On the Optimal Solutions to AND/OR Series-Parallel Graphs

RICHARD SIMON AND RICHARD C. T. LEE

National Institutes of Health, Public Health Service, Department of Health, Education and Welfare, Bethesda, Maryland*

ABSTRACT. This paper is concerned with efficient ways to find optimal solutions to AND/OR graphs. Although the general methods are still at large, we have found an efficient way to obtain optimal solutions to AND/OR series-parallel graphs. This is achieved by reducing an AND/OR series-parallel graph to an AND/OR tree. Once a graph is reduced to a tree, all the known exact and heuristic methods of tree searching can be applied.

KEY WORDS AND PHRASES: AND/OR series-parallel graphs, optimal solutions, problem-solving, sum-cost, max-cost, s-p decomposable, hyper-series-parallel graphs, dynamic programming, branch-and-bound approach

CR CATEGORIES: 3.64, 5.32, 5.41

1. *Introduction*

It has been pointed out by many artificial intelligence researchers that problem-solving procedures can be represented by AND/OR trees [3, 6, 8]. A natural extension of the AND/OR tree representation is the AND/OR graph representation. Consider Figure 1. Interpret all the vertices with arcs through their departing edges as AND-vertices. Interpret all the other vertices as OR-vertices (we shall follow this rule in the rest of this paper). Then Figure 1 depicts that problem A can be solved only by solving both problems B and C. Problem B is solved by either solving problem D or solving problem E. All the terminal vertices represent problems that have already been solved. The numbers labeled on the edges represent costs. For example, in Figure 1, since edge EG is labeled 6, edge EH is labeled 5, and vertex E is an AND-vertex, the total cost of solving problem E is $6 + 5 = 11$.

A solution to an AND/OR graph G is a subgraph of G and is defined in the same fashion as solutions to an AND/OR tree [6]. Three solutions of the AND/OR graph of Figure 1 are given in Figure 2.

Nilsson in [6] mentioned two cost functions, the sum-cost and the max-cost. Briefly, the sum-cost of a solution G is the sum of all the edge costs of G, while the max-cost is based on the concept of a path through the solution.

Let us now assume that the sum-cost is used. Consider Figure 3. The total cost of the tree can be found easily by working backwards:

$$\begin{aligned} C(D) &= c(D, F) + c(D, G), \\ C(E) &= c(E, H) + c(E, I), \\ C(B) &= c(B, D) + C(D), \\ C(C) &= c(C, E) + C(E), \end{aligned}$$

* Division of Computer Research and Technology.

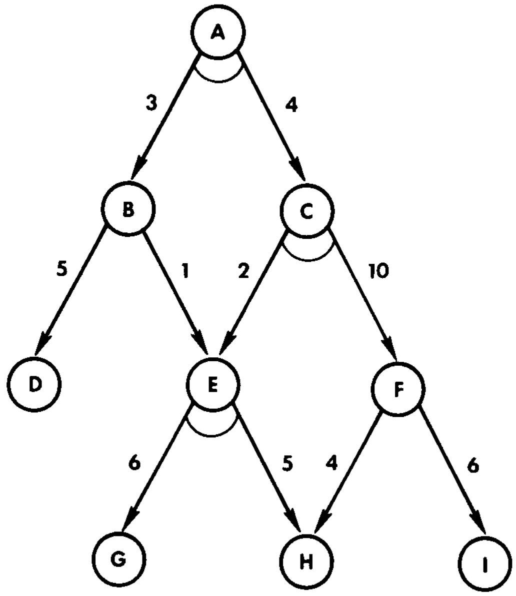

FIG. 1. An AND/OR graph.

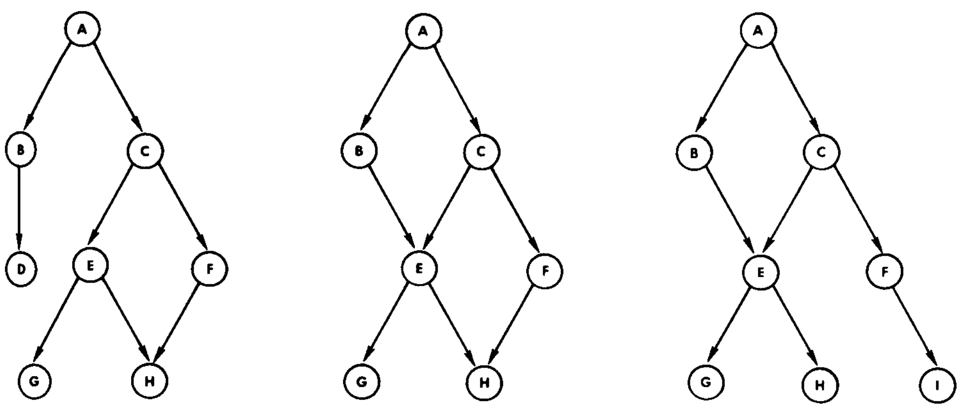

FIG. 2. Three solutions to the AND/OR graph in Figure 1.

and

$$C(A) = c(A, B) + C(B) + c(A, C) + C(C).$$

Consider Figure 4. The cost of the AND/OR graph cannot be found easily as in the previous case. A common mistake is to use the following formulas:

$$\begin{aligned} C(D) &= c(D, E) + c(D, F), \\ C(B) &= c(B, D) + C(D), \\ C(C) &= c(C, D) + C(D), \end{aligned}$$

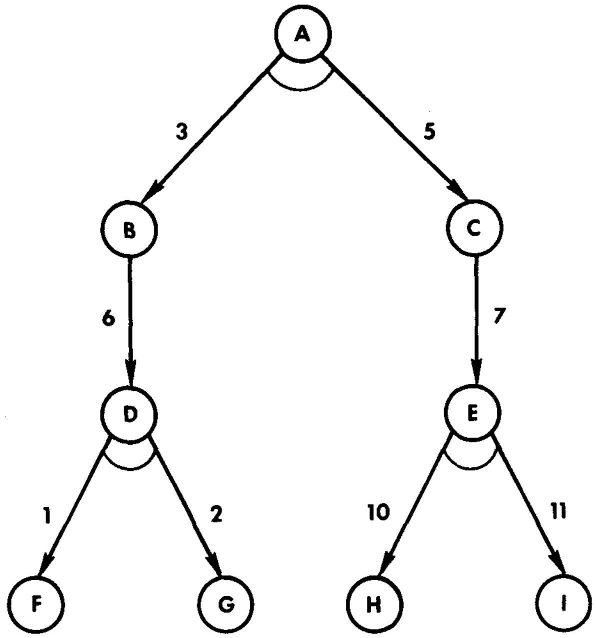

FIG. 3

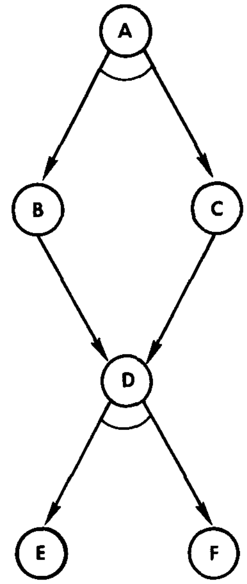

FIG. 4

and

$$\begin{aligned} C(A) &= c(A, B) + C(B) + c(A, C) + C(C) \\ &= c(A, B) + c(A, C) + c(B, D) + c(C, D) + 2c(D, E) + 2c(D, F). \end{aligned}$$

One can see immediately that this is not correct. The cost of vertex D is counted more than once. Imagine that D represents a theorem which can be proven by using axioms E and F. Then, once theorem D is proved, it will be stored in the computer memory as a proved theorem and one can use it without having to prove it again.

Thus, for a very large and complex AND/OR graph G, it is by no means easy to obtain optimal solutions to G. In this paper, we shall show that one can reduce a series-parallel graph to a directed tree without changing the cost of optimal solutions. Exact or heuristic techniques can then be applied to the AND/OR tree and the resulting solution can be easily transformed into a solution to the original AND/OR series-parallel graph. If the solution is optimal for the tree, then the transformed solution is optimal for the series-parallel graph. This result is later extended to hyper-series-parallel graphs.

2. *Series-Parallel Graphs*

A *graph* is a pair (V, r) in which V is a set of elements called vertices and r is a binary, symmetric, irreflexive relation defined on V. If vrv', then we say that there is an edge joining v and v'. A *directed* graph is a pair (V, R) in which V is a set of elements called vertices, and R is a subset of $V \times V$. In this section we shall define the set of series-parallel graphs and show how the edges of a series-parallel graph can be assigned a natural orientation so that the graph may be considered a directed graph.

A two-terminal graph is a 4-tuple (V, r, v, v') in which (V, r) is a graph, and v and v' are distinguished vertices called the first terminal and second terminal, respectively. A two-terminal graph (V, r, v, v') is called the *series composition* of two-terminal graphs (V_1, r_1, v_1, v_1') and (V_2, r_2, v_2, v_2') if $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{v_1'\}$, $r = r_1 \cup r_2$, $v = v_1$, $v' = v_2'$, and $v_1' = v_2$. A two-terminal graph (V, r, v, v') is called the *parallel composition* of two-terminal graphs (V_1, r_1, v_1, v_1') and (V_2, r_2, v_2, v_2') if $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{v, v'\}$, $r = r_1 \cup r_2$, $v = v_1 = v_2$, and $v' = v_1' = v_2'$. Examples of series and parallel compositions of two-terminal graphs are given in Figures 5 and 6, respectively.

The *unit graph* of two vertices is the graph containing only the two vertices and the edge joining them. The class of *two-terminal series-parallel (s-p) graphs* is the smallest class of two-terminal graphs which contains the unit graphs and is closed under series and parallel compositions; that is, if (V_1, r_1, v_1, v_1') and (V_2, r_2, v_2, v_2') are in the class, then (1) $v_1' = v_2$ and $V_1 \cap V_2 = \{v_1'\}$ imply that the series composition of these two graphs is in the class and (2) $v_1 = v_2$, $v_1' = v_2'$, and $V_1 \cap V_2 = \{v_1, v_1'\}$ imply that the parallel composition of these two graphs is in the class.

A graph (V, r) with distinguished vertices $(v, v_1, \cdots, v_n)$ is said to be a *series-parallel (s-p) graph* if it is a two-terminal s-p graph or if the two-terminal graph (V', r', v, v') is a two-terminal s-p graph, where (V', r', v, v') is obtained from $(V, r, v, v_1, \cdots, v_n)$ by including a new vertex v' and edges $\{v', v_i\}$ for $i = 1, \cdots, n$. The graph (V', r', v, v') is called the two-terminal s-p graph associated with $(V, r, v, v_1, \cdots, v_n)$ for $n \geq 1$. If $n = 1$ and (V, r, v, v_1) is a two-terminal s-p graph, then it is its own associated graph. An example of an s-p graph and its associated two-terminal s-p graph is given in Figure 7. Figure 8 shows a graph that is not series-parallel.

An s-p graph $(V, r, v, v_1, \cdots, v_n)$ is called the series composition of a two-terminal s-p graph (V_1, r_1, v, v^*) and an s-p graph $(V_2, r_2, v^*, v_1, \cdots, v_n)$ if $V =$

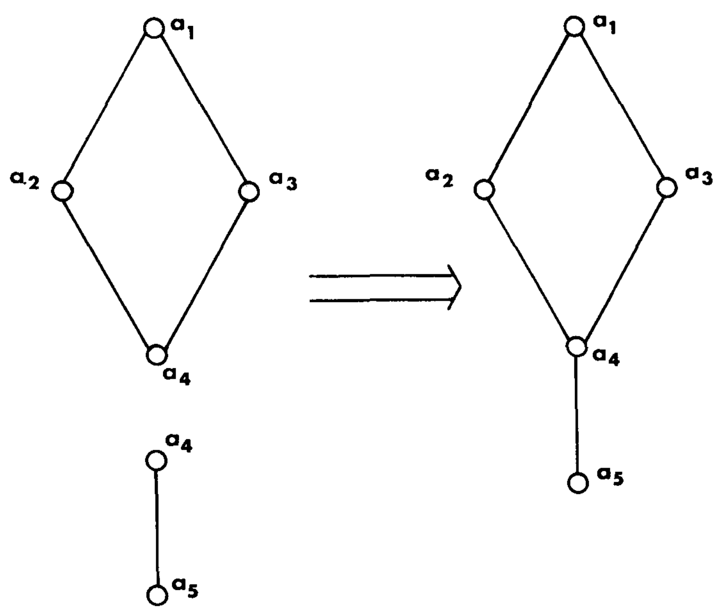

FIG. 5. Series composition of two-terminal graphs.

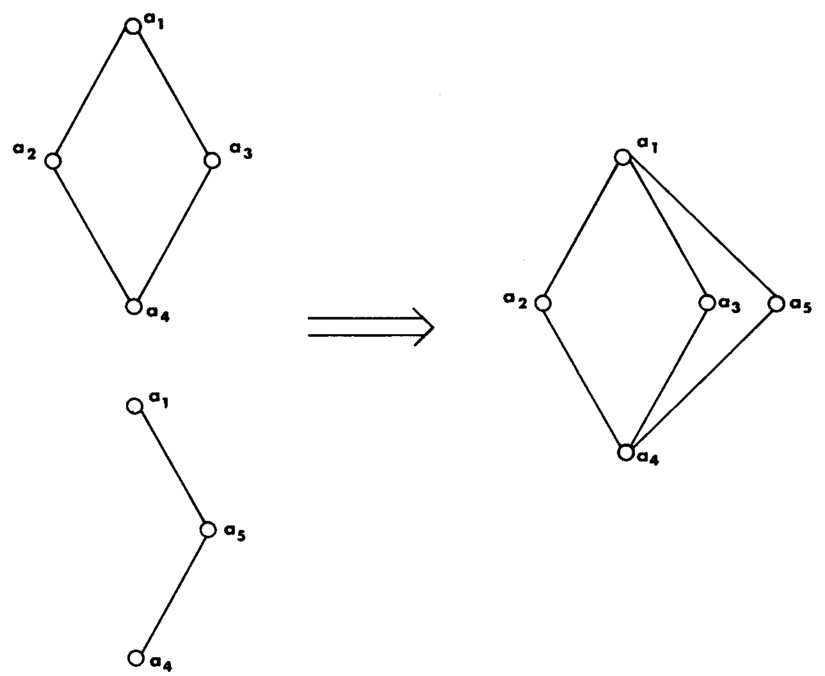

FIG. 6. Parallel composition of two-terminal graphs.

$V_1 \cup V_2$, $V_1 \cap V_2 = \{v^*\}$, and $r = r_1 \cup r_2$. An s-p graph $(V, r, v, v_1, \cdots, v_n)$ is called the parallel composition of s-p graphs $(V_1, r_1, v, x_1, \cdots, x_k)$ and $(V_2, r_2, v, y_1, \cdots, y_m)$ if the associated two-terminal s-p graph (V, r, v, v') is the parallel composition of the two-terminal s-p graph (V_1, r_1, v, v') associated with $(V_1, r_1, v, x_1, \cdots, x_k)$ and the two-terminal s-p graph (V_2, r_2, v, v') associated with $(V_2, r_2, v, y_1, \cdots, y_m)$. Figures 9 and 10 give examples of the series and parallel compositions of s-p graphs, respectively. It follows from the definition of the class of s-p graphs that each s-p graph, other than the unit graph, is either the series or parallel composition of two s-p graphs.

The edges of an s-p graph can be assigned a natural orientation using the notion of semipath. A semipath from vertex v_1 to vertex v_n of a graph (V, r) is a finite sequence of distinct vertices $(v_1, v_2, \cdots, v_n)$ such that $(v_i, v_{i+1}) \in r$ for $1 \leq i \leq n - 1$. For any two-terminal graph (V, r, v, v'), the link relation R is said to hold between vertices x and y if x immediately precedes y in some semipath from v to v'. It is shown in [2] that for any two vertices x and y joined by an edge in a two-terminal s-p graph, either $(x, y) \in R$ or $(y, x) \in R$ but not both. R establishes a natural orientation for each edge. Given an s-p graph, we let the orientation of each edge be given by the orientation of the corresponding edge in its associated two-terminal s-p graph. In the following sections we shall consider an s-p graph as a directed graph.

In a directed graph (V, R), a vertex y is called a successor (predecessor) of a vertex x if $(x, y) \in R$ $[(y, x) \in R]$. The set of successors (predecessors) of a vertex x is denoted by $T(x)$ $(T^{-1}(x))$. The outdegree and indegree of a vertex x are defined as $| T(x) |$ and $| T^{-1}(x) |$, respectively, where $| \quad |$ denotes cardinality. A ver-

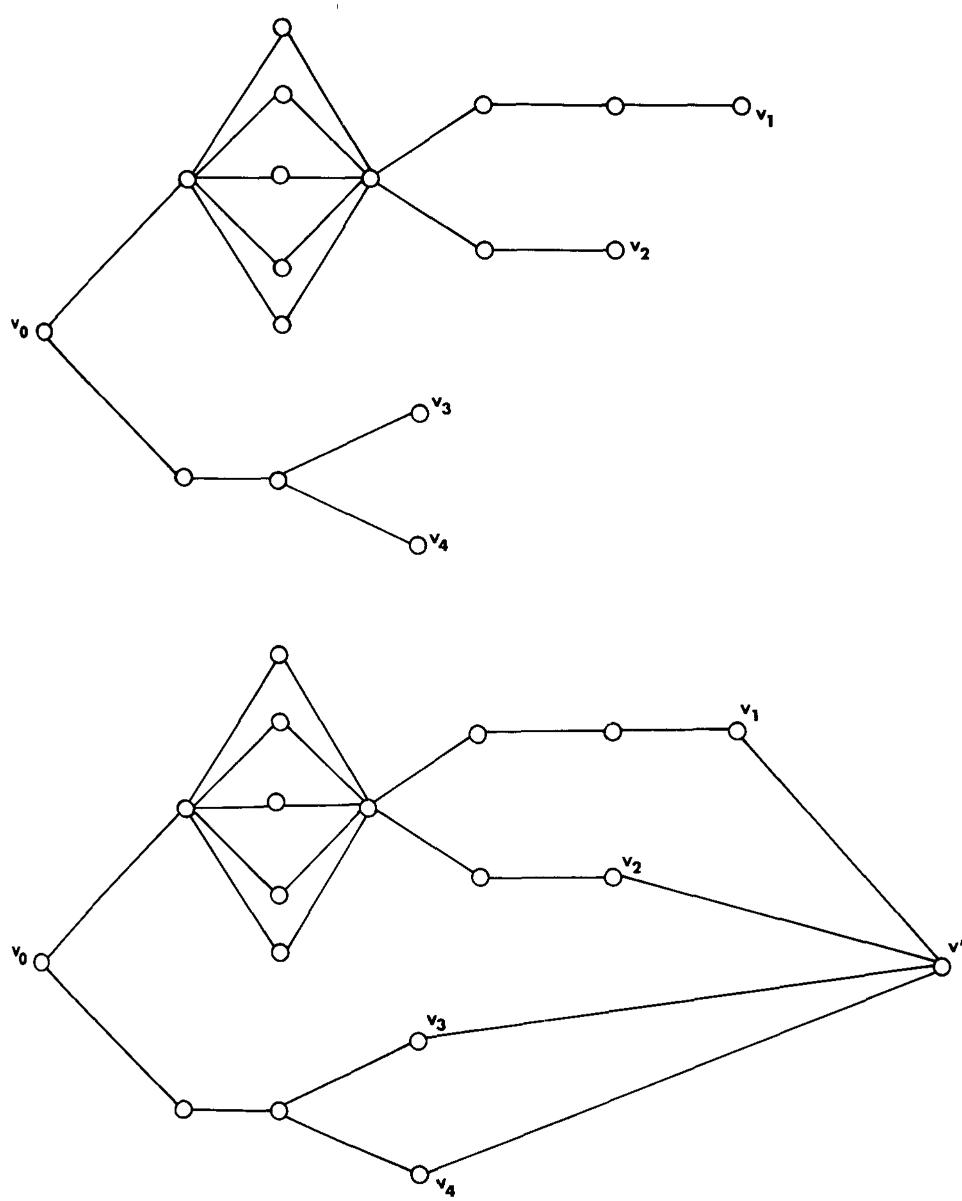

FIG. 7. An s-p graph and its associated two-terminal s-p graph.

tex of indegree zero is called a root. In an s-p graph $(V, r, v_0, v_1, \cdots, v_n)$, it is easily shown that, relative to the link relation R, v_0 is the unique root and $v_1, \cdots, v_n$ are the only vertices with outdegree zero.

3. *Reduction of Series-Parallel Graphs to Directed Trees*

In this section we shall define the notion of a "solution" to an AND/OR directed graph as in [6], and shall introduce the concept of series-parallel decomposability of a cost function. We shall show that the commonly used cost functions, the sum-

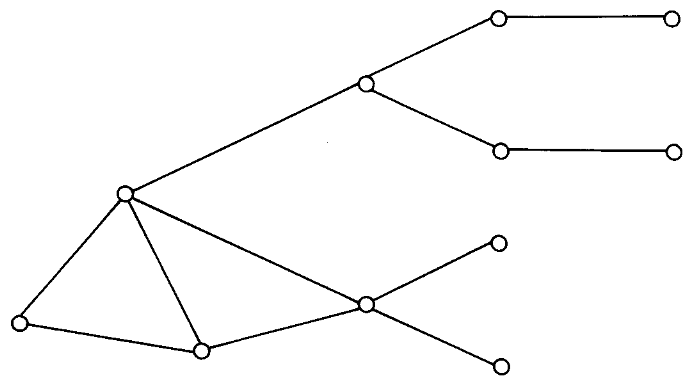

FIG. 8. A non-s-p graph.

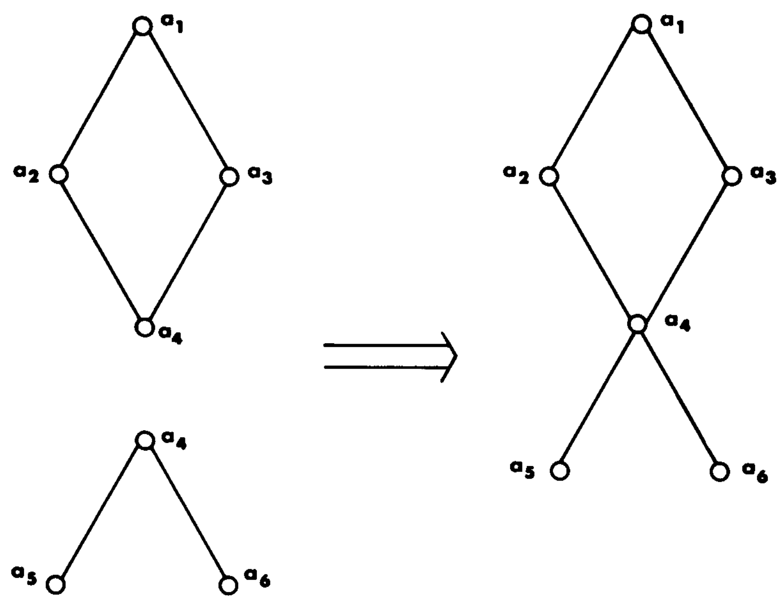

FIG. 9. Series composition of s-p graphs.

cost and the max-cost, are decomposable. Finally, we present the main contribution of this paper in the form of a theorem, i.e. an s-p graph can be reduced to a directed tree in a manner which permits a correspondence to be made between solutions of the tree and solutions of the graph; and if the cost function is s-p decomposable, then this correspondence preserves the cost of a solution.

Let (V, R) denote a directed graph with unique root v_0. Let the vertices of (V, R) having outdegree greater than zero be partitioned into sets A (denoting the "and" vertices) and O (denoting the "or" vertices). A subgraph of (V, R) is a directed graph (X, B) such that $X \subset V$ and $B \subset R$. A *solution* to (V, R) is a subgraph (X, B) having the following properties:

(1) $v_0 \in X$,

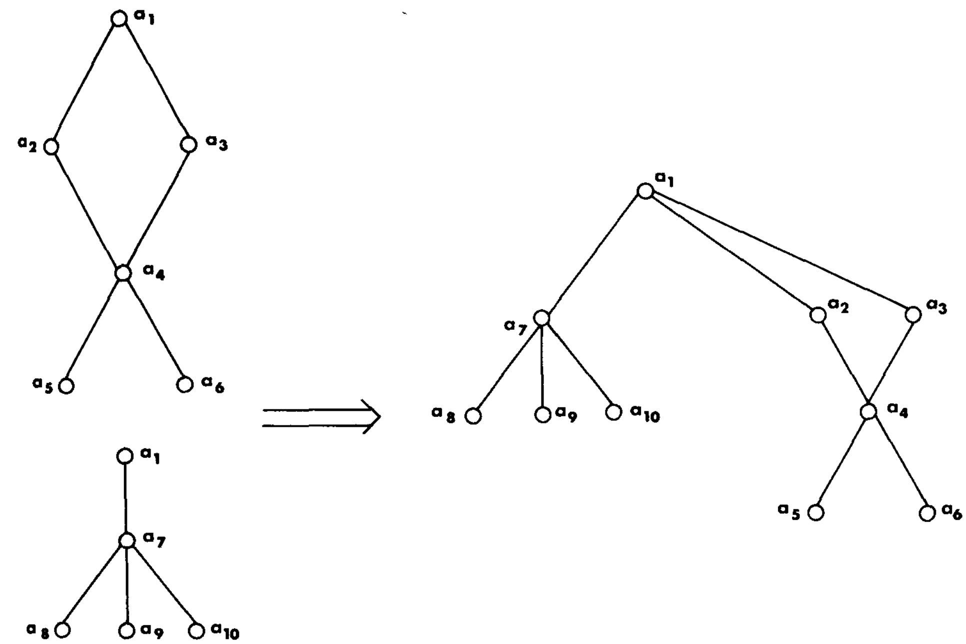

FIG. 10. Parallel composition of s-p graphs.

(2) $v \in X \cap A$ implies that $v' \in X$ and $(v, v') \in B$ for each $v' \in T(v)$,

(3) $v \in X \cap O$ implies that $v' \in X$ and $(v, v') \in B$ for some $v' \in T(v)$.

Condition (1) states that the root of (V, R) must be in the solution. Conditions (2) and (3) state that if v is in the solution and is an "and" ("or") vertex, then each (at least one) immediate successor v' of v is in the solution and the directed edge (v, v') is in the solution.

The following results are easily proven by induction on the size of the graph and the fact that each s-p graph other than the unit graph is either the series or parallel composition of two s-p graphs:

LEMMA 1. *A solution to an s-p graph is an s-p graph.*

LEMMA 2. *If an s-p graph (V, R) is the series (parallel) composition of s-p graphs (V_1, R_1) and (V_2, R_2), and if (X_j, B_j) is a solution to (V_j, R_j) for $j = 1, 2$, then the series (parallel) composition of (X_1, B_1) and (X_2, B_2) is a solution to (V, R).*

Let S denote the set of s-p graphs with edge costs defined on them. A cost function is a mapping from S to the real numbers. A cost function C is called *s-p decomposable* if the following condition holds: If an s-p graph (V, R) is the series (parallel) composition of s-p graphs (V_1, R_1) and (V_2, R_2), and an s-p graph $(\bar{V}, \bar{R})$ is the series (parallel) composition of s-p graphs $(\bar{V}_1, \bar{R}_1)$ and $(\bar{V}_2, \bar{R}_2)$, then (1) $C(V_j, R_j) = C(\bar{V}_j, \bar{R}_j)$ for $j = 1, 2$ implies that $C(V, R) = C(\bar{V}, \bar{R})$, and (2) $C(V_j, R_j) \leq C(\bar{V}_j, \bar{R}_j)$ for $j = 1, 2$ implies that $C(V, R) \leq C(\bar{V}, \bar{R})$.

The most commonly used cost functions are the sum-cost and the max-cost [6]. The sum-cost of a solution (X, B) is defined by

$$C_{\mathrm{sum}}(X, B) = \sum_{b \in B} c(b).$$

A *path* of a directed graph (X, B) is a sequence of distinct vertices $(v_1, v_2, \cdots,$

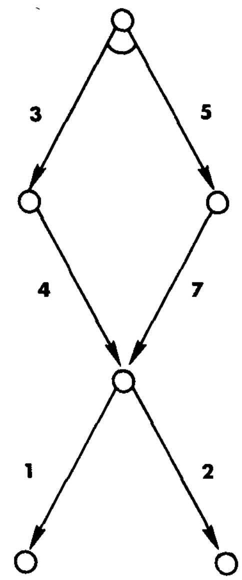

FIG. 11

v_n) such that $(v_i, v_{i+1}) \in B$, $1 \leq i \leq n - 1$. The edges (v_i, v_{i+1}) are called the edges of the path. The cost of a path P of a solution (X, B) is defined as the sum of the edge costs in the path P. The max-cost $C_{\max}(X, B)$ of a solution (X, B) is the cost of the path of (X, B) having maximum path cost.

Figure 11 shows one solution graph to a certain AND/OR directed graph. On each edge is labeled the cost of that edge. The sum-cost of this solution is $C_{\text{sum}} = 3 + 5 + 4 + 7 + 1 + 2 = 22$. The max-cost of this solution is

$$\begin{aligned} C_{\max} &= \max\{3 + 4 + 1, 3 + 4 + 2, 5 + 7 + 1, 5 + 7 + 2\} \\ &= \max\{8, 9, 13, 14\} \\ &= 14. \end{aligned}$$

LEMMA 3. *The sum-cost is s-p decomposable.*

PROOF. If an s-p graph (V, R) is the series or parallel composition of s-p graphs (V_1, R_1) and (V_2, R_2), then the sum-cost C can be written

$$\begin{aligned} C(V, R) &= \sum_{r \in R} c(r) \\ &= \sum_{r \in R_1} c(r) + \sum_{r \in R_2} c(r) \\ &= C(V_1, R_1) + C(V_2, R_2). \end{aligned}$$

It follows easily that the sum-cost is s-p decomposable. Q.E.D.

LEMMA 4. *The max-cost is s-p decomposable.*

PROOF. Let an s-p graph (V, R) be the series composition of s-p graphs (V_1, R_1) and (V_2, R_2), and let $C(\quad)$ denote the max-cost. If p_1 and p_2 are paths of maximum cost in (V_1, R_1) and (V_2, R_2) respectively, then the catenation of p_1 and p_2 is a path of maximum cost in (V, R). Thus,

$$C(V, R) = C(V_1, R_1) + C(V_2, R_2). \tag{1}$$

Suppose that (V, R) is the parallel composition of (V_1, R_1) and (V_2, R_2). By definition of parallel composition, any path of (V, R) is either a path of (V_1, R_1) or a path of (V_2, R_2). Thus,

$$C(V, R) = \max\{C(V_1, R_1), C(V_2, R_2)\}. \tag{2}$$

It follows from (1) and (2) that the max-cost C is s-p decomposable. Q.E.D

A solution to an s-p graph is said to be *optimal* if it has minimum cost. The cost of an optimal solution to (V, R) is denoted $C^*(V, R)$; that is,

$$C^*(V, R) = \min_{(Y,D)} C(Y, D),$$

where the minimum is taken over the set of solutions to (V, R).

LEMMA 5. *If an s-p graph (V, R) either (a) is the series composition of s-p graphs (V_1, R_1) and (V_2, R_2), or (b) has an "and" vertex as its root and is the parallel composition of s-p graphs (V_1, R_1) and (V_2, R_2), and if relative to an s-p decomposable cost function (X_j, B_j) is an optimal solution to (V_j, R_j) for $j = 1, 2$, then in case (a) the series composition and in case (b) the parallel composition of (X_1, B_1) and (X_2, B_2) is an optimal solution to (V, R).*

PROOF. Let (Y, D) be an optimal solution to (V, R). Let $Y_j = Y \cap V_j$ and $D_j = D \cap R_j$ for $j = 1, 2$. It is easily shown that (Y_j, D_j) is a solution to (V_j, R_j) for $j = 1, 2$ and that (Y, D) is the composition of (Y_1, D_1) and (Y_2, D_2). Thus, $C(Y_j, D_j) \geq C(X_j, B_j)$ for $j = 1, 2$. From the s-p decomposability of C it thus follows that $C(Y, D) \geq C(X, B)$, where (X, B) is the composition of (X_1, B_1) and (X_2, B_2). Thus, (X, B) is an optimal solution to (V, R). Q.E.D.

LEMMA 6. *If an s-p graph (V, R) has an "or" vertex as its root and is the parallel composition of s-p graphs (V, R) and (V_2, R_2), and if relative to an s-p decomposable cost function (X_j, B_j) is an optimal solution to (V_j, R_j) for $j = 1, 2$, then either (X_1, B_1) or (X_2, B_2) or the parallel composition of (X_1, B_1) and (X_2, B_2) is an optimal solution to (V, R).*

PROOF. It is easily shown that any solution to (V, R) consists of either (a) a solution to (V_1, R_1) for some $i \in \{1, 2\}$ or (b) the parallel composition of a solution to (V_1, R_1) and a solution to (V_2, R_2). Let (Y, D) be an optimal solution to (V, R) and assume that (Y, D) consists of the parallel composition of (Y_1, D_1) and (Y_2, D_2), where (Y_1, D_1) is a solution to (V_1, R_1) and (Y_2, D_2) is a solution to (V_2, R_2). Thus, $C(Y_j, D_j) \geq C(X_j, B_j)$; since C is s-p decomposable, it follows that $C(Y, D) \geq C(X, B)$, where (X, B) is the parallel composition of (X_1, B_1) and (X_2, B_2). Since (Y, D) is optimal, it follows that (X, B) is an optimal solution to (V, R).

Suppose instead that (Y, D) is an optimal solution to (V, R) and that it is a solution to (V_i, R_i) for either $i = 1$ or $i = 2$. Thus, $C(Y, D) \geq C(X_i, B_i)$. Since (X_i, B_i) is a solution to (V, R), it follows that (X_i, B_i) is an optimal solution to (V, R). Q.E.D.

We shall now show that if a cost function C is s-p decomposable, then the induced function C^* is also s-p decomposable. The s-p decomposability of C^* is to be interpreted as in the definition of s-p decomposability with the additional stipulation that, in the case of parallel composition, the roots of (V, R) and $(\bar{V}, \bar{R})$ must be either both "and" vertices or both "or" vertices.

LEMMA 7. *If C is s-p decomposable, then C^* is s-p decomposable.*

PROOF. Let (V, R) be an s-p graph which is the series (parallel) composition of s-p graphs (V_1, R_1) and (V_2, R_2). Let $(\bar{V}, \bar{R})$ be an s-p graph which is the series (parallel) composition of s-p graphs $(\bar{V}_1, \bar{R}_1)$ and $(\bar{V}_2, \bar{R}_2)$. Let (X_j, B_j) be an optimal solution to (V_j, R_j) for $j = 1, 2$, and let $(\bar{X}_j, \bar{B}_j)$ be an optimal solution to $(\bar{V}_j, \bar{R}_j)$ for $j = 1, 2$. Let (X, B) be the composition of (X_1, B_1) and (X_2, B_2), and let $(\bar{X}, \bar{B})$ be the composition of $(\bar{X}_1, \bar{B}_1)$ and $(\bar{X}_2, \bar{B}_2)$. We shall distinguish two cases.

Case 1: Either we are dealing with series compositions or we are dealing with parallel compositions in which the roots of V and $\bar{V}$ are both "and" vertices. If $C^*(V_j, R_j) = C^*(\bar{V}_j, \bar{R}_j)$ for $j = 1, 2$, then $C(X_j, B_j) = C(\bar{X}_j, \bar{B}_j)$ for $j = 1, 2$. It follows from the s-p decomposability of C that $C(X, B) = C(\bar{X}, \bar{B})$. It follows from Lemma 5 that (X, B) and $(\bar{X}, \bar{B})$ are optimal solutions to (V, R) and $(\bar{V}, \bar{R})$, respectively. Thus, $C^*(V, R) = C^*(\bar{V}, \bar{R})$.

If $C^*(V_j, R_j) \leq C^*(\bar{V}_j, \bar{R}_j)$ for $j = 1, 2$, then the same argument as above is valid when "=" is replaced by "$\leq$". Thus, $C^*(V, R) \leq C^*(\bar{V}, \bar{R})$.

Case 2: (V, R) is the parallel composition of (V_1, R_1) and (V_2, R_2), $(\bar{V}, \bar{R})$ is the parallel composition of $(\bar{V}_1, \bar{R}_1)$ and $(\bar{V}_2, \bar{R}_2)$, and the roots of V and $\bar{V}$ are both "or" vertices. If $C^*(V_j, R_j) = C^*(\bar{V}_j, \bar{R}_j)$ for $j = 1, 2$, then $C(X_j, B_j) = C(\bar{X}_j, \bar{B}_j)$ for $j = 1, 2$. It follows from the s-p decomposability of C that $C(X, B) = C(\bar{X}, \bar{B})$. From Lemma 6 it follows that either (X_1, B_1) or (X_2, B_2) or (X, B) is an optimal solution to (V, R), and that either $(\bar{X}_1, \bar{B}_1)$ or $(\bar{X}_2, \bar{B}_2)$ or $(\bar{X}, \bar{B})$ is an optimal solution to $(\bar{V}, \bar{R})$. It follows that $C^*(V, R) = C^*(\bar{V}, \bar{R})$.

If $C^*(V_j, R_j) \leq C^*(\bar{V}_j, \bar{R}_j)$ for $j = 1, 2$, then the same argument as above is valid when "=" is replaced by "$\leq$". Thus, $C^*(V, R) \leq C^*(\bar{V}, \bar{R})$. Q.E.D.

For the rest of this section, we shall discuss the reduction of s-p graphs.

A *parallel component* of an s-p graph (V, R) is a two-terminal s-p subgraph $(\bar{V}, \bar{R})$ which is the parallel composition of distinct two-terminal s-p subgraphs. Figure 12 depicts an s-p graph and its parallel components.

A *proper parallel component* of (V, R) is a parallel component which is not (V, R) itself.

If $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$ is a parallel component of an s-p graph (V, R), then the *reduction of* (V, R) *by* $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$ is the graph $(\hat{V}, \hat{R})$ defined as

$$\hat{V} = (V - \bar{V}) \cup \{\bar{v}, \bar{v}'\},$$

$$\hat{R} = (R - \bar{R}) \cup \{(\bar{v}, \bar{v}')\}.$$

Intuitively, $(\hat{V}, \hat{R})$ is obtained from (V, R) by replacing $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$ by the vertices $\bar{v}$ and $\bar{v}'$ and an edge from $\bar{v}$ to $\bar{v}'$. The reduction of the s-p graph of Figure 12 by the first of its depicted parallel components is shown in Figure 13(a). The reduction of the resulting graph by its unique parallel component is shown in Figure 13(b).

In [2], an *admissible* graph is defined as a connected two-terminal graph such that, for any edge $\{x, y\}$, either $(x, y) \in R$ or $(y, x) \in R$ where R is the link relation. Theorem 4.12 of [2] states that a two-terminal graph G is series-parallel if and only if G is admissible and its link relation is asymmetric. Using this result, it is easily shown that the reduction of an s-p graph by one of its parallel components is an s-p graph.

We now present the main contribution of this paper.

THEOREM. *Let* $(\hat{V}, \hat{R})$ *denote the reduction of an s-p graph* (V, R) *by a proper parallel component* $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$. *In* $(\hat{V}, \hat{R})$ *define* $c(\bar{v}, \bar{v}') = C^*(\bar{V}, \bar{R})$, *and let the*

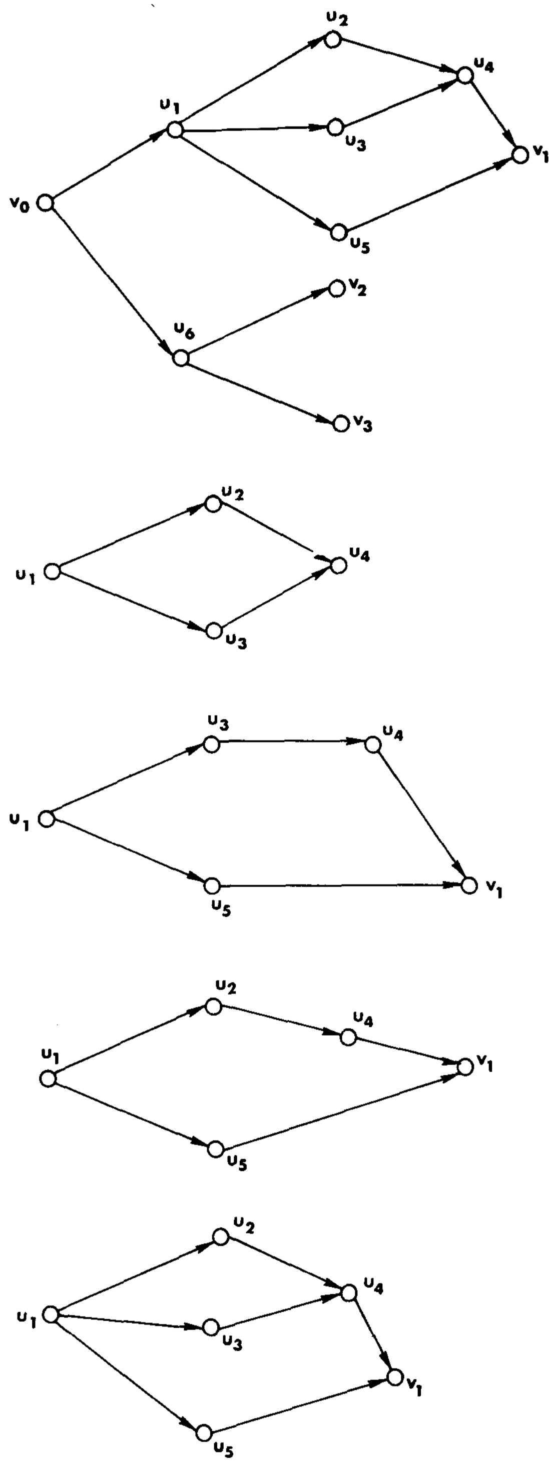

FIG. 12. An s-p graph and all its parallel components.

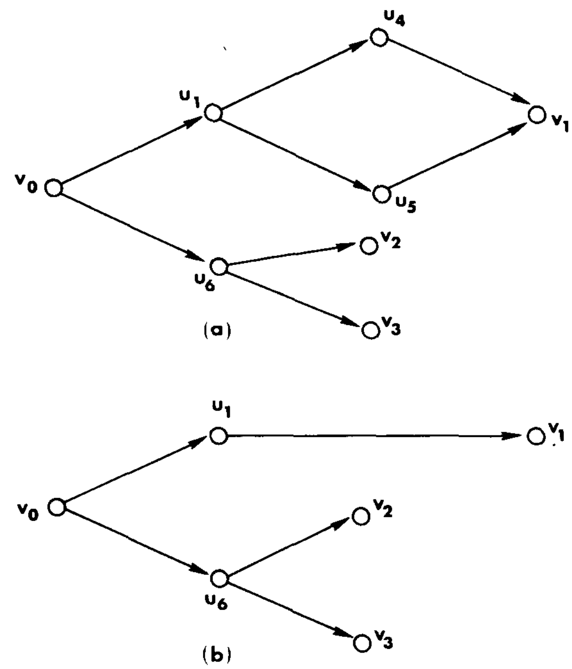

FIG. 13. The reduction of an s-p graph by one of its parallel components.

costs of all other edges remain unchanged. If C is s-p decomposable, then $C^*(V, R) = C^*(\hat{V}, \hat{R})$.

PROOF. The proof will be accomplished by induction on the size of R. If $|R| = 1$, then the result is vacuously true because (V, R) contains no parallel components. Assume that $|R| = N$ and that the theorem is true for $|R| < N$. Since $(\hat{V}, \hat{R})$ is an s-p graph, and since $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$ is a proper parallel component of (V, R), it follows that $(\hat{V}, \hat{R})$ is not a unit graph but is the composition of s-p graphs $(\hat{V}_1, \hat{R}_1)$ and $(\hat{V}_2, \hat{R}_2)$. Without loss of generality it may be assumed that the edge $(\bar{v}, \bar{v}')$ is contained in $(\hat{V}_1, \hat{R}_1)$. If (V_1, R_1) is the graph obtained from $(\hat{V}_1, \hat{R}_1)$ by replacing $(\bar{v}, \bar{v}')$ with $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$, then it is easily seen that $(\hat{V}_1, \hat{R})$ is the reduction of (V_1, R_1) by $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$ and that (V, R) is the composition of (V_1, R_1) and $(\hat{V}_2, \hat{R}_2)$. Since $|R_1| < N$, it follows from the induction hypothesis that $C^*(\hat{V}_1, \hat{R}_1) = C^*(V_1, R_1)$, where in $(\hat{V}_1, \hat{R}_1)$ we define $c(\bar{v}, \bar{v}') = C^*(\bar{V}, \bar{R})$. It follows from the s-p decomposability of C and from Lemma 7 that the cost of an optimal solution to the composition of $(\hat{V}_1, \hat{R}_1)$ and $(\hat{V}_2, \hat{R}_2)$ equals the cost of an optimal solution to the composition of (V_1, R_1) and $(\hat{V}_2, \hat{R}_2)$; that is, $C^*(\hat{V}, \hat{R}) = C^*(V, R)$. Q.E.D.

The above theorem can be utilized in the following manner. Given an s-p graph (V, R), select any proper parallel component $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$ and construct the reduction of (V, R) by $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$. The edge $(\bar{v}, \bar{v}')$ in the reduced graph is then labeled by the cost of an optimal solution to $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$. If the parallel component is "minimal" in the sense that it properly contains no parallel component, an optimal solution to $(\bar{V}, \bar{R}, \bar{v}, \bar{v}')$ can be easily obtained by inspection. The procedure is then repeated on the reduced graph, etc. This procedure will terminate in a finite number of steps with an s-p graph having no proper parallel component. The resulting

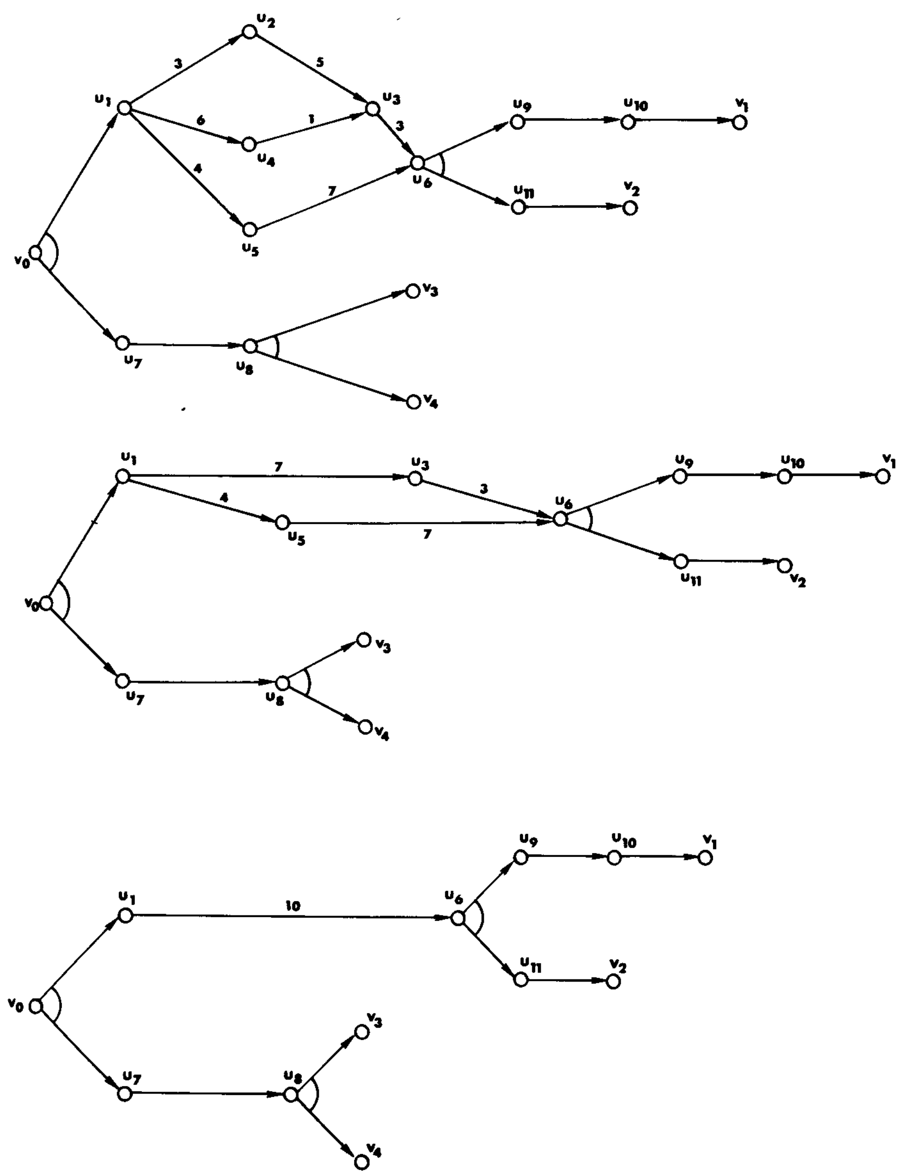

FIG. 14. The reduction of an s-p graph to a directed tree.

graph is thus either a directed tree or is itself a minimal parallel component. An example illustrating this procedure is given in Figure 14.

If the reduced graph is a directed tree, any available technique to efficiently search for an optimal solution to a directed AND/OR tree can be applied. In this paper, we shall mention two approaches: the dynamic programming approach [1] and the branch-and-bound approach [4]. For a more comprehensive discussion of tree searching techniques, see [7].

The essential concept of the dynamic programming approach is backward recursion. Applied to AND-OR tree searching, this means that we can work backwardly, starting with the terminal vertices. In this way, we can associate an optimal

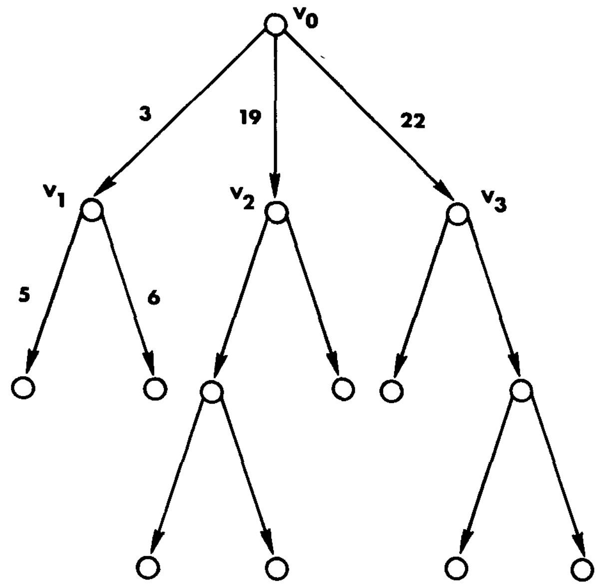

FIG. 15

cost with every vertex v of the tree (V, R), including the root v_0. The basic equations governing the calculation of the costs are as follows:

(1) $L(v) = 0$ if v is a terminal vertex.
(2) Otherwise,
Case 1: $v \in A$.
(1a): Sum-cost
$L(v) = \sum_{v' \in T(v)} (c(v, v') + L(v'))$.
(1b): Max-cost
$L(v) = \max_{v' \in T(v)} \{c(v, v') + L(v')\}$.
Case 2: $v \in 0$.
For both sum-cost and max-cost,
$L(v) = \min_{v' \in T(v)} \{c(v, v') + L(v')\}$.

For every $v \in 0$, let $h(v)$ be the element of $T(v)$ at which L is a minimum. We then can find an optimal solution (X, B) to the directed tree by the following recursive rules:

(1) $v_0 \in X$;
(2) $v \in X \cap A$ implies that $T(v) \subset X$ and $(v, v') \in B$ for each $v' \in T(v)$;
(3) $v \in X \cap 0$ implies that $X \cap T(v) = \{h(v)\}$ and $(v, h(v)) \in B$.

It is easy to show that (X, B) is an optimal solution and that its cost is $L(v_0)$. That this backward recursion approach is indeed efficient is elegantly discussed by Nemhauser [5].

The above method can be further improved by adopting a branch-and-bound approach [4]. Consider Figure 15, for example. It is obvious that one does not have to probe the tree below v_2 because $c(v, v_2) = 19 > c(v, v_1) + L(v_1)$. Similarly, one does not have to probe the tree below v_3 either. For a discussion of this kind of cutoff, consult [9].

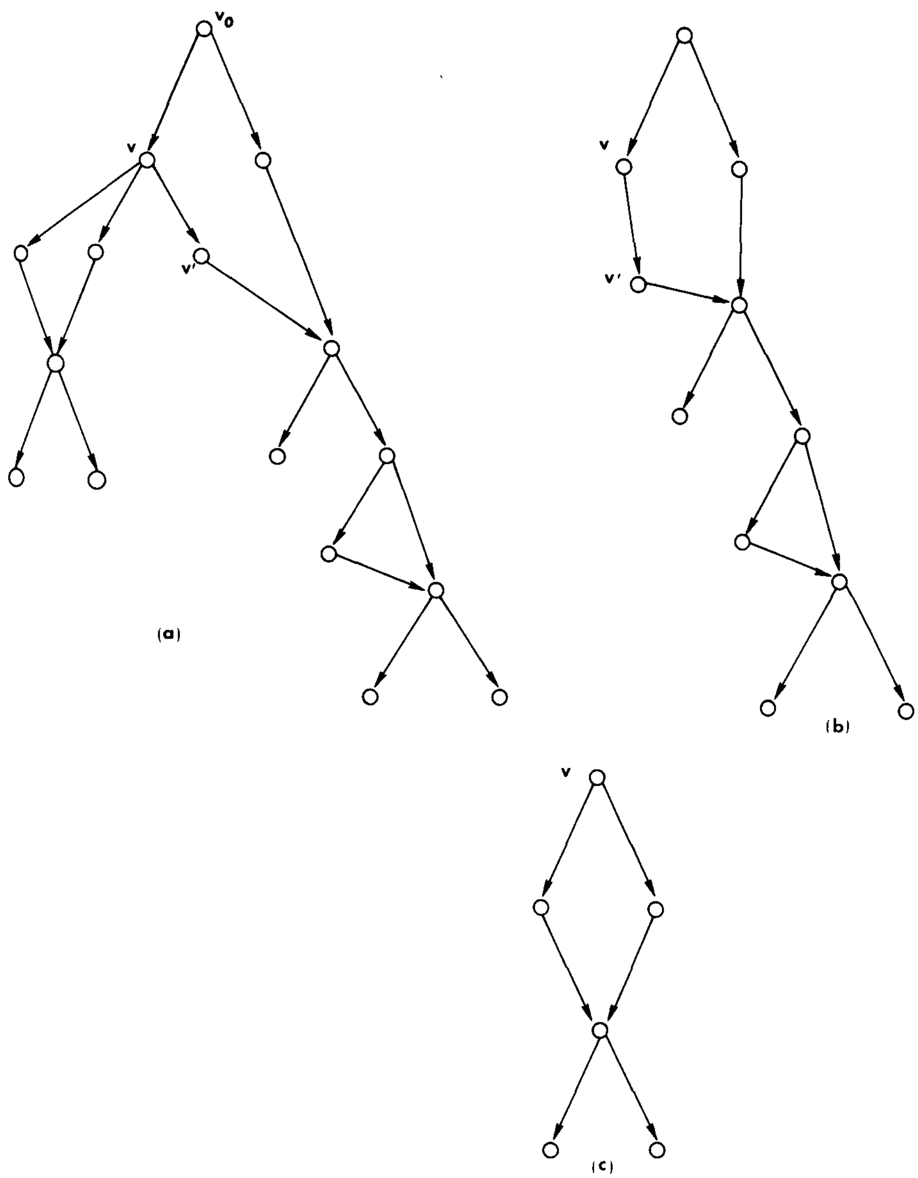

FIG. 16

4. Hyper-Series-Parallel Graphs

The reduction technique described in Section 3 is applicable to some non-s-p graphs. Consider, for example, the graph shown in Figure 16(a). This graph consists of the s-p graph shown in Figure 16(b) linked with the s-p graph shown in Figure 16(c). If vertex v is an "and" vertex, then the graph of Figure 16(a) is equivalent to the s-p graph of Figure 16(b) if the cost of the edge (v, v') is redefined as $c(v, v') = c(v, v') + C^*[16(c)]$, where $C^*[16(c)]$ denotes the cost of an optimal solution to the graph shown in Figure 16(c). Thus, one must first find an optimal solution to

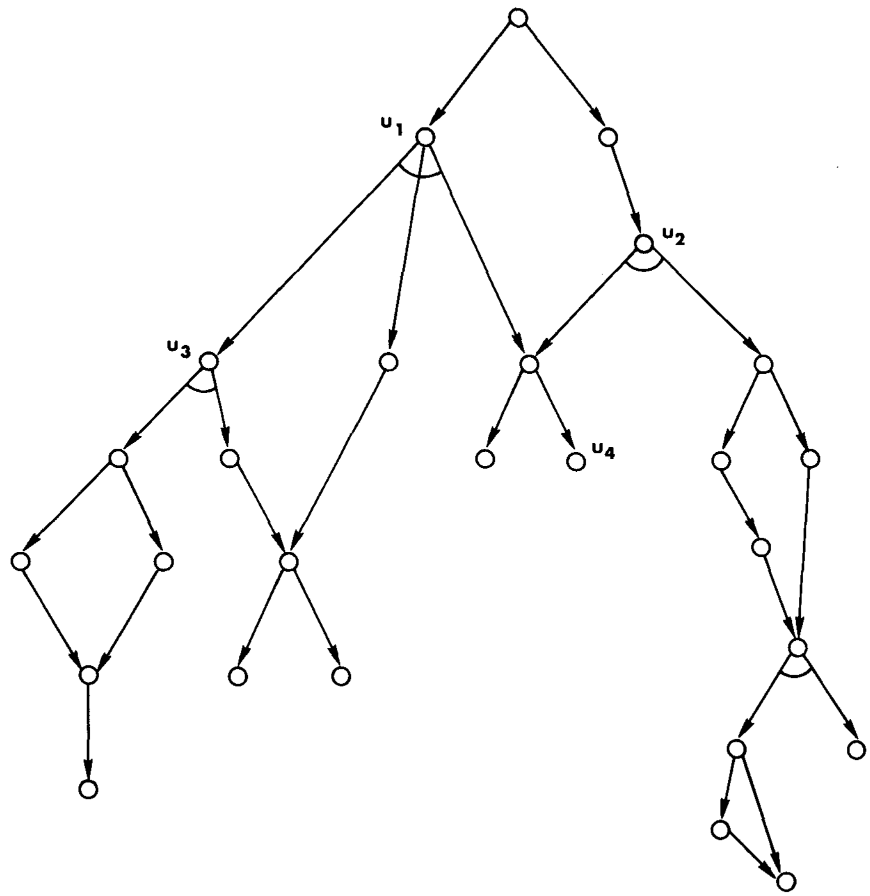

FIG. 17. A hyper-s-p graph.

the s-p graph of 16(c), and then find an optimal solution to the s-p graph of 16(b) with $c(v, v')$ redefined as above. Each of these steps involves utilizing the reduction algorithm described in Section 3. Quite complex non-s-p graphs can be treated in this manner.

A directed graph (V, R, v_1) with root v_1 is the *composition* of a directed graph (V_1, R_1, v_1) with unique root v_1 and a directed graph (V_2, R_2, v_2) with unique root v_2, if $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{v_2\}$, and $R = R_1 \cup R_2$. The vertex v_2 is called the *vertex of connection* of (V_1, R_1, v_1) and of (V_2, R_2, v_2). The class of *hyper-series-parallel* (*hyper-s-p*) *graphs* is the smallest class containing the s-p graphs and having the property that the composition of any two graphs in the class is again in the class.

A hyper-s-p graph can thus be decomposed into s-p graphs. An example of a hyper-s-p graph is given in Figure 17. If each vertex of connection of a decomposition of a hyper-s-p graph is an "and" vertex, then an optimal solution for the hyper-s-p graph can be obtained as follows:

(a) Identify the "minimal" components as those s-p graphs whose only vertex of connection is its root.

(b) Find an optimal solution for each minimal component, using the reduction procedure of Section 3.

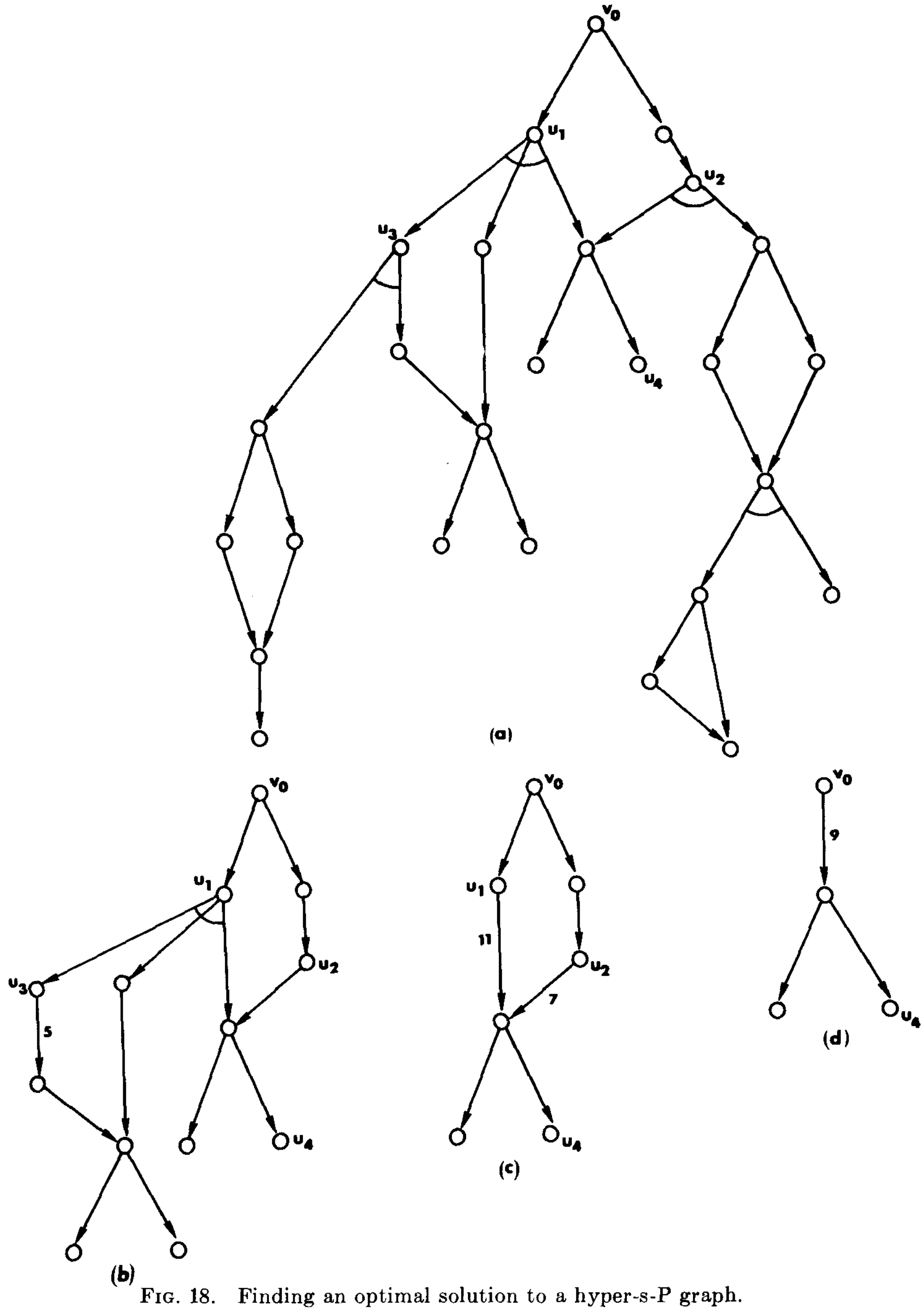

FIG. 18. Finding an optimal solution to a hyper-s-P graph.

(c) If (V, R, v) is a minimal component with root v, and if $v' \in T(v)$ but v' is not in V, then delete the minimal component (except for v) and relabel $c(v, v') = c(v, v') + C^*(V, R, v)$. Do this for each minimal component.

(d) If the resulting graph is not an s-p graph, then return to step (a); otherwise, apply the reduction procedure of Section 3 to the resulting s-p graph.

For the graph of Figure 17, this algorithm is illustrated in Figure 18. It is as-ımed that each edge cost is initially unity and that the sum-cost is to be used.

REFERENCES

1. Bellman, R. *Dynamic Programming*. Princeton U. Press, Princeton, N. J., 1957.
2. Elgot, C. C., and Wright, J. E. Series-parallel graphs and lattices. *Duke Math. J. 26*, (June 1959), 325–338.
3. Hart, P. E., Nilsson, N. J., and Raphael, B. A formal basis for the heuristic determination of minimum cost functions. *IEEE Trans. System Sci. Cybernetics SSC-4, No. 2* (July 1968), 100–107.
4. Lawler, E. L., and Wood, D. E. Branch-and-bound methods: a survey. *Operations Research, 14* (1966), 699–719.
5. Nemhauser, G. L. *Introduction to Dynamic Programming*. Wiley, New York, 1967.
6. Nilsson, N. J. Searching problem-solving and game-playing trees for minimal cost solutions. *IFIP Congress 68, Booklet H: Applications 3*, H125–H130.
7. Nilsson, N. J. *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill, New York, 1971 (in press).
8. Slagle, J. R., and Bursky, P. Experiments with a multipurpose theorem-proving heuristic program. *J. ACM 15*, 1 (Jan. 1968), 85–99.
9. Slagle, J. R., and Dixon, J. K. Experiments with some programs that search game trees. *J. ACM 16*, 2 (April 1969), 189–207.

RECEIVED OCTOBER 1970; REVISED JANUARY 1971.