IBM Technical Disclosure Bulletin Vol. 15 No. 3 August 1972

ALGORITHMS FOR PARTITIONING OF GRAPHS AND COMPUTER LOGIC BASED ON EIGENVECTORS OF CONNECTION MATRICES

W. E. Donath and A. J. Hoffman

Partitioning of graphs and computer logic occurs in design automation of computers, packaging of computer programs, and classification problems.

The algorithm described here is a new method for partitioning. It uses as a starting point the eigenvectors of a modified connection matrix of a block graph in order to decide on a partitioning. Basic to the algorithm is the bound given in Theorem I.

Sections I and III describe basic mathematical concepts required for an understanding of the algorithms. Section IV describes the actual algorithm used for partitioning.

I. Basic Graph Theoretical Definitions

(A) A graph G is a set of vertices V and edges N; each edge (i, j) is considered to connect vertices i and j, where at least $i \varepsilon V$ or $j \varepsilon V$. If either $i \varepsilon V$ or $j \varepsilon V$, the edge is considered to be an "external" connection. In addition, a vertex i is considered to have an area a_i. Also, each edge (i,j) is considered to have capacity f_{ij}. In all cases considered here, it is always the case that $f_{ij} = f_{ji}$ -- i.e., the graph is symmetric. The normal situation is that f_{ij} is 1.

(B) A partition $\underset{\sim}{P}$ of a graph G is considered to be a division of all the nodes V into some disjoint subsets $P_1, P_2, \ldots, P_m$ such that for all $P_k \varepsilon \underset{\sim}{P}$

$$\sum_{i \varepsilon P_k} a_i \leq A_k \tag{1}$$

where a_k is some prespecified limit on the area of each subset, which is normall taken to be a constant. The number of terminals t_k of module k is defined as

$$t_k = \Sigma\{(i,j)\ \varepsilon N: (i \varepsilon P_k\ j \varepsilon P_k) \vee (i \varepsilon P_k \wedge j \varepsilon P_k)\}\ f_{ij} \tag{2}$$

where $\Sigma\ \{x\}\ f(x)$ denotes a sum restricted to the set $\{x\}$. Essentially, equation (2) denotes the "external flow" of group k.

Generally, $t_k \leq T_k$, the total terminal count for group K.

Also, one requires

$$\bigcup_{i=1}^{m} P_i = V \tag{3}$$

$$P_i \subset V.$$

(C) Computer logic is best defined as a "Net Graph", i.e., a Net Graph G consists of a set of vertices V and a set of nets N. Each net $n \varepsilon N$ is a subset of V and may (or may not) include a vertex not belonging to V, in which case it is an external (internal) net of G.

(D) A partition $\tilde{P}$ of a net graph G is a division of the nodes V into some disjoint subsets (or "modules") $P_1, P_2, \ldots, P_m$ and that for all $P_k \varepsilon \tilde{P}$

$$\sum_{i \varepsilon P_k} a_i \leq A_k \tag{4}$$

where A_k is some prespecified limit for each subset, and the terminal count t_k of each subset k is limited by

$$t_k \leq T_k. \tag{5}$$

Define $$t_k = \Sigma \{n: (N \cap P_k) \neq \phi \wedge (n - P_k) \neq \phi\} f_n , \tag{6}$$

i.e., the number of nets which have vertices both in P_k and outside of P_k.

(E) Define a graph "implementation" of a net graph in two steps:

(1) Let a tree τ on a set of vertices W be any graph whose vertices are W which is "connected" and has $|W| - 1$ edges; "connected" here means that if i_0 and j are nodes of W, there exists in τ some set of edges $(i_0 i_1)$ $(i_1 i_2)$ ... $(i_{\ell-1} i_\ell)$ $(i_\ell\ j)$ such that $i_k \varepsilon W$

$(K = 1, 2, \ldots \ell)$.

(2) Assume that for every net $n \varepsilon N$ we have a set of edges τ_n, which are a tree on n. Then an implementation $J(G)$ consists of V and $\bigcup_{n \varepsilon N} \tau_n$.

II. Connection Matrices

(A) Connection Matrices are matrices whose elements are defined for all pairs of nodes of V, so that they are of order $|V| \times |V|$. For a graph, the connection matrix C is defined as follows:

(1) Let all C_{ij} be initially 0

(2) For each edge (i,j), add f_{ij} to C_{ij} and C_{ji}.

(B) The "degree matrix" D for a connection matrix is defined as follows:

$$i \neq j \rightarrow D_{ij} = 0 \tag{7}$$

$$D_{ii} = \sum_j C_{ij}$$

(C) A "trace zero" matrix is any matrix for which

$$U_{ij} = 0 \quad \text{if } i \neq j \tag{8}$$

$$\sum_i U_{ii} = 0$$

(D) A connection matrix C of a net graph G is defined somewhat differently. Let

$$C_{ij} = \sum_{n \varepsilon N} g_{ij}^{(n)} \tag{9}$$

where $g_{ij}^{(n)}$ is nonzero only if $i \varepsilon n$ V and $j \varepsilon n$ V. It is of advantage to require of g_{ij} that for any subset $m \subset n$,

$$\sum_{i \varepsilon m} \sum_{j \varepsilon n-m} g_{ij}^{(n)} \leq f_n, \tag{10}$$

where f_n is some constant of the net, which may be fixed as 1 for any internal net n and 1/2 for any external net.

(E) A simple way of satisfying equation (10) is givn by letting

$$g_{ij}^{(n)} = \frac{4f_n}{|n|^2} \text{ if } |n| \text{ is even; } g_{ij}^{(n)} = \frac{4f_n}{|n|^2 - 1} \text{ if } |n| \text{ is odd} \quad (11)$$

(F) For $|n| \geq 4$, there may be some advantage in letting

$$g_{ij}^{(n)} = f_n \, h_i^{(n)} \, h_j^{(n)}$$

with

$$\sum_{i \varepsilon n} h_i^{(n)} = 2, \quad (12)$$

in order to satisfy equation (8).

Note: However C_{ij} is defined, the definitions of D (Section II-B) and U are still valid.

III. Eigenvalues and Eigenvectors

(A) In eigenvalue λ_r and eigenvector x_r of the matrix C - D + U

$(x_r = (x_{r1}, x_{r2}, \ldots, x_{r|v|}))$ satisfy the equation (13)

$$(C - D + U) \, x_r = \lambda_r A x_r,$$

where A is a diagonal matrix with elements a_i, and where an ordering is demanded on λ_i as

$$\lambda_1 \geq \lambda_2 \cdots \geq \lambda_{|v|} \quad (14)$$

Let

$$\Lambda_k = \lambda_1 + \lambda_2 + \cdots + \lambda_k . \quad (15)$$

Given the above definitions, the following is true.

Theorem I Given this partition of a graph or net graph into k equal sized subsets, then

$$\sum_{i=1}^{k} \tau_i \geq - |V| \Lambda_k/k. \tag{16}$$

For partitioning purposes, the set of the k highest eigenvalues is of interest. U is best varied to minimize Λ_k, where k is to a considerable extent arbitrary, but $k \geq 2$. If one is considering a net graph, one may use either technique IIE to determine $g_{ij}^{(n)}$ or, for some nets, technique IIF (or possibly other methods). If $g_{ij}^{(n)}$ is to be varied, it is best varied so as to minimize Λ_k. The general idea being that the higher the value of the bound is, the better will be the partition.

Assuming that the k_m eigenvectors for the k_m largest eigenvalues has been computed, it is then convenient to define a distance d_{ij} between any pair of nodes as

$$d_{ij}{}^2 = \sum_{r=1}^{k_m} (x_{ri} - x_{rj})^2 \tag{17}$$

The distance can be used as a criterion for giving an implementation of a net graph.

(B) A tree τ on the nodes of n is found, and also that

$$d_n^\alpha = \sum_{(i,j)\varepsilon\, \tau_n} d_{ij}^\alpha$$

is a minimum using the technique of minimal spanning tree. The tree τ_n is the same for all positive α, so one uses the most easily computed α, which is 2. Minimal spanning tree techniques are well known.

IV. Partitioning Algorithms

First, the manipulation of grouping (IVA) will be described; secondly, a grouping method (IVB) and a possible expansion (IVC) will be described; lastly, a "clean-up" algorithm will be given. The algorithms take advantage of the fact that for each edge of the graph (or the graph implementation of a net graph) a distance d_{ij} is given, where a short distance means that nodes i and j tend to favor association -- i.e., groups i and j like to be in the same grouping. It is also important to note that the constraints given in Section II-B for graphs and II-D for net graphs play a controlling role.

(A) "Elemental" groups are those which have only a single node. Initially, one creates for each node an elemental group. "Composite" groups are the union of two other groups. Each node i has just one group to which it belongs, which is denoted by g(i). If two groups g_1 and g_2 are "joined" to form g_3, then all nodes which belonged to groups g_1 and g_2 now belong to group g_3. g_1 and g_2 are "descendants" of g_3, while g_3 is the "parent" of g_1 and g_2.

Division of a group g_3, whose descendants are g_1 and g_2, means that groups g_1 and g_2 are "restored", i.e., any element that belonged to g_1 (or g_2) before g_1 and g_2 were joined to form g_3 now belong again to g_1 or(or g_2).

"Removal" of a group h from group g is somewhat more complex; let us say that g is the parent of g_1, which is the parent of g_2, etc., and finally g_j is the parent of h (see Figure 1); then in Figure 2 one can see that g_j is removed, while g_{j-1} now is the parent of g'_j and g'_{j-1}, while h is a separate group. All the elements of h now belong to h, also.

$$\begin{tikzcd} g \arrow[r] & g_1 \arrow[r] & g_2 \arrow[r] & \cdots \arrow[r] & g_{j-1} \arrow[r] \arrow[d] & g_j \arrow[r] \arrow[d] & h \\ & & & & g'_{j-1} & g'_j & \end{tikzcd}$$

Figure 1

$$\begin{tikzcd} g \arrow[r] & g_1 \arrow[r] & g_2 \arrow[r] & \cdots \arrow[r] & g_{j-1} \arrow[r] \arrow[d] & g'_j \arrow[r] & h \\ & & & & g'_{j-1} & & \end{tikzcd}$$

Figure 2

(B) Initially, there are only the elemental groups. The edges are sorted in order of increasing distance. A complete pass is then made through this list and for each edge (i,j) the following procedure is carried out:

(1) Groups g(i) and g(j) are determined.
(2) If the union of groups g(i) and g(j) does not violate any constraints, they are joined, and work is started on the next edge.
(3) If the union of groups g(i) and g(j) does not violate packaging constraints, either nothing is done and one considers the next edge in line, or a rearrangement may be attempted as suggested in (C). Alternatively, the following may be done:

a. Merging of groups may be stopped when the first violation of a constraint is observed.
b. When the edges reach some prespecified distance.

(C) An attempt is made to remove from the group g(i) (or g(j)) a subgroup k, which contains i (or j) and join it with g(j) (or g(i)). Criteria for such a move may be that such a move does not violate any constraints, that total terminal count is improved, and that one or the other of the groups is closer to the area limit.

(D) A clean-up algorithm is defined as follows:

Consider the biggest M groups (supposing it is desired to achieve M groups) and treat these as "basis" set; consider the remaining groups as "excess" set.

(1) The smallest group in the excess set is selected, and it is checked whether this group can be merged with any of the groups in basis set.
(2) If no basis group could be merged with the selected group, step 3 is executed. Else, it is merged with any of the basis groups it could be merged with and which also yields maximum pin saving, in case several merges were possible.
(3) If the selected group is not elemental, it is divided into its 2 descendant groups, and both groups are included in the excess set, and step 1 is executed. If the selected group is elemental, then the algorithm may be stopped as unsuccessful.

Alternatively, it may be advantageous to use the grouping derived to define a new graph and use other partitioning algorithms to derive the final grouping.