problems is to develop rules to eliminate uneconomical diameter combinations and networks without enumeration. The approach taken consists of several stages. First, optimal pipe diameters for fixed network structures, to minimize total investment and operating cost, are selected. Next, pipeline configurations that yield economical systems are generated. These techniques are then used to solve the problem of integrating a set of newly discovered fields

optimally into an existing pipeline network. As the basic feature of our study, we consider the analysis and optimization of the networks that contain the fewest number of pipelines that can deliver gas from the fields to the separation plants. Such structures, called *trees*, if carefully selected, are initially economical and are easy to expand as new fields are added.

Any gas field or delivery point is called a node and any line between two nodes is called a branch. Given a tree containing n nodes, and therefore $n - 1$ branches, with seven diameter choices for each branch, there are

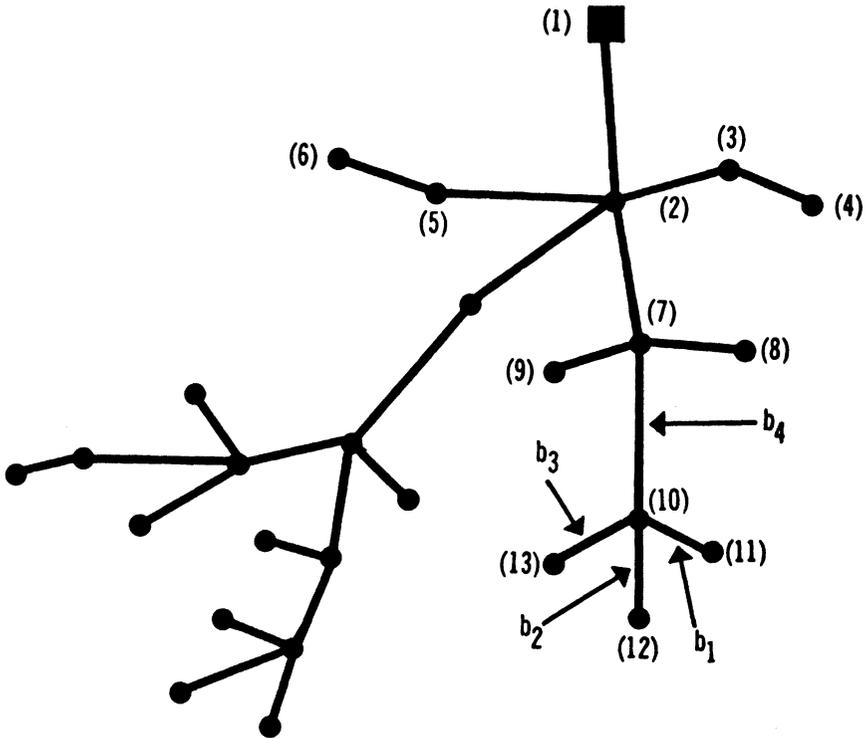


Fig. 2. Tree used to demonstrate parallel- and serial-merge techniques.

7^{n-1} possible diameter assignments. For example, for the 25-node tree shown in Fig. 2, there are $7^{24} \approx 2 \times 10^{20}$ diameter assignments. Among these possible choices of diameters, we wish to find the one that leads to the least expensive gathering network.

The flows in each branch of a tree are determined when the flows out of each gas field are known. Thus, specifying a diameter for a branch also specifies a difference of the squares of the pressures across that branch [see equation (1)]. Therefore, given a diameter assignment for all the

branches, the node at which the pressure is greatest can be determined. Since there is a maximum allowable node pressure P_{\max} , the pressure at the delivery point must be set low enough that the necessary pressure-squared differences can be maintained without exceeding the pressure limit anywhere in the network. The path in the tree from the delivery point to the node of greatest pressure will be called the *critical path*. The sum of the pressure-squared differences along the critical path determines the delivery-point pressure, and hence the cost of compression.

Choosing diameters for some of the branches and leaving the diameters for the remaining branches unspecified will be called a *partial assignment*. Given the very large number of possible diameter assignments, methods must be developed that recognize partial assignments that cannot be in the optimal assignment. It is imperative to eliminate these 'foolish' partial assignments early in the processing to keep the number of candidate partial assignments tractable. If we are to find a globally optimal assignment, we cannot discard any partial assignment that might possibly be part of the optimal assignment.

Let each branch have a vector called PCOST such that the i th component is the pipe cost associated with the i th smallest diameter choice. Similarly, let each branch have a vector called PSQ such that the i th component is the pressure-squared difference across the branch arising from a choice of the i th smallest diameter size for that branch. The values of the elements of PCOST are in increasing order and those of PSQ in decreasing order. The two vectors taken together will be called a *branch list*. Two techniques have been developed that, when used together on a given tree, can efficiently process these branch lists to obtain the optimal diameter assignment.

The first technique we shall consider is called the *parallel merge*. It can be used on any set of branches that directly connect nodes of degree one to a common node. (The *degree* of a node is the number of branches that are incident to the node.)

We will use branches b_1 , b_2 , and b_3 from the tree of Fig. 2 to illustrate the procedure. Suppose the lists for these branches are as follows:

b_1 list:	PSQ	(120, 111, 92, 66, 54, 40, 31)
	PCOST	(13, 17, 23, 29, 36, 45, 58)
b_2 list:	PSQ	(150, 139, 118, 87, 75, 70, 67)
	PCOST	(6, 9, 14, 21, 30, 40, 56)
b_3 list:	PSQ	(94, 86, 80, 61, 55, 48, 32)
	PCOST	(8, 12, 18, 26, 34, 43, 57)

A testing block is set up as shown in Table II. Each branch list being merged has a column in the testing block as indicated. If the index in a column is set to i , then

TABLE II
THE FIRST TESTING BLOCK FOR THE EXAMPLE

	b_1	b_2	b_3
PSQ	120	150	94
PCOST	13	6	8
INDEX	1	1	1

the PSQ and PCOST entries in that column are the i th components of the list. Initially, the indices are set to 1 and the testing block is as shown above.

The procedure locates the largest entry in the PSQ row of the testing block. In our example, this occurs in the b_2 column and the entry is shown in bold-face type. If the smallest pipe diameter is chosen for b_2 , then b_1 and b_3 can *never* be on the critical path. Thus choosing other than the minimum diameters for b_1 and b_3 , when b_2 has the minimum diameter, will increase the total cost of the pipes, but cannot reduce the compression cost. If an optimal assignment has b_2 at its minimum diameter, then b_1 and b_3 must also have minimum diameters.

We now enter the bold-face PSQ entry and the sum of the PCOST entries of the testing block on a new list. This entry on the new list corresponds to a partial assignment of minimum diameters to b_1 , b_2 , and b_3 :

New list: PSQ (150)
PCOST (27).

Since no better choice of diameters for b_1 and b_3 is possible with b_2 at this diameter, we promote the index in the b_2 column of the testing blocks, which yields Table III.

In the updated testing block, the new maximum PSQ entry is still in the b_2 column. This means that, if b_2 has the second smallest diameter, it still will not pay to have b_1 or b_3 at any diameters other than the smallest. We make a second entry in the new list as before to give:

New list: PSQ (150, 139),
PCOST (27, 30).

TABLE III
THE SECOND TESTING BLOCK FOR THE EXAMPLE

	b_1	b_2	b_3
PSQ	120	139	94
PCOST	13	9	8
INDEX	1	2	1

This new entry represents a partial assignment of the second smallest diameter to b_2 and the smallest diameter to b_1 and b_3 .

We again promote the index in the b_2 column to yield the updated testing block in Table IV. Now the largest entry in the PSQ row is in the b_1 column. If an optimal assignment contains b_1 at its smallest diameter, it cannot contain b_2 at a larger diameter than the third smallest, or b_1 at a diameter other than the smallest. This yields a new entry in the updating of the new list:

$$\text{New list: } \begin{matrix} \text{PSQ} & (150, 139, 120), \\ \text{PCOST} & (27, 30, 35), \end{matrix}$$

and we promote the index in the b_1 column.

The process terminates when the largest entry of the PSQ row of the testing block occurs in a column whose index has been promoted to 7. Further promotion of the other indices would correspond to partial assignments of greater pipe cost and no possible savings in compression costs. The remainder of the sequence of testing

TABLE IV
THE THIRD TESTING BLOCK FOR THE EXAMPLE

	b_1	b_2	b_3
PSQ	120	118	94
PCOST	13	14	8
INDEX	1	3	1

blocks and the complete resulting new list are shown in Table V. This completes a parallel merge of the b_1 , b_2 , and b_3 lists.

Each entry on the final new list (*see* Table V) represents an assignment of diameters to the branches b_1 , b_2 , and b_3 . Furthermore, no other partial assignments for these branches need be considered. Note that the number of possible partial assignments for these three branches is $7^3 = 343$. However, the parallel-merge techniques will produce a new list with at most 19 components, one from the original testing block and one each from the testing blocks resulting from a maximum of 18 index promotions. The minimum number of components on a new list is 7.

The parallel merge produces a new list whose entries correspond to partial assignments that are candidates for inclusion in the optimal assignment. The new list can be viewed as the list of an equivalent branch that replaces the branches whose lists were merged. This equivalent branch can be thought of as a branch connected between node 10 and a node consisting of a combination of nodes 11, 12, and 13. (Hence, the name parallel merge.) Note that the components of PCOST and PSQ are respectively in increasing and decreasing order so that no reordering of the list is required.

It now becomes desirable to combine somehow the list of the equivalent branch with the list of b_4 to create a new equivalent branch list for b_1 , b_2 , b_3 , and b_4 . Again we wish to retain as few partial assignments as possible without eliminating any partial assignments that can possibly be in the optimal assignment. A technique for accomplishing this, which we call the *serial merge*, is described next.

TABLE V
TESTING BLOCKS SHOWING COMPLETION OF THE PARALLEL-MERGE PROCESS FOR THE EXAMPLE

	b_1	b_2	b_3									
PSQ	111	118	94	111	87	94	92	87	94	92	87	86
PCOST	17	14	8	17	21	8	23	21	8	23	21	12
INDEX	2	3	1	2	4	1	3	4	1	3	4	2
PSQ	66	87	86	66	75	86	66	75	80	66	75	61
PCOST	29	21	12	29	30	12	29	30	18	29	30	26
INDEX	4	4	2	4	5	2	4	5	3	4	5	4
PSQ	66	70	61	66	67	61						
PCOST	29	40	26	29	56	26						
INDEX	4	6	4	4	7	4						

The final new list—equivalent branch list for b_1 , b_2 , and b_3 :

PSQ (150, 139, 120, 118, 111, 94, 92, 87, 86, 80, 75, 70, 67) PCOST (27, 30, 35, 39, 46, 52, 56, 62, 71, 77, 85, 95, 111)

The serial merge can be used on any two branches incident to a common node of degree two if at least one of the two branches is also incident to a node of degree one. The branches may be actual branches or equivalent branches.

Thus, for example, the serial merge can merge the list of the equivalent branch just obtained, which corresponds to the actual branches b_1 , b_2 , and b_3 , with the b_4 list. Both branches are incident to node 10, which is of degree two. As previously discussed, the equivalent branch can be considered to be incident to a node consisting of a combination of nodes 11, 12, and 13, and this node is of degree one.

TABLE VI
TESTING BLOCK USED IN ILLUSTRATING THE SERIAL MERGE

	1	2	3	4	5	6	7
PSQ	283	274	254	228	215	201	192
PCOST	33	37	42	50	60	70	86
INDEX	1	1	1	1	1	1	1

We will use the equivalent branch list obtained above and the following b_4 list to illustrate the serial merge:

b_4 list: PSQ (133, 124, 104, 78, 65, 51, 42),
PCOST (6, 10, 15, 23, 33, 43, 59).

We set up a testing block with 7 columns as shown in Table VI. The i th column corresponds to the i th smallest-diameter choice for b_4 and an index equal to j in a column corresponds to the j th partial assignment of b_1 , b_2 , and b_3 in the equivalent branch list. The PSQ and PCOST entries in a column are the corresponding pressure-squared difference and pipe cost that would result from such a partial assignment of b_1 , b_2 , b_3 , and b_4 .

Initially, the indices are all set to 1. The testing block given in Table VI therefore gives all the data for the partial assignments for every choice of diameter for b_4 with the partial assignment of b_1 , b_2 , and b_3 corresponding to the first component of the equivalent branch list. Thus, the PSQ entry in the i th column of the initial testing block is the sum of the i th PSQ component for b_4 and the first PSQ component on the equivalent branch list. Similarly, the i th column PCOST entry is the sum of the i th b_4 PCOST component and the first equivalent-branch PCOST component.

We now locate the maximum entry in the PSQ row of the testing block. Initially, this will always occur in the first column. The PSQ and PCOST entries of this column become candidate components in a new equivalent branch list. We then promote the index in the first column to yield the result in Table VII. The largest PSQ entry is now in the second column whose PSQ and PCOST entries become the second component in the candidate list. The updating of the testing block yields Table VIII. The current candidate list is now

TABLE VII

	1	2	3	4	5	6	7
PSQ	272	274	254	228	215	201	192
PCOST	36	37	42	50	60	70	86
INDEX	2	1	1	1	1	1	1

Candidate list: PSQ (283, 274, 272),
PCOST (33, 37, 36).

Each component on the candidate list corresponds to a partial assignment of b_1 , b_2 , b_3 , and b_4 . By our method of construction, the PSQ vector will have its components in nonincreasing order. Note, however, that the last PCOST component on the candidate list is smaller than the second PCOST component. The partial assignment corresponding to the third list component is always preferable to the partial assignment corresponding to the second component, since the former has a lower pipe cost and cannot result in a greater cost of compression. We therefore can eliminate the second component from further consideration. Our candidate list reduces to:

Candidate list: PSQ (283, 272),
PCOST (33, 36).

In general, when a new candidate is added to the list, we can eliminate all the candidates already on the list that have PCOST components that are not smaller

TABLE VIII

	1	2	3	4	5	6	7
PSQ	272	263	254	228	215	201	192
PCOST	36	40	42	50	60	70	86
INDEX	2	2	1	1	1	1	1

than the latest entry. Since after each change the PCOST vector components in the candidate list will be in increasing order, the updating of the candidate list is easy to implement.

As we proceed with the serial-merge technique, each of the 13 components of the equivalent branch list will form a candidate with each of the seven b_4 list components. Thus, a total of $7(13) = 91$ candidates must be processed. However, as we have seen, some of these candidates can be eliminated. For the example under consideration, there are 91 candidates generated. Of these 91 candidates, only 31 are retained, and these constitute an equivalent branch list for b_1 , b_2 , b_3 , and b_4 . In general, only a small fraction of the candidates in a serial merge will be retained. A greater percentage of the earlier and later candidates will generally be retained than those in the middle, so the power of the elimination procedure is not fully illustrated by the small example given.

Since the parallel- and serial-merge techniques can be applied to lists of both actual and equivalent branches, the entire tree can be processed to yield a single equivalent branch list. The cost of the diameter assignment corresponding to each entry on this final list can then be evaluated by summing its pipe cost (PCOST) and the cost of compression associated

with the pressure-squared difference (PSQ). The diameter assignment with the smallest cost is the optimal-diameter assignment.

Of critical importance is the size of the intermediate and final lists produced by the parallel and serial merge techniques. For examples with trees of 20 nodes, the maximum list size has never been greater than 1000 and appears to grow approximately linearly as a function of the number of nodes. Recall that the number of possible diameter assignments grows exponentially as a function of the number of nodes.

It takes about one second of computer time on a UNIVAC 1108 to determine the optimal diameter assignment for a 20-node tree. Thus, the techniques presented in this section provide a new and efficient tool for designing pipeline networks.

AUTOMATIC TREE GENERATION

IN THIS SECTION, methods for automatically generating trees are described. It is assumed throughout this section that for each tree the pipe diameters are chosen optimally by the optimization method discussed above. Two procedures are described. The first, called the *starting routine*, is a heuristic program for generating initial trees. The second, called Δ -*opt*, is an iterative optimization program that searches for and adopts local transformations that lower the cost. In conjunction with the starting routine, this provides a completely automatic design procedure.

The heuristic starting routine. A program was written that incorporated some of the heuristic rules garnered from trial-and-error experience. While this program can be refined, to do so appears unnecessary in light of the efficiency of the local optimization program described later in this section. A description of the program and some results are presented below.

It became clear at the outset that the pipes connected to the separation plants play an especially significant role in the development of a tree. We shall call these pipes *arms* and the pipes connected to each arm its *subtree*. It is assumed that the number and location of the arms is provided to the program by the user. In most cases there are only a few reasonable choices and all can be tried.

In addition to the prespecification of arms, two important guidelines in the design of trees emerge:

- Efficient trees have low total pipe mileage.
- Efficient trees have nearly equal flow in their arms.

If the consideration of only low total pipe length were important, one could construct a 'minimal length' tree, with the restriction that it contain the specified arms, as follows: at each stage connect the unconnected node

whose distance to the closest connected node is smallest.^[3] In order to take the second guideline into account, the unconnected node closest to the subtree with lowest total flow is sometimes added. This is done with a certain probability P , which is set as a parameter of the program. The introduction of this random element in the growth of the tree results in a

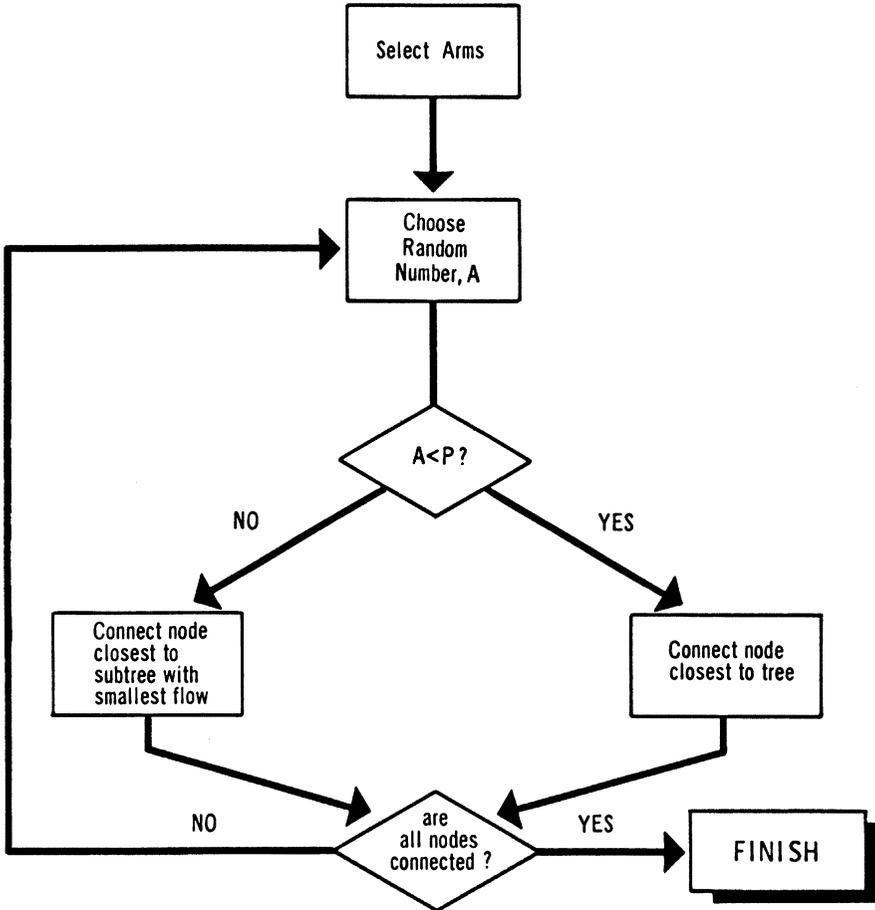


Fig. 3. Block diagram of the heuristic tree-generation program.

distribution of answers; the one with the lowest total cost can be chosen. A block diagram of the program is shown in Fig. 3.

To illustrate the operation of the algorithm, 19 offshore fields from the Gulf Coast area under study were chosen as an example. Ten runs were made with two arms and $P=0.8$. Figure 4(a) shows the best solution of the ten, with a cost

of \$198,572,000 over a 20-year period. Figure 4(b) shows another solution of the ten, with a cost of \$210,100,000. The second of these is closer to the minimal-length tree, but has a cost significantly greater.

Figure 5(a) shows the best of ten runs made with three arms and $P=0.8$, with

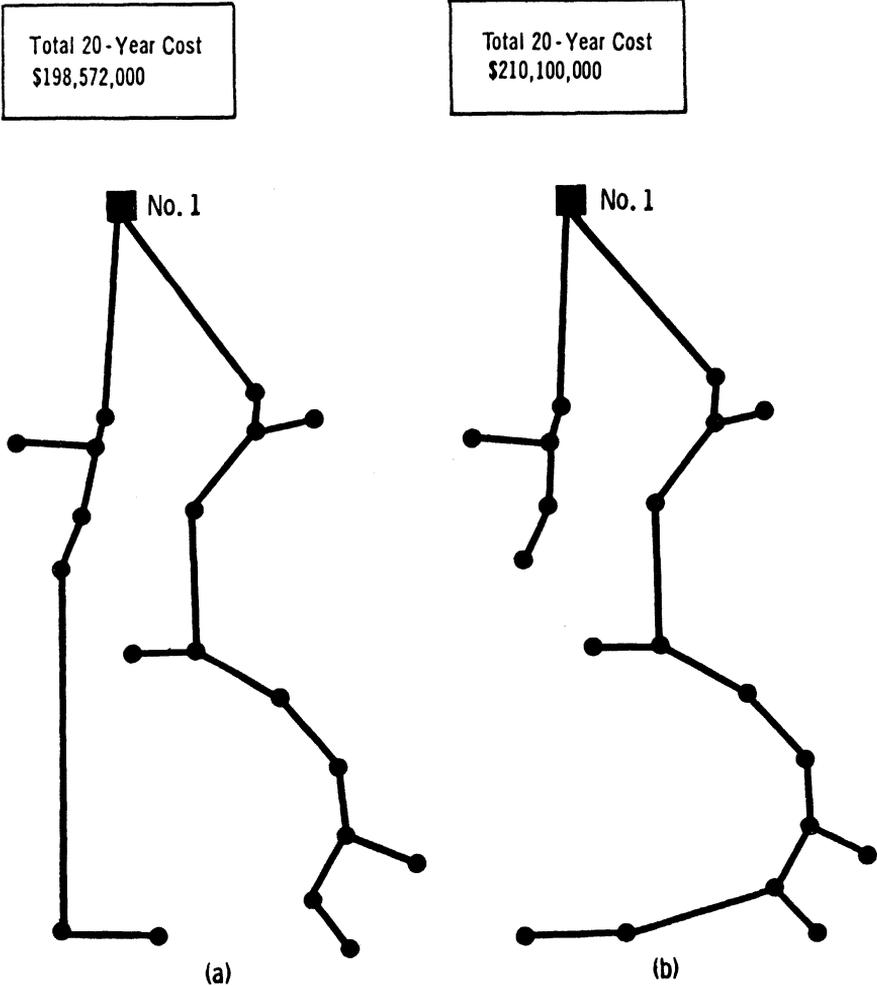


Fig. 4. Two-arm network examples of starting routine.

a 20-year cost of \$199,538,000. Possibilities for local improvements of this solution are quite clear from the figure. The lowest-cost tree for this 19-field example, which was obtained by human inspection and trial and error, is shown in Figure 5(b), with a 20-year cost of \$192,588,000. An automatic tree-generating program as described above can produce reasonable first approximations to good structures.

When combined with a local optimization program, such as the one to be described next, extremely economical networks are obtained.

The Δ -optimization routine. We now describe an iterative optimization technique that produces locally optimal trees with respect to a given

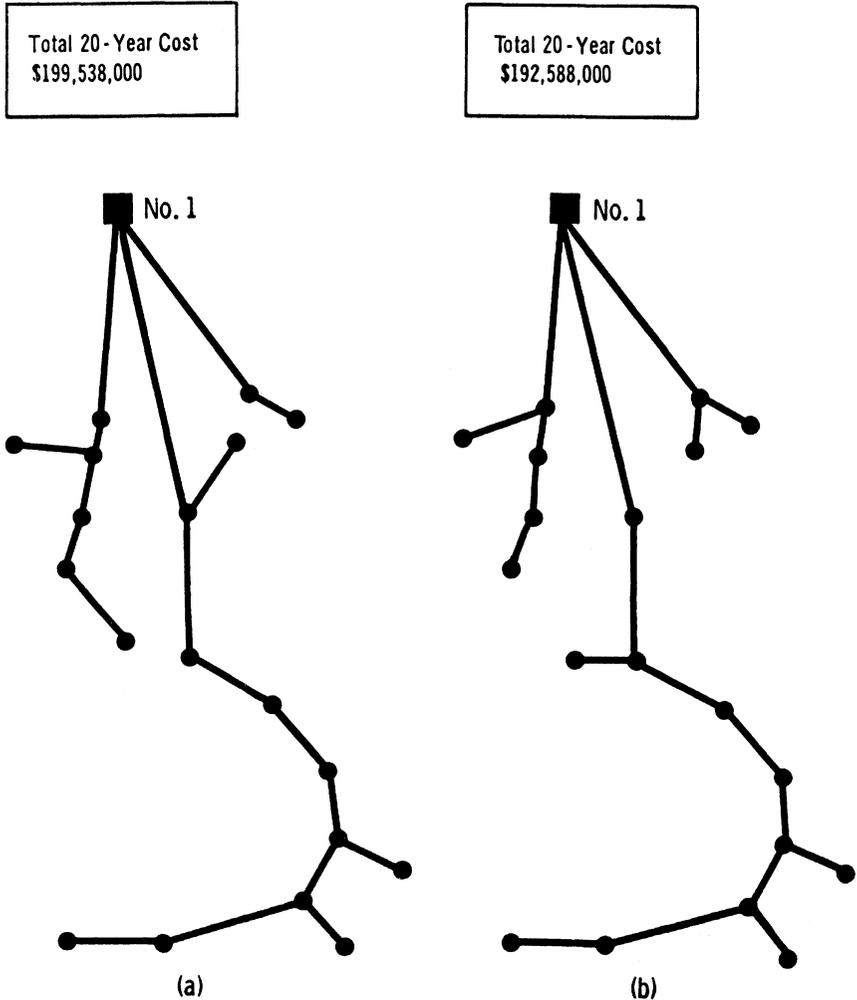


Fig. 5. Three-arm network examples of starting routine.

transformation called a Δ -change. Two characteristics of this transformation make it extremely suitable for iterative optimization: first, it attempts changes that have a high probability of improving cost; second, the number of possible changes is small enough so that all of them can be examined for a given tree of 20 or 30 nodes.

A Δ -change is a special kind of elementary tree transformation. Specifically, given a tree, choose a node n_1 . Find the node $n_{1,1}$ closest to n_1 but not adjacent to it. Two nodes are said to be adjacent to each other if there is a branch directly connecting them. n is the number of nodes in the network. Add the branch from n_1 to $n_{1,1}$ and determine the circuit formed. This circuit can be found easily using a labeling algorithm.^[3] Call the branches in this circuit b_1, \dots, b_m , where b_1 is the added branch from n_1 to $n_{1,1}$. New trees can be formed by deleting in turn each of the branches b_2, \dots, b_m . The process of obtaining each new tree is said to be a Δ -change and is illustrated in Figures 6(a)–(c). This process can now be repeated by finding the nonadjacent node $n_{1,2}$ next closest to n_1 . In the program, the three nearest nonadjacent neighbors of each node are used to generate Δ -changes. This means that if there are an average of k branches in each circuit, on the order of $3kn$ trees must be evaluated to certify a local optimum. Restricting the choice to the three nearest nonadjacent neighbors keeps computation time small.

Since it is highly likely that a node will be connected to one of its nearest neighbors in the global optimum, Δ -changes will uncover an extremely large class of local improvements in a computation time that is not prohibitive.

Figure 7 shows a block diagram of the over-all design procedure that incorporates the heuristic starting procedure and, of course, the pipe-diameter optimization routine. Each node n_i is examined in turn, until a new tree with lower cost is obtained. This tree is used as a starting point for an additional application of the optimization routine. The procedure terminates when a tree is obtained for which no further favorable Δ -changes exist. Such a tree is called Δ -opt. It is possible that a Δ -opt tree will not be globally optimal. However, using the optimization procedure with several different starting points produces a best answer that has a high likelihood of global optimality. In any event, the designs produced are better than those generated by many trial-and-error human iterations using the diameter optimization program. Δ -opt trees represent a practical solution to the problem of choosing the structure for a pipeline network.

Figures 8(a)–(m) demonstrate the operation of the program for the gas fields in a section of the Gulf of Mexico. Figure 8(a) shows the heuristic starting solution obtained with one arm and $P=1$. This starting tree corresponds to a minimal-length tree; that is, the total length of pipe in the tree is as small as possible. Figures 8(b)–(m) show successive improvements found by the Δ -optimization program. On each figure, the two branches involved in a Δ -change are indicated by the words “NEW” (the branch to be added) and “OLD” (the branch to be removed). The pipe diameters are indicated by integers next to the branches ranging from 1 to 7. A 1 represents the smallest diameter, $10\frac{3}{4}$ inches, allowed

in Table I. The 2 represents the second smallest diameter, $12\frac{3}{4}$ inches, and, in general, the integer i represents the i th smallest diameter in Table I. It is not possible to guarantee the global optimality of the tree in Fig. 8(m). However, the

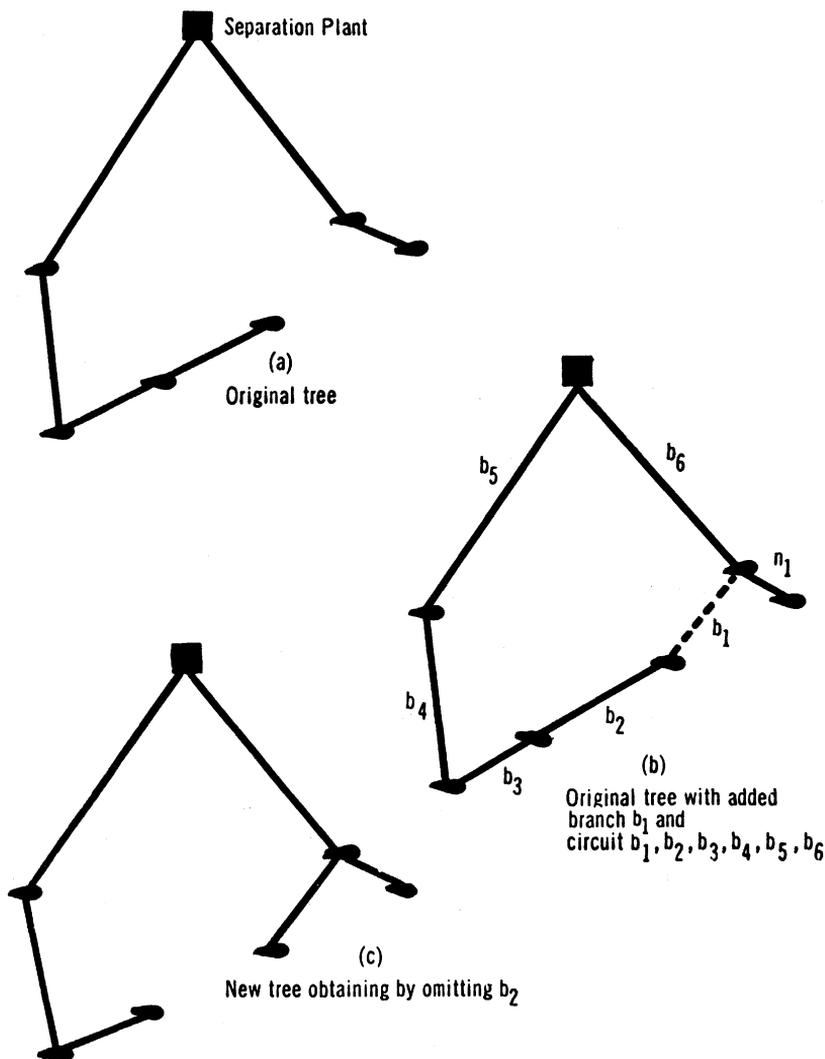


Fig. 6. Illustration of a Δ -change.

cost of this tree is lower than the cost of trees obtained by any other method. Some of the branch locations and diameters may be fixed during the optimization process. This is useful for adding new sources to an existing network. However, in this case it is usually possible to exhaust all ways of adding the new sources, provided that only a limited number of new branches are allowed.

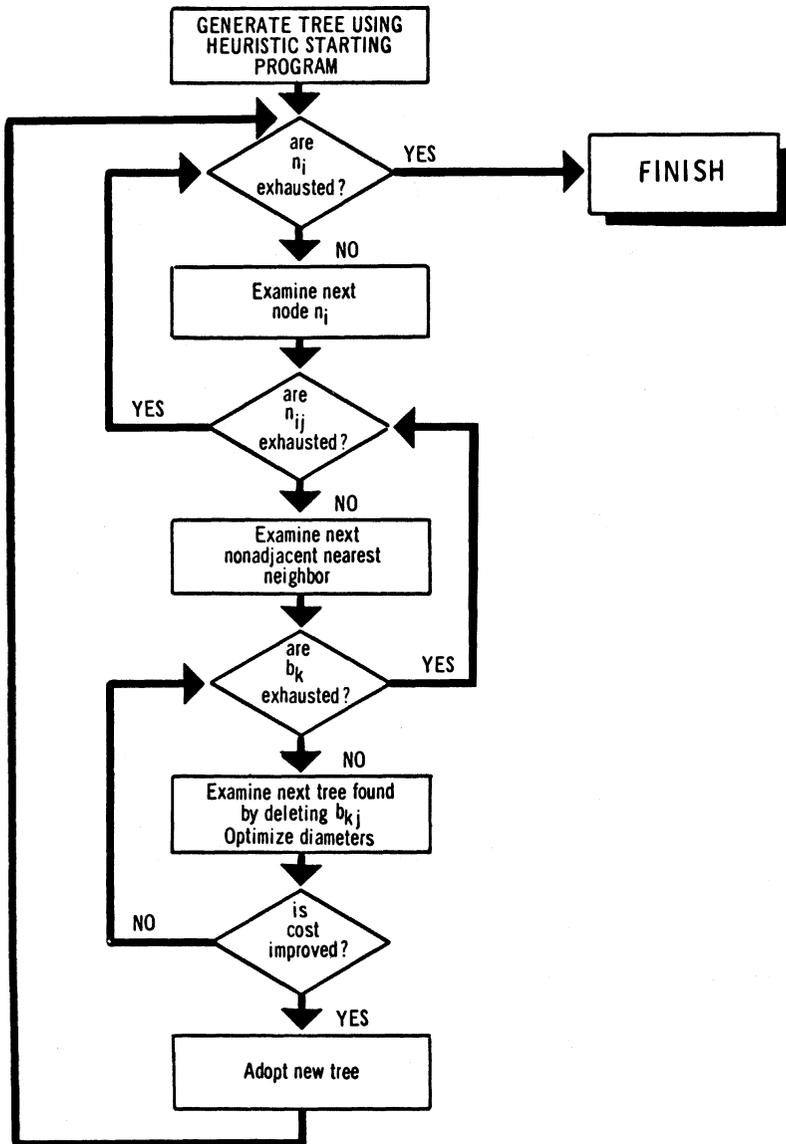


Fig. 7. Block diagram of the over-all design procedure.

EXTENSIONS

THE PRECEDING discussion of the pipeline-network design problem accounted for the key physical factors that govern gas flow and pipeline construction. A great number of variations in the model of the physical system can be included without changing the validity of the analytic

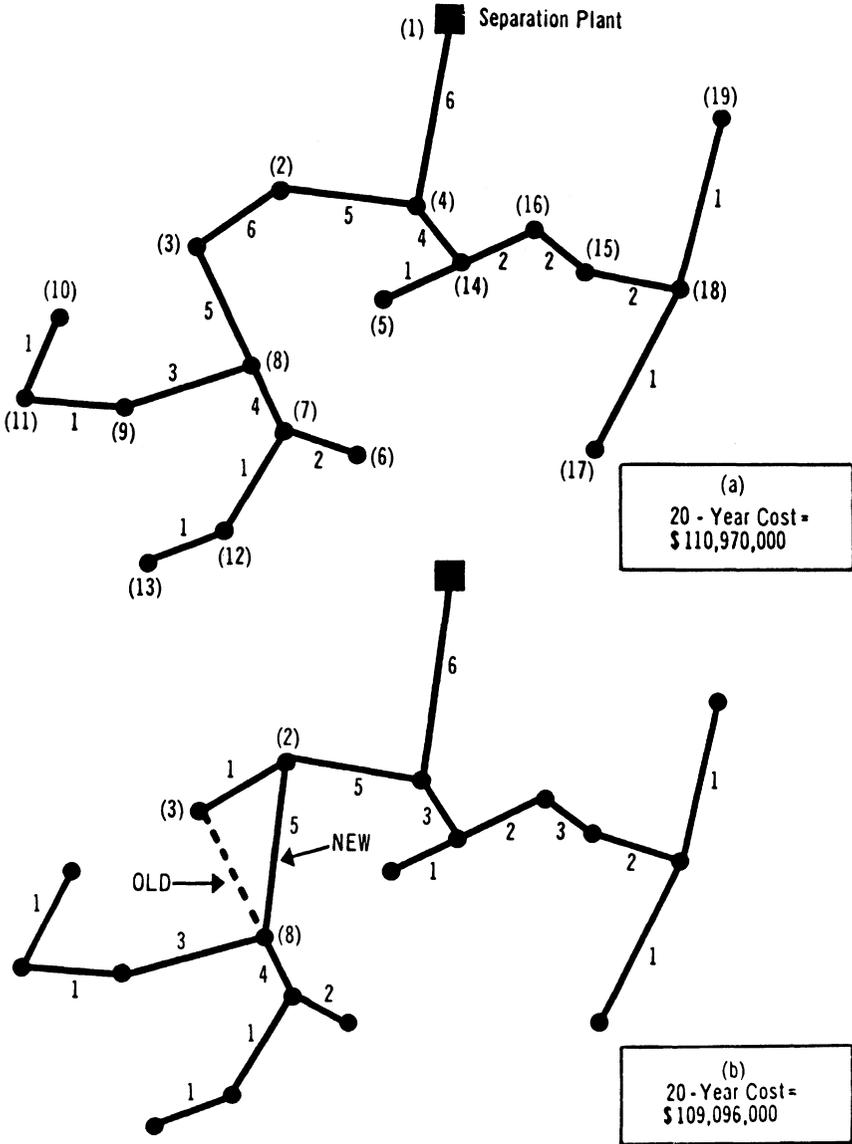


Fig. 8 (a-b)

Fig. 8. Example of the tree-optimization procedure. Part (a) shows the shortest tree with optimally assigned diameters. Parts (b)–(m) show improvements made by exchanging branches and reoptimizing diameters.

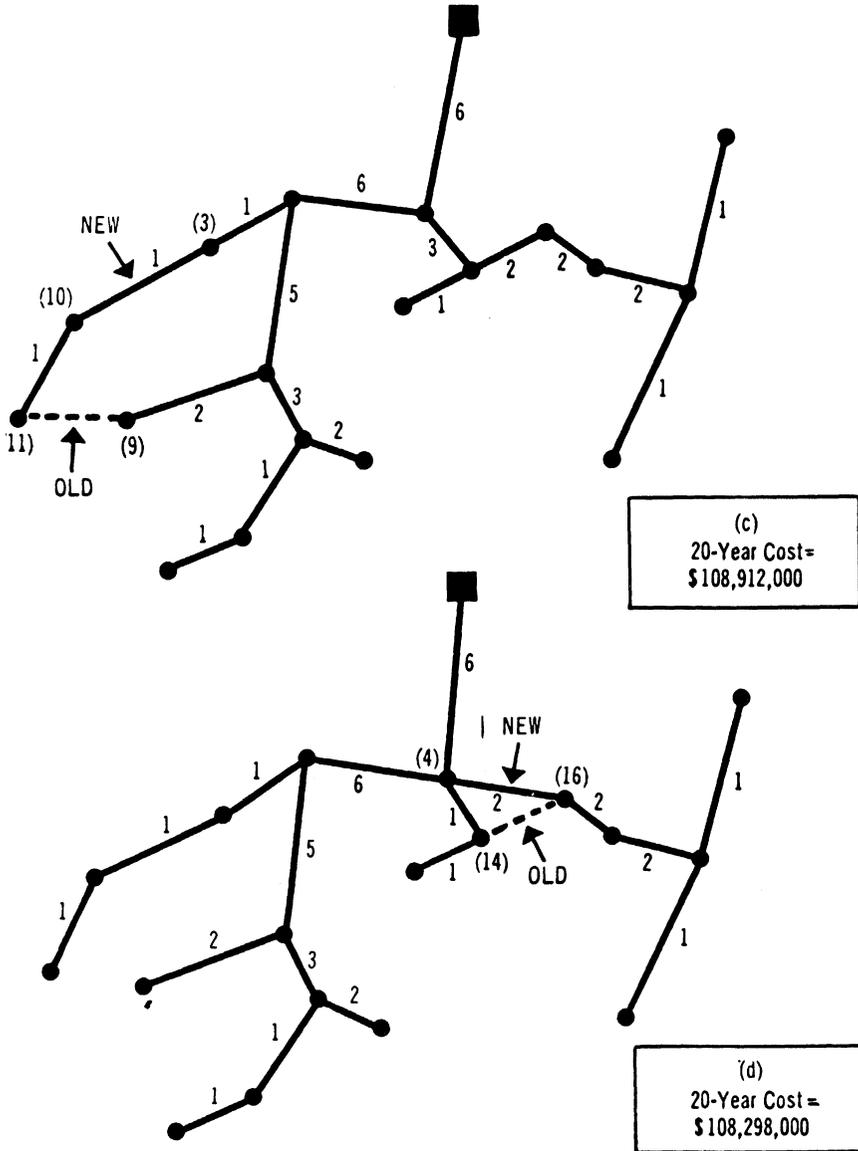


Fig. 8 (c-d)

techniques. Because of time limitations, these features were omitted from the present study. Among the modifications that are most easily included are:

The flow formula. The flow formula used in this study is given by equation (1), but the pipeline optimization methods do not depend on the

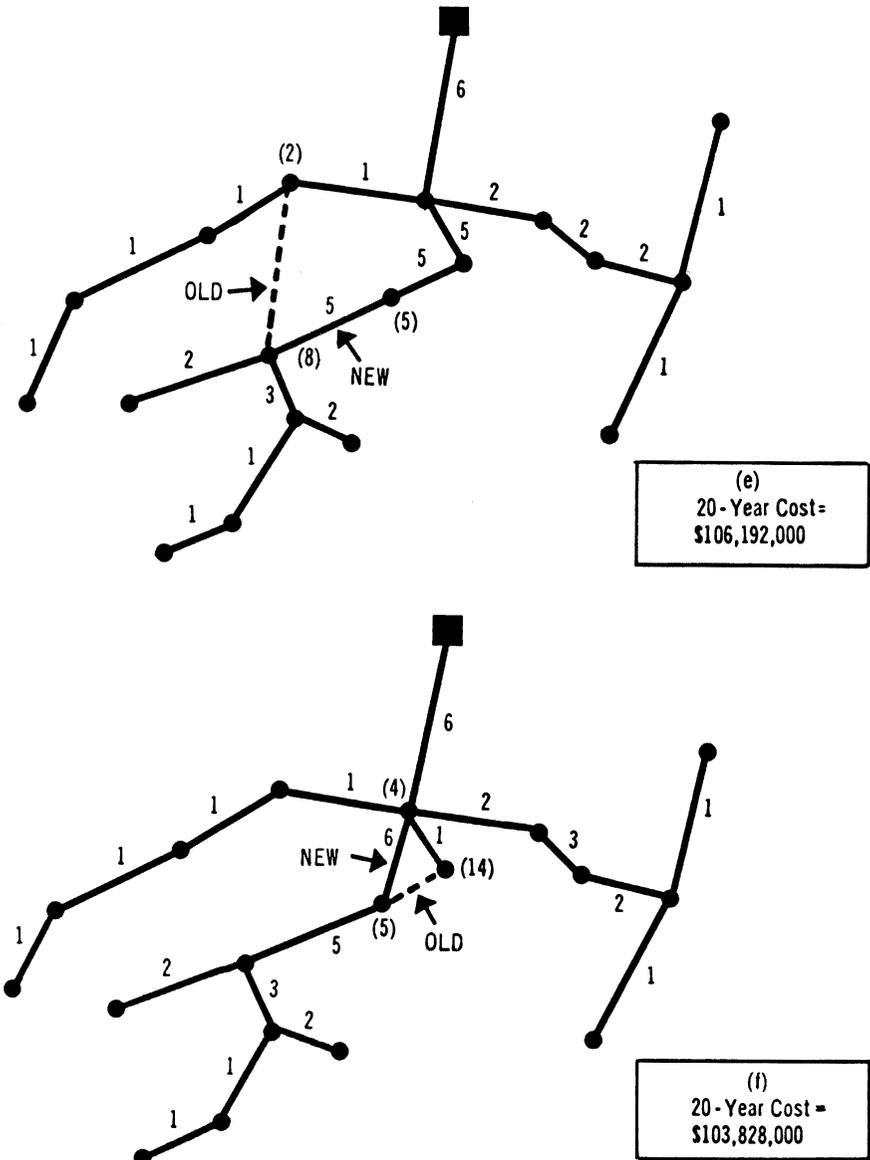


Fig. 8 (e-f)

exact form of this equation. For example, the efficiency factor of the pipeline may be changed, or an entirely different flow formula may be adopted. This could be desirable, for example, if two-phase flow systems are to be extensively examined. Another easily included feature is the effect of elevation or temperature variation on pipeline flow, which in this

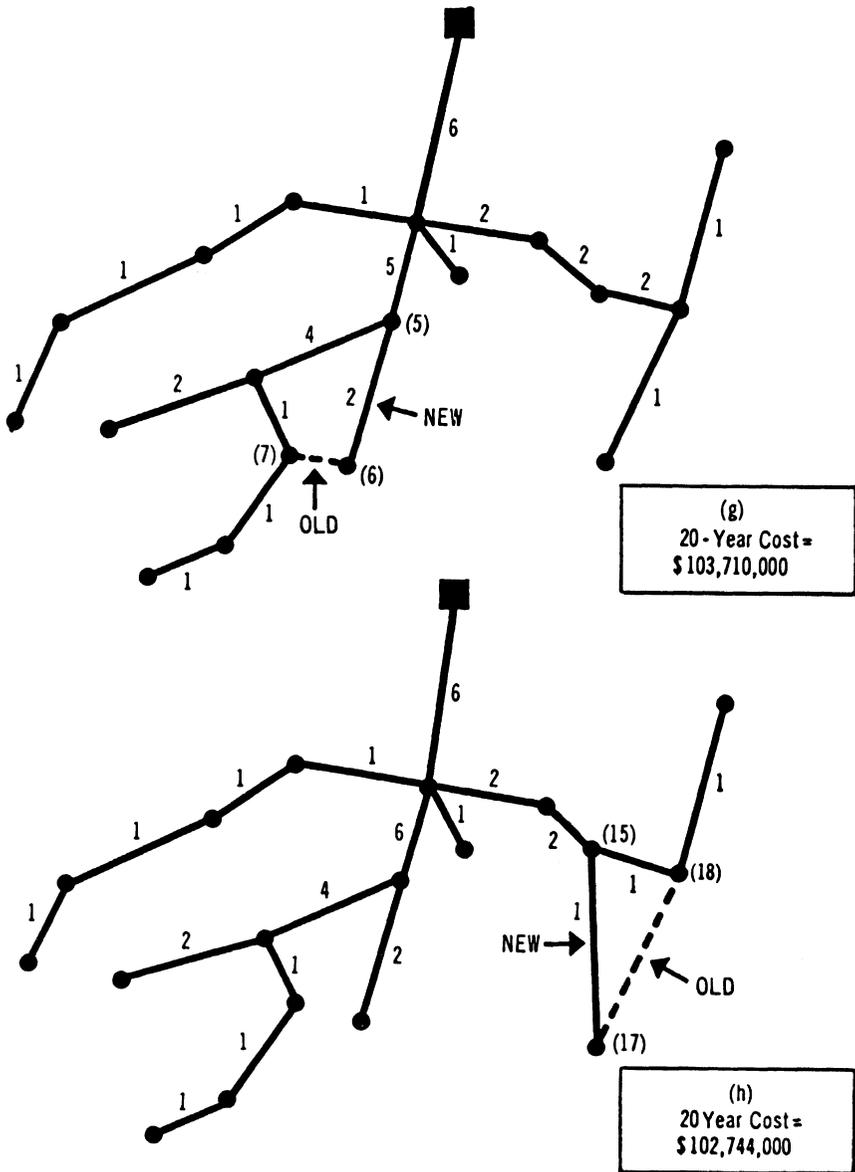


Fig. 8 (g-h)

study were taken to be negligible. In addition, different flow formulas may be used for different pipes.

Cost factors. The cost figures for pipeline construction, compressor operation, and amortization rates were supplied by the Federal Power

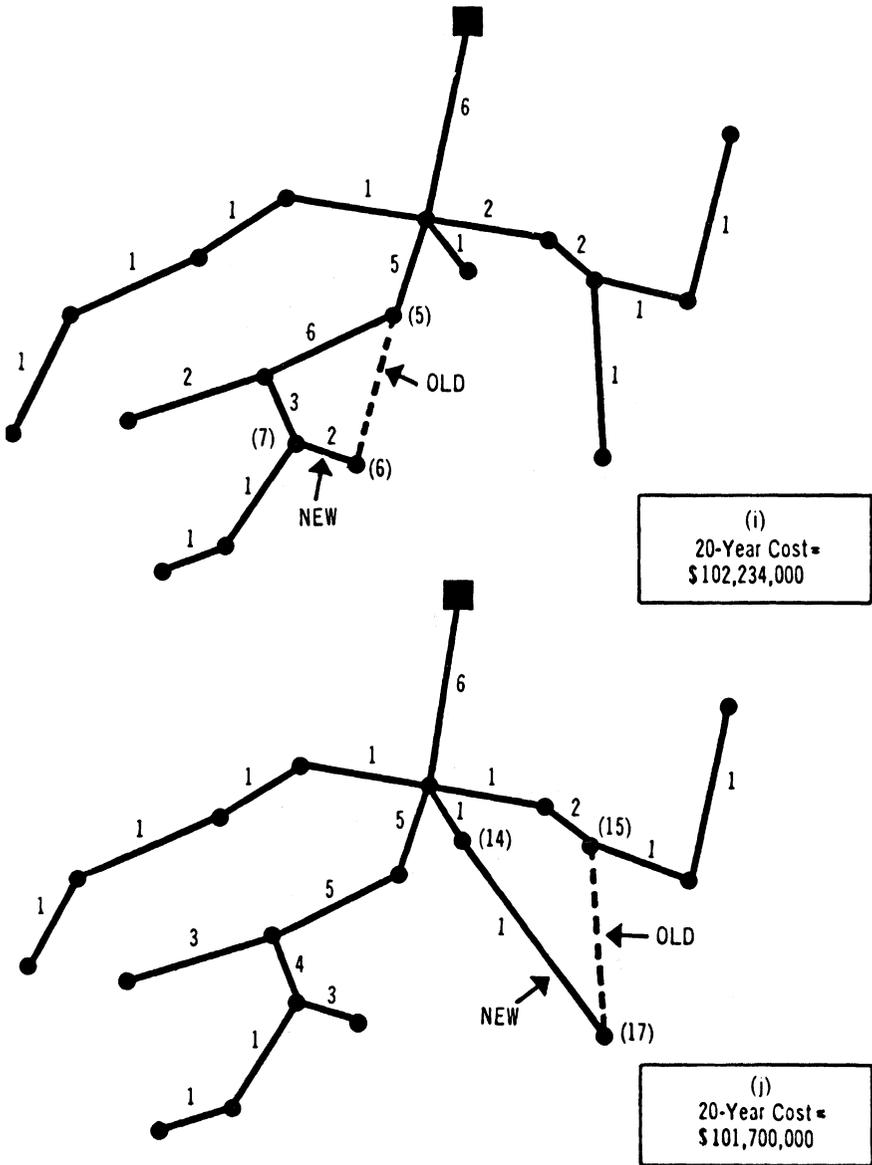


Fig. 8 (i-j)

Commission. Other factors can be introduced into the cost computation without difficulty. Two such factors are the cost of 'tie-ins' between pipelines and the cost of raising a pipeline from the ocean floor to a platform. Since the design methods only require that PCOST components for each diameter type be known, any method of costing can be used. For

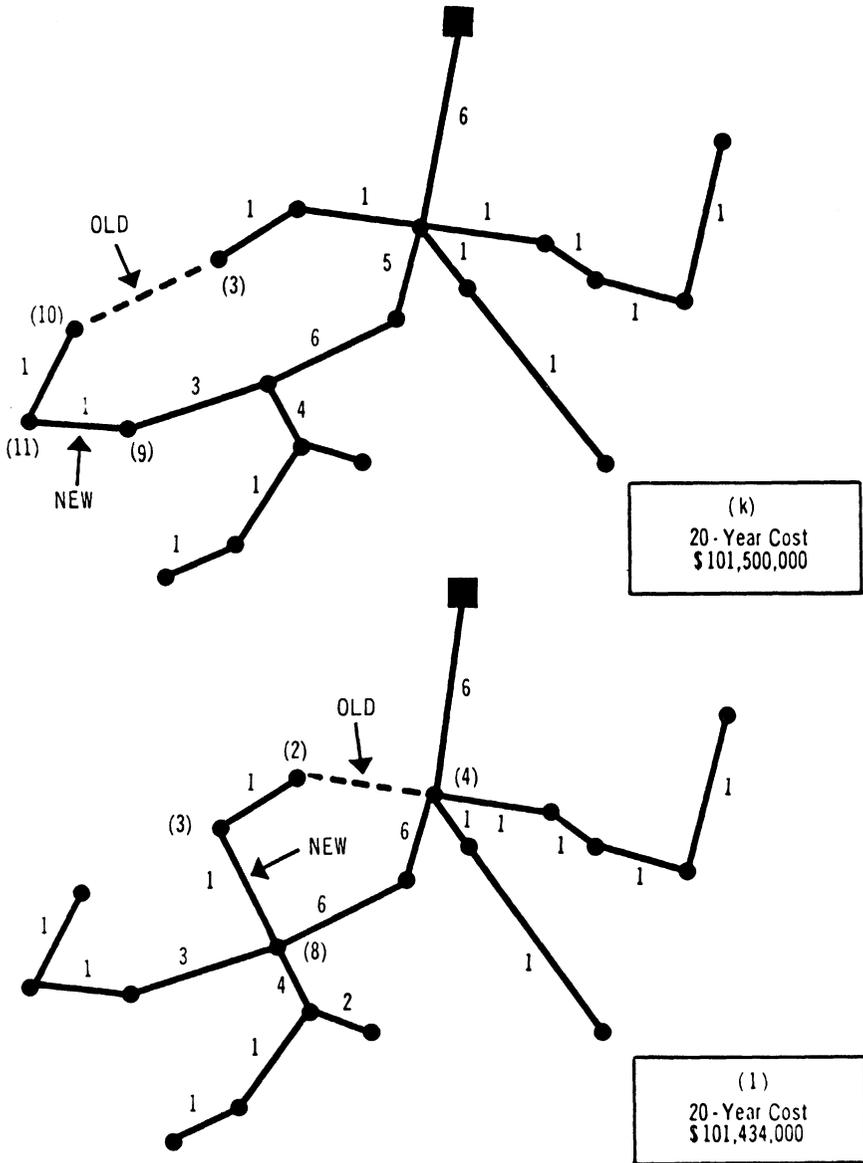


Fig. 8 (k-l)

example, if the terrain is uneven, the optimal location of a pipe connecting a specified pair of points could be the subject of a calculus-of-variations problem. Once the location and length of a pipe is computed, the PCOST and PSQ entries are known for each diameter type.

of the study. In all of these examples, it was not economical to construct new offshore platforms for collection points.

There are several other directions in which this study can be extended. The remainder of this section discusses a major modification to which further attention should be devoted, namely, the consideration of nontree networks.

If a system is initially designed and then augmented with the proposed optimization techniques, it is never necessary to analyze structures other than trees. However, such a necessity may arise with an existing system designed without regard to these optimization methods. In this case, it is important to be able to analyze the existing network if it is ever to be augmented efficiently.

For example, consider the network given in Fig. 9. The removal of line A from this network results in the tree shown in Fig. 1. The difference between these two structures is that, in the nontree structure shown in Fig. 9, gas at junction (11) may be sent through both pipes A and B. Even given the exact amount of gas being delivered by each field, it is not clear how much gas is flowing in pipe A or B. A new network model can be constructed by conceptually splitting node (11) into two nodes labeled (11') and (11''). An example of this is shown in Fig. 10. A tree network has now been obtained that models the nontree situation, and the tree techniques can be applied immediately. In this case, the amount of gas in each pipe is now a decision variable that must be iteratively adjusted in order that the pressures at (11) and (11') be equal. (These pressures must be equal since, in reality, the points represent a single junction with one input gas stream.)

FURTHER REMARKS

THE METHODS described in this paper have been programmed for a Univac 1108, and several programs have been written. One program, called OPTIMIZE, selects the pipe diameters optimally for a specified tree. The program accepts as input the coordinates and flows of up to 50 nodes, and any number of trees connecting these nodes. Each tree is independently processed to determine the pipe diameters that minimize cost.

The running time for OPTIMIZE depends on the number of nodes and the number of fixed-diameter branches in the trees to be processed. For a 20-node problem, the running time is about one second per tree.

The second program, called DESIGN, accepts as input a maximum of 50 nodes with their coordinates and flows. The program grows a starting tree that is then optimized by the heuristic method described earlier. Any set of branches can be specified to be in every tree considered. If desired, the diameter of these branches can also be specified. A typical running time for a 14-node problem with no fixed branches is 6.5 minutes on a

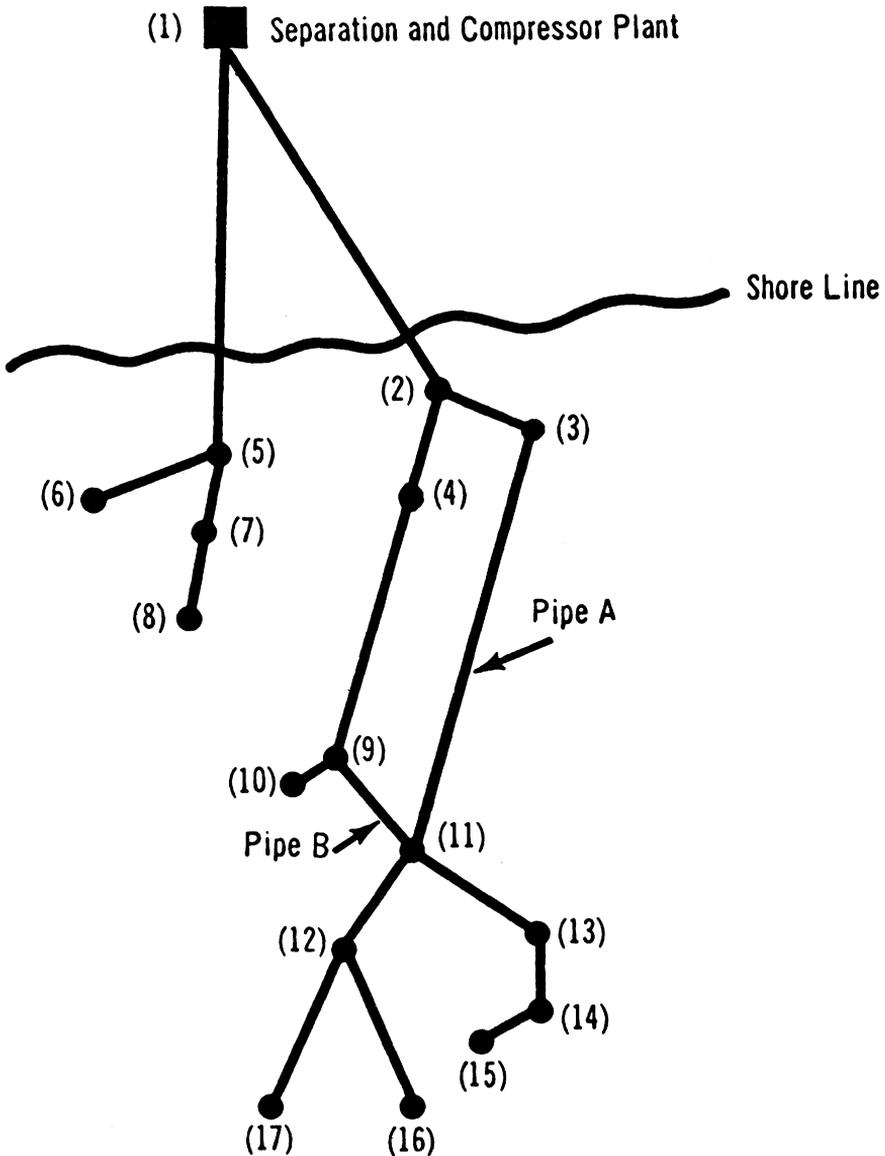


Fig. 9. A nontree structure. Removal of pipe A results in the tree shown in Fig. 1.

Univac 1108. In addition, since the completion of the computer program, methods have been developed that can reduce running time by at least an order of magnitude.

The last program written is called MOTHER. This program generates

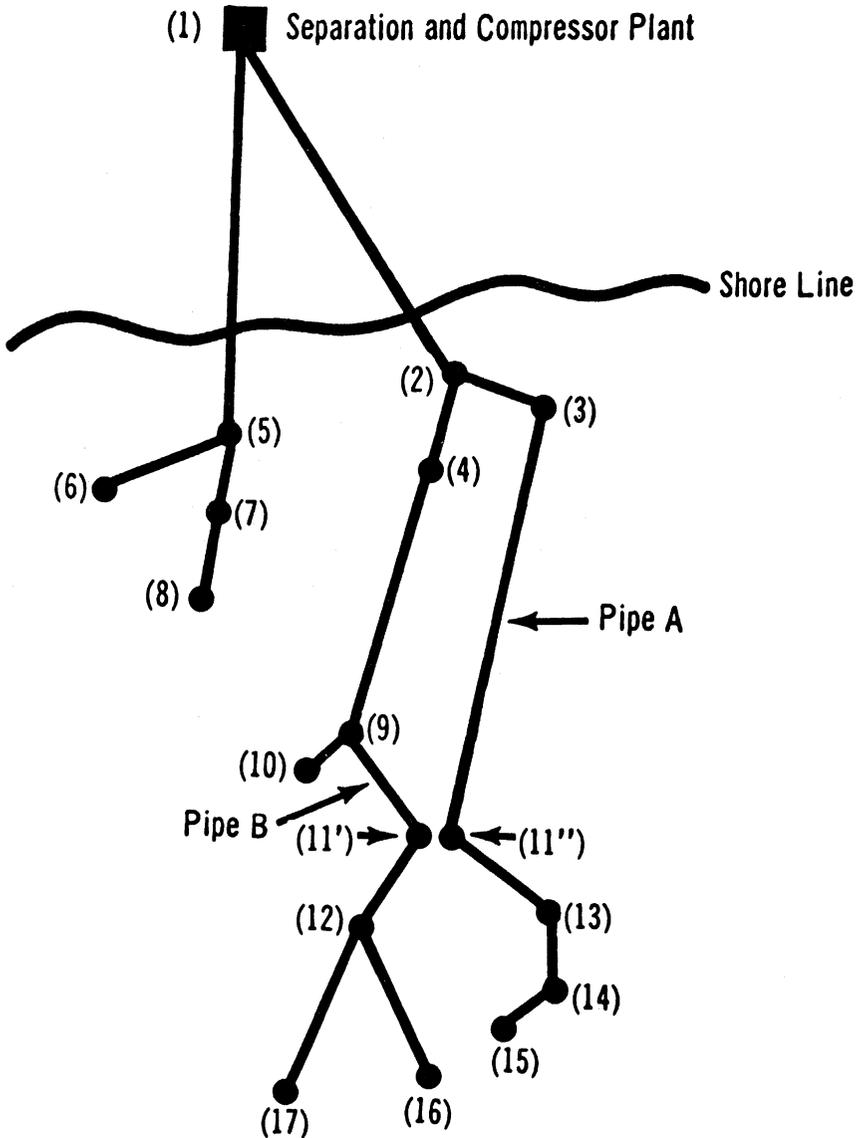


Fig. 10. A tree structure obtained by 'splitting' node (11) into two nodes labeled (11') and (11'').

all trees from an arbitrary 'mother' network. If desired, any set of branches may be fixed, along with their diameters. The output of MOTHER may then be used as the input of OPTIMIZE to determine the optimum pipe sizes for each tree.

Program MOTHER is useful for expanding an existing pipeline net-

work optimally to include newly discovered fields. To update a network, the entire existing structure can be specified as fixed while all possible augmentations are evaluated. A globally optimum augmentation is thus assured. A practical limitation to this approach is the addition of six or seven new fields. When adding more than this number of fields, it is desirable to limit the number of possible connections at each field. To add more than ten or twelve new fields, it is advisable to use program DESIGN. Again the existing branches are specified as fixed. The remaining structure is then generated to obtain a locally optimal tree.

The size of the programs that may be attacked by MOTHER has been found to be limited by OPTIMIZE. For example, a 20-node problem with 25 unfixed branches required 4.2 seconds to generate 1021 trees, but required 24.4 minutes to process these trees with OPTIMIZE.

It is interesting to note that, on problems for which it was feasible to obtain global optima using MOTHER, the same global optima were found using DESIGN. Finally, we note that more sophisticated versions of the methods have been developed by three of the authors (B.R., H.F., and D.M.R.) that can optimize networks with several hundred nodes in 10-15 minutes of time.

ACKNOWLEDGMENT

THIS WORK WAS supported by the Executive Office of the President, Office of Emergency Preparedness, Washington, D. C., and summarizes a study prepared for the Federal Power Commission.

REFERENCES

1. D. L. KATZ, *Handbook of Natural Gas Engineering*, McGraw Hill, New York, 1959.
2. *Steady State Flow Computation Manual for Natural Gas Transmission Lines*, American Gas Association, New York, 1964.
3. R. G. BUSAKER AND T. L. SAATY, *Finite Graphs and Networks*, McGraw Hill New York, 1965.