CUTSETS AND PARTITIONS OF HYPERGRAPHS

by

Eugene L. Lawler

Memorandum No. ERL-M340

1 May 1972

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

CUTSETS AND PARTITIONS OF HYPERGRAPHS

Eugene L. Lawler

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory,
University of California, Berkeley, California 94720

ABSTRACT

A hypergraph is a combinatorial structure with nodes and arcs, similar to an ordinary "linear" graph, except that arcs are incident to arbitrary subsets of nodes, instead of pairs of nodes. Cutsets of hypergraphs are defined in a natural way, and it is shown that optimal cutsets can be found by means of a network flow computation. The optimal cutset computation can be used to generate a family of subsets of nodes, which we call LS sets. Intuitively, an LS set is a subset of nodes that are more strongly connected to each other than to nodes in the complementary set. LS sets are useful for constructing optimal or near-optimal partitions of the nodes. A polynomial-bounded partitioning algorithm is presented, and various applications are suggested.

Research sponsored by the Naval Electronic Systems Command, Contract N00039-71-C-0255.

1. Introduction

A hypergraph [1] is a combinatorial structure with nodes and arcs, similar to an ordinary "linear" graph, except that arcs are incident to arbitrary subsets of nodes instead of pairs of nodes. Thus a hypergraph with n nodes may have as many as 2^n distinct arcs.

Many of the definitions and concepts of graph theory apply to hypergraphs without any particular difficulty. For example, let $G = (N,A)$ be a hypergraph, and $S \subseteq N$ be a subset of nodes. Let $A(S)$ denote the set of all arcs that are incident to one or more nodes in S. Then S is said to be a component of G if $A(S) \cap A(N-S)$ is empty and S is maximal with respect to this property. Let s, t be distinct nodes of G. Then $C \subseteq A$ is said to be an (s,t) cutset if there exists a subset $S \subseteq N$, where $s \in S$, $t \in N-S$, such that $C = A(S) \cap A(N-S)$.

Suppose $w: A \rightarrow R^+$ is a weighting of the arcs of G with positive real numbers. For a given subset $C \subseteq A$, let $w(C)$ denote the sum of the weights of the arcs in C. For a given subset $S \subseteq N$, let $\bar{w}(S) = w(A(S) \cap A(N-S))$. I.e. $\bar{w}(S)$ is the sum of the weights of the arcs disconnecting or cutting S from N-S.

The first problem we consider is that of finding, for given s and t, a minimum-weight (s,t) cutset. This can be formulated and solved as a conventional network flow problem.

The second problem we consider is that of identifying all the "LS sets" of a weighted hypergraph. A subset $S \subseteq N$ is said to be an LS set (after Luccio and Sami [5], who called them "minimal groups") if, for all proper subsets $T \subset S$, it is the case that $\bar{w}(T) > \bar{w}(S)$. Intuitively, an LS set is a subset of nodes that are more strongly connected to each other than to nodes in the complementary set. We develop a computational procedure for identifying all the LS sets of a hypergraph, using the minimum-weight cutset algorithm as a subroutine. This procedure is efficient, in the sense that the number of computational steps is bounded by a polynomial function of the number of nodes and arcs.

We next consider the problem of optimally partitioning the nodes of a hypergraph. Various optimality criteria are formulated and it is shown that, for certain of these criteria, there exist optimal or near-optimal partitions in which each block is an LS set. A simple and efficient algorithm is presented for obtaining optimal partitions into LS sets.

Hypergraphs are useful models for a variety of structures. Some of these are indicated, and possible applications of the partitioning algorithm are discussed.

2. Minimum-Weight Cutsets

Let $G = (N,A)$ be a given hypergraph, $w\colon\ A \to R^+$ be a weighting of its arcs with positive real numbers, and let s, t be distinct nodes. A minimum-weight (s,t) cutset in G is identified with a minimum-capacity cutset in a flow network G' constructed as follows.

The flow network G' has one node i for each node i of G and two nodes, j' and j", for each arc a_j of G. There is an arc from j' to j" in G' and its capacity is set equal to the weight of arc a_j in G. If arc a_j is incident to node i in G, G' has arcs with infinite capacity from i to j' and from j" to i in G'. These relationships are illustrated in Figure 1. Numbers on arcs in G' indicate capacities.

<u>Theorem 2.1</u>

There is a one-one correspondence between minimum-weight (s,t) cutsets of the hypergraph G and minimum-capacity (s,t) cutsets of the flow network G'. Arc j of an (s,t) cutset of G is identified with arc (j', j'') of the corresponding (s,t) cutset of G'.

Proof: Follows immediately from the construction of G'.

A minimum-capacity (s,t) cutset of G' can be found by a standard network flow computation. It is well-known that such a cutset is the dual of a maximum value (s,t) flow [3].

We can establish an upper bound on the length of the maximum flow computation, as follows. Suppose the hypergraph G has n nodes and m arcs. The structure of G' is such that each augmenting path can be assumed to contain no more than $3(n-1)$ arcs. Suppose for each augmentation we choose the shortest available augmenting path. Then it follows from the reasoning of Edmonds and Karp [2] that no more than $O(mn)$ augmentations are required.

Each augmentation requires the execution of a "labelling" routine. When the label of one of the n nodes i is "scanned," as many as m nodes j' may have to be labelled as a result. The scanning of any one of the m nodes j' requires fixed effort. The scanning of any one of the m nodes j'' may require the labelling of as many as n nodes i. We conclude that the labelling routine requires $O(mn)$ computational steps.

Since there are $O(mn)$ augmentations, and each augmentation requires $O(mn)$ steps, it follows that the overall computation is $O(m^2n^2)$ in length.

We note that the flow computation does not actually have to be carried out over the network G'. I.e. a specialized procedure could be designed to operate "directly" on the hypergraph G. The design of such an algorithm is, however, a relatively straightforward matter, and we do not deal with it here.

Finally we note that the cutset computation of the previous section can be adapted to find a minimum-weight cutset separating X and Y, where X and Y are arbitrary disjoint subsets of nodes. We construct the flow network G' as before, except with a dummy source node s and dummy sink node t. Source s is connected to each node in X by an arc of infinite capacity and each node in Y is connected to t by an arc of infinite capacity. Then a minimum-capacity (s,t) cutset in G' corresponds to a minimum-weight cutset separating X and Y in G.

3. LS Sets

We recall the definition of an LS set from Section 1. A subset $S \subseteq N$ is said to be an LS set if, for all proper subsets $T \subset S$, it is the case that $\bar{w}(T) > \bar{w}(S)$. (By "proper", we mean that $T \neq S$ and $T \neq \phi$.)

Note that all singleton sets of nodes are LS sets. Moreover, if G is connected then N is itself an LS set. Thus every hypergraph has at least n, and possibly n+1, trivial LS sets.

Theorem 3.1

Let S be an arbitrary subset of nodes and T be a proper subset of S for which $\bar{w}(T)$ is minimal. Then any LS set properly contained in S is contained either in T or in S-T.

Proof:

Suppose there were to exist an LS set $U \subset S$, where $U \not\subseteq T$ and $U \not\subseteq S-T$. The sets S, T, U induce a partition of the nodes of the hypergraph into five subsets as follows:

$$N_1 = N-S,$$

$$N_2 = S - (T \cup U),$$

$$N_3 = T-U,$$

$$N_4 = T \cap U,$$

$$N_5 = U-T.$$

In turn, the sets $N_1, N_2, \cdots, N_5$ induce a partition of the arcs into 31 subsets, as follows:

$$A_1 = A'(N_1) \cap A'(N_2) \cap A'(N_3) \cap A'(N_4) \cap A(N_5)$$

$$A_2 = A'(N_1) \cap A'(N_2) \cap A'(N_3) \cap A(N_4) \cap A'(N_5),$$

$$\vdots$$

$$A_{31} = A(N_1) \cap A(N_2) \cap A(N_3) \cap A(N_4) \cap A(N_5),$$

where $A'(N_i)$ denotes the complement of $A(N_i)$, i.e. $A'(N_i) = A-A(N_i)$.

Because T is such that $\overline{w}(T)$ is minimal,

$$\overline{w}(T-U) \geq \overline{w}(T). \tag{3.1}$$

Because U is an LS set and U-T is a proper subset of T (by assumption $U \not\subseteq T$, $U \not\subseteq S-T$), (3.2) $$\overline{w}(U-T) > \overline{w}(U).$$

It is possible to represent $\overline{w}(T-U)$, $\overline{w}(T)$, $\overline{w}(U-T)$, $\overline{w}(U)$ in terms of $w(A_i)$, $i = 1, 2, \cdots, 31$. For example,

$$\begin{aligned}\overline{w}(T-U) = w(A_1) &+ w(A_6) + w(A_7) + w(A_{12}) \\ &+ w(A_{13}) + w(A_{14}) + w(A_{15}) \\ &+ w(A_{20}) + w(A_{21}) + w(A_{22}) \\ &+ w(A_{23}) + w(A_{28}) + w(A_{29}) \\ &+ w(A_{30}) + w(A_{31}).\end{aligned}$$

It can then be shown that (3.1) implies that $w(A_6) \geq w(A_3)$, and that (3.2) implies that $w(A_3) > w(A_6)$. This is a contradiction, from which it follows that the LS set U cannot exist.

Corollary 3.2 (Luccio-Sami [5]).

LS sets do not partially overlap. I.e. if S_1 and S_2 are LS sets, then either $S_1 \cap S_2 = \phi$ or $S_1 \subseteq S_2$ or $S_2 \subseteq S_1$.

Proof: Let S_1, S_2 be LS sets and apply the theorem to $S = S_1 \cup S_2$.

Note: Luccio and Sami proved this corollary as their main theorem, in the specialized form in which the arcs have equal weights.

Corollary 3.3

In a hypergraph with n nodes, there are at most 2n-1 LS sets.

Proof: Let T be a proper subset of N for which $\overline{w}(T)$ is minimal, and suppose $|T| = p$. Then, by inductive hypothesis, there are at most 2p-1 LS sets in T and 2(n-p)-1 LS sets in N-T. These exhaust the LS sets, except for N itself. Hence there are at most $(2p-1) + (2(n-p)-1) + 1 = 2n-1$ LS sets in all.

4. Identification of LS Sets

In their paper [5] Luccio and Sami proposed a relatively inefficient (non-polynomial-bounded) procedure for identifying LS sets. Theorem 3.1 suggests an alternative approach to the problem of identifying all the LS sets of a given hypergraph. That is, start with $S = N$ and find a proper subset $T \subset S$ for which $\overline{w}(T)$ is minimal. Then repeat, with T and N-T in the role of S. Continue to separate subsets into smaller and smaller subsets until only singleton subsets remain. The collection of all subsets thus generated contains all the LS sets. Those sets S in the collection which are not LS sets (but are unions of LS sets) are easily identified, since S is an LS set if and only if $\overline{w}(T) > \overline{w}(S)$.

Now let us consider how to identify a proper subset $T \subset S$ for which $\overline{w}(T)$ is minimal. Let x and y be nodes in S. Let $X = \{x\}$ and $Y = N-S \cup \{y\}$. Then a minimum-weight cutset separating X and Y determines a proper subset $T \subset S$ for which $\overline{w}(T)$ is minimal, subject to the constraint that $x \in T$ and $y \in S-T$.

Without loss of generality, let $S = \{1,2,\cdots,p\}$. Suppose we compute minimum-weight cutsets separating X and Y, for $x = 1$ and $y = 2, 3, \cdots, p$ and also for $y = 1$ and $x = 2, 3, \cdots, p$. Then we shall have exhausted all possibilities for proper subsets $T \subset S$. I.e. the minimum of the minimum-weight cutsets determined for the $2(p-1)$ X, Y pairs determines

T for which $\overline{w}(T)$ is minimal.

Thus the computation of T requires $O(n)$ network flow computations. There are $O(n)$ computations of T to be performed, in order to identify all LS sets. Each network flow computation is $O(m^2n^2)$ in length. Thus the overall computation of the LS sets is $O(m^2n^4)$ in length.

As an example, of the LS-set computation, consider the hypergraph with 5 nodes and 6 arcs defined by the incidence matrix below:

$$\begin{array}{cc}
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} &
\begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0
\end{bmatrix} \\
& \begin{array}{cccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{array}
\end{array}$$

Let $w(a_i)=1$, $i=1,2,\ldots,5$, and $w(a_6)=4$. The result of the computation is indicated in Figure 2. A set S is identified with each node of the binary tree. One branch below each node is directed to T and the other to S-T. The values of $\overline{w}(T)$ and $\overline{w}(S-T)$ are indicated in parentheses. All sets generated are LS sets, except {1,2}.

5. Optimal Partitions

Consider the following optimization problem. Given an arc-weighted hypergraph G and an integer k, find an optimal partition of the nodes of G into k blocks $S_1, S_2, \cdots, S_k$. The objective of the optimization may be to minimize

(5.1) $$\sum_{i=1}^{k} \overline{w}(S_i)$$

or to minimize

(5.2) $$\max_{i} \{\overline{w}(S_i)\}$$

or to lexicographically minimize the sequence

(5.3) $$(\overline{w}(S_1), \overline{w}(S_2), \cdots, \overline{w}(S_k)).$$

We refer to these optimization criteria as <u>min-sum</u>, <u>min-max</u> and <u>min-lexicographic</u> respectively.

We are unable to propose a good algorithm for any one of these optimization criteria. However, we can solve the following type of problem. Find a partition of the nodes of G into <u>at least</u> k blocks $S_1, S_2, \cdots, S_p$, $(p \geq k)$, such that (5.1), (5.2) or (5.3) is minimized. (Thus, the value of the objective function depends upon the k blocks S_i for which $\overline{w}(S_i)$ is smallest.) We refer to a solution to this type of problem as a <u>weakly optimal</u> partition.

<u>Theorem 4.1</u>

For any G and any k, there exists a weakly optimal partition in which each block is an LS set.

Proof:

Suppose $S_1, S_2, \cdots, S_p$ is a weakly optimal partition, and S_i is not an LS set. Split S_i into two blocks S_i' and $S_i - S_i'$ where S_i' is an

LS set and $\bar{w}(S_i') \leq \bar{w}(S_i)$. Continue to split blocks in this way until a weakly optimal solution is obtained in which each block is an LS set.

The following simple recursive procedure solves the weak optimization problem for the min-sum criterion (5.1). Let

$W(S,k)$ = the minimum weight of a partition of S into at least k blocks.

Clearly, if S is a singleton set,

$$(5.4) \qquad W(S,k) = \begin{cases} 0 & \text{, if } k = 0 \\ \bar{w}(S), & \text{if } k = 1 \\ +\infty & \text{, if } k > 1 \end{cases}$$

If S is not a singleton set, and T is a proper subset of S for which $\bar{w}(T)$ is minimal (as determined by the LS-set computation),

$$(5.5) \qquad W(S,k) = \begin{cases} 0, & \text{if } k = 0 \\ \min \{\bar{w}(S), \bar{w}(T)\}, & \text{if } k = 1 \\ \min\limits_{0 \leq k' \leq k} \{W(T,k') + W(S-T,k-k')\}, & \text{if } k > 1. \end{cases}$$

For the min-max criterion (5.2), equation (5.5) becomes

$$(5.6) \qquad W(S,k) = \begin{cases} 0, & \text{if } k = 0 \\ \min \{\bar{w}(S), \bar{w}(T)\}, & \text{if } k = 1 \\ \min\limits_{0 \leq k' \leq k} \max \{W(T,k'), W(S-T,k-k')\}, & \text{if } k > 1 \end{cases}$$

The recursion for the lexicographic minimum criterion (5.3) is formally similar to (5.5), except that W(S,k) denotes an ordered sequence, and the

"+" operation in (5.5) is replaced by a sequence-merging operation.

It is possible to employ essentially the same type of recursion to compute an optimal partition, subject to the constraint that each block is one of the sets in the collection $\mathcal{S}$ generated by the LS-set identification procedure of the previous section. Let

$$W^*(S,k) = \text{the minimum weight of a partition of } S \text{ into exactly } k \text{ blocks drawn from } \mathcal{S}.$$

If S is a singleton set,

$$(5.7) \qquad W^*(S,k) = \begin{cases} \overline{w}(S), & \text{if } k = 1, \\ +\infty, & \text{if } k > 1. \end{cases}$$

If S is not a singleton set, and T is a proper subset for which $\overline{w}(T)$ is minimal,

$$(5.8) \qquad W^*(S,k) = \begin{cases} \overline{w}(S), & \text{if } k = 1 \\ \min\limits_{1 \leq k' \leq k-1} \{W^*(T,k') + W^*(S-T,k-k')\}. & \text{if } k > 1. \end{cases}$$

Similar recursion equations can, of course, be formulated for the min-max and min-lexicographic criteria.

We can estimate the complexity of the computation of both W(N,k) and $W^*(N,k)$ as follows. There are 2n-1 subsets S and at most n values of k for which (5.4) or (5.5) must be solved. Each equation in the set (5.5) involves minimization over at most n alternatives. Thus the overall computation is $O(n^3)$.

Clearly W(N,k), the cost of a weakly optimal solution, is a lower bound on the cost of an optimal solution. And $W^*(N,k)$, the cost of a feasible solution, is an upper bound. If it happens that $W(N,k) = W^*(N,k)$, then an optimal solution has been found by the computation of $W^*(N,k)$.

As an example, if the partitioning algorithm is applied to the example in Section 4, the following results are obtained:

k	W(N,k)	$W^*(N,k)$	W^*-Partition
1	0	0	{1,2,3,4,5}
2	2	2	{1,2,3,4},{5}
3	5	6	{1,2},{3,4},{5}
4	7	7	{1},{2},{3,4},{5}
5	17	17	{1},{2},{3},{4},{5}

In two of the three nontrivial cases, i.e. k = 2 and k = 4, the partition determined by the W^*-computation is optimal because $W(N,k)= W^*(N,k)$. In the third case, k = 3, the computed partition can be shown to be optimal by an ad hoc examination of possibilities.

It is believed that for a variety of problems of interest, partitions of hypergraphs into LS sets and unions of LS sets will often be optimal, and nearly always quite close to optimal.

6. Applications

There are a number of contexts in which one may seek to partition the nodes of a hypergraph. Consider the following.

Network Analysis. For the purpose of analysis, it is desired to partition the nodes of a given network in such a way that the number of arcs extending between blocks of the partition is as small as possible.

In this case, the hypergraph is an ordinary linear graph, and the weight of each arc is unity.

Information Storage and Retrieval. A collection of technical documents is indexed by "descriptors". It is desired to partition the documents into subcollections in such a way that each subcollection is concerned with a coherent technical area.

A hypergraph can be defined in which each node is identified with a document, and each arc is identified with a descriptor. Cf. reference [6].

Numerical Taxonomy. The presence or absence of certain attributes has been determined for each of several biological species. It is desired to find some objective basis upon which to partition the species (where a block of the partition is a phylum, order, genus, or whatever) in accordance with the observed data.

A hypergraph can be defined in which each node is identified with a species and each arc is identified with an attribute.

Packaging of Electronic Circuits. A number of digital circuit modules are to be grouped together into "packages". Each circuit module has identified with it (as inputs and outputs) various "signals". Electrical interconnections must be provided to transmit each signal between the modules with which it is identified. Interconnections between packages are less desirable than connections made between modules in the same package.

A hypergraph can be defined in which each node is identified with a circuit module and each arc is identified with a signal.

In a previous paper [4], the author considered the optimal partitioning problem in the context of electronic circuit packaging. The objective was to minimize the sum of the weights of the blocks of the partition, where the weights were assigned by an arbitrary function

$$\omega: \mathcal{P}(N) \to R^+.$$

It was shown that if ω is monotone, in the sense that $S \subseteq T$ implies $\omega(S) \leq \omega(T)$, then the optimal partitioning problem can be formulated as a covering problem. However, this does not provide a good solution method, except for relatively small problems.

A monotone weighting function of particular interest in the case of electronic circuit packaging is

$$\omega(S) = w(A(S)),$$

where, as before, w is the arc-weighting function. The effect of this weighting function ω can be achieved by adding an (n+1)st node to the hypergraph, and making all arcs incident to that node. Then, if node n+1 does not belong to S,

$$\overline{w}(S) = w(A(S)).$$

The partitioning procedure of Section 5 can then be applied to the (n+1)-node hypergraph, to obtain a weakly optimal partition of the original n nodes. However, such a weakly optimal solution will not necessarily be very close to optimal.

References

[1] C. Berge, Graphes et Hypergraphes, Dunod, Paus (1970).

[2] J. Edmonds and R. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," Journal Assoc. for Computing Mach., 19 (April 1972), 248-264.

[3] L. R. Ford and D. R. Fulkerson, Flows in Networks, Princeton University Press (1962).

[4] E. L. Lawler, "Electrical Assemblies with a Minimum Number of Interconnections," IEEE Trans. Electronic Computers, EC-11 (Feb. 1962) 781-782.

[5] F. Luccio and M. Sami, "On the Decomposition of Networks into Minimally Interconnected Networks," IEEE Trans. on Circuit Theory, CT-16 (May 1969) 184-188.

[6] G. Martella and M. G. Sami, "A Proposal for the Organization of Large Files in a Hierarchical Structure," Technical Report, Instituto di Elettotecnica ed Elettronica del Politecnico di Milano, Italy (Dec. 1971).

LIST OF FIGURES

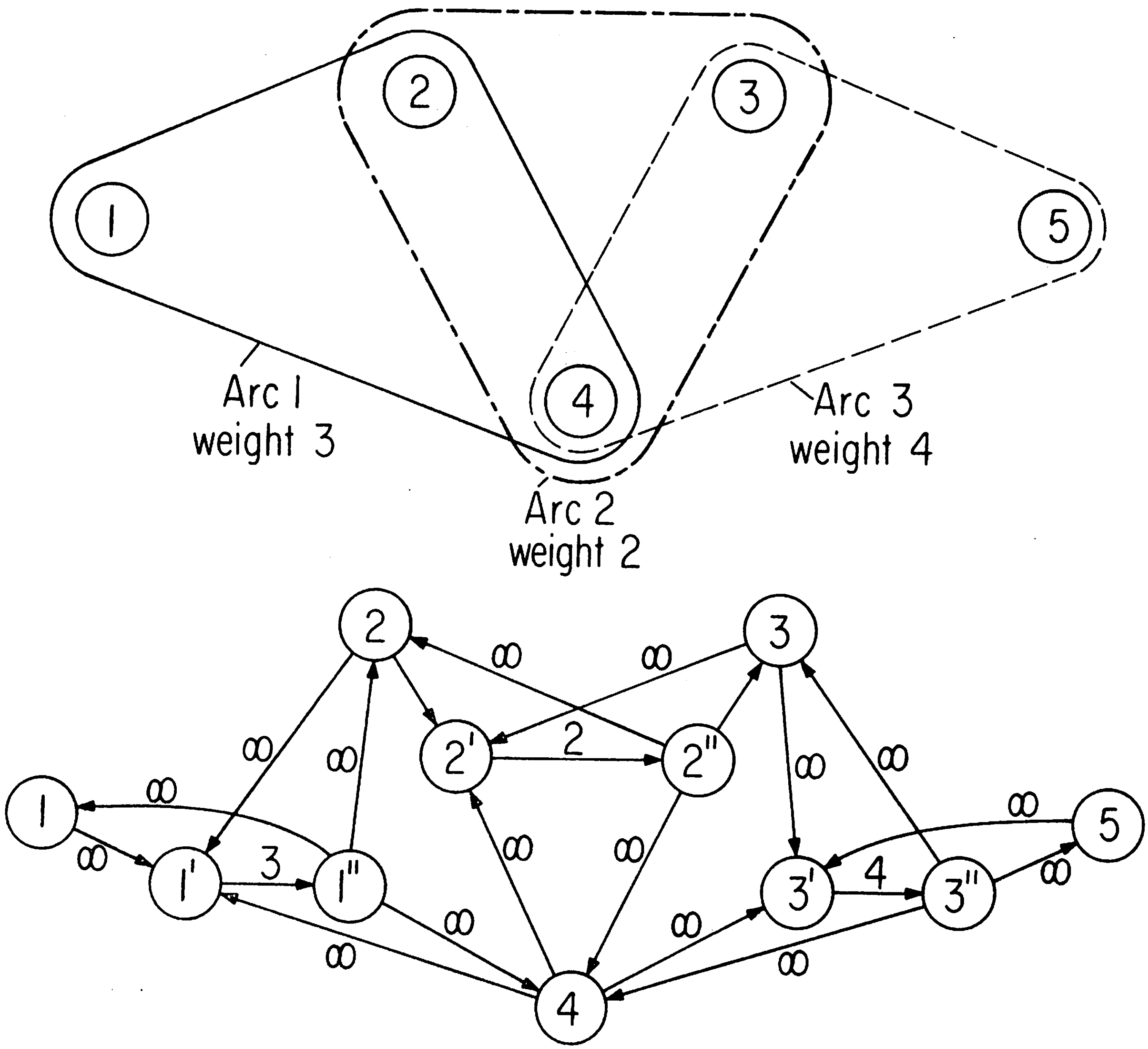
1
2
3
4
5
Arc 1
weight 3
Arc 2
weight 2
Arc 3
weight 4
2
3
1
2'
2''
1'
1''
3'
3''
5
4
∞
2
3
4

Figure 1

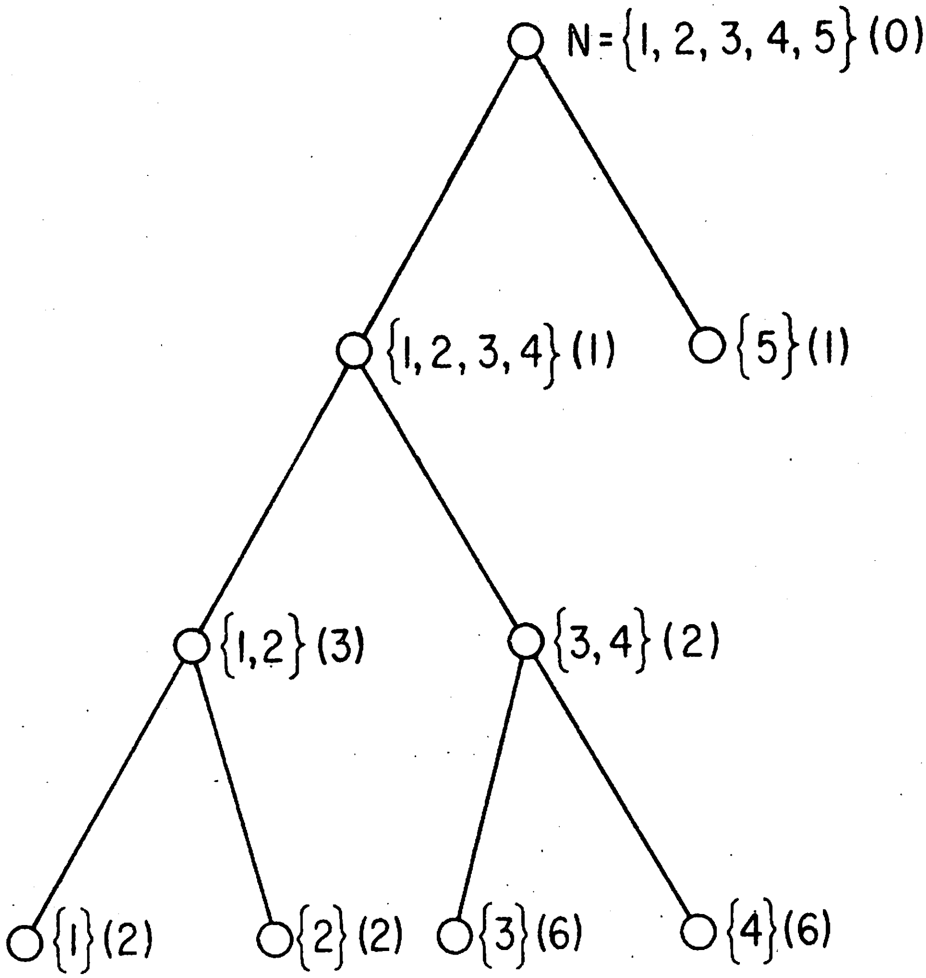

Figure 2