# VIAS ASSIGNMENT PROBLEM IN MULTILAYER PRINTED CIRCUIT BOARD

by

B. S. Ting, E. S. Kuh and A. Sangiovanni-Vincentelli

Memorandum No. UCB/ERL M77/13

21 March 1977

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

# Vias Assignment Problem in Multilayer Printed Circuit Board

B. S. Ting, E. S. Kuh and A. Sangiovanni-Vincentelli

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

## ABSTRACT

In the routing of a multilayer printed circuit board an important phase is the via assignments. In this paper, the via assignment problem is given a graph theoretic formulation. Some related optimization problems are proven to belong to a particular class of hard combinatorial problem: the class of NP-complete problems. This result suggests that the only way to solve efficiently the optimization problems is to introduce heuristic algorithms. Hence some heuristic algorithms are proposed and their performances are evaluated.

---

Research sponsored by the National Science Foundation Grant ENG74-06651-A01 and the Joint Services Electronics Program Contract F44620-C-0100.

## I. Introduction

With the advancement of integrated circuit technology, it is now possible to design and fabricate complex circuits within a small chip. Much more complex, large-scale electronic system can be designed using semiconductor chips as basic building blocks. Multilayer printed circuit (MPC) board is in general the vehicle and the medium for interconnecting those building blocks. The components blocks are mounted on top of the board where the terminals are inserted through the drilled-through holes, called pins. Connections among blocks are made by way of printed wires. Printed wires are either made by etching process or by additive process[1] on every layer of the board. In order to connect wires on different layers, plated-through holes, called vias, are provided for interlayer connections. The MPC routing problem consists of the determination of vias locations and physical routes of printed wires to satisfy the interconnection specifications of the circuits.

There are various kinds of physical constraints associated with the problem, typically: the size of the multilayer board, the feasible number of layers, the minimum width of the conductor path, and the necessary separation between two adjacent parallel conductor paths. Therefore, a key question is whether a given problem can indeed be realized with a specified multilayer board; and if it cannot be realized, whether there is an optimal realization for obtaining a maximum degree of interconnections. Or, the problem could be formulated in another way: for a given interconnection specification, what is an optimal design of multilayer board to facilitate one hundred percent routing?

In this paper, we make three basic assumptions along the approach proposed by.[2]

(i) The geometries of the pins and vias are at fixed locations. With standard size IC chips mounted on the board, this first assumptions does not impose much restriction on the problem structure.

(ii) Only points (pins and/or vias) on the same line (row or column) can be connected directly and the physical routes must be confined within the channels on both sides of the line.

(iii) Connections for points on a row are in one layer and connections for points on a column are in another layer.

In this connection, So has made an important contribution. By using the above three assumptions, the multilayer problem can be essentially reduced to several single-line, single-layer problems.[3]

To obtain the several single-line, single-layer problems from the general multilayer problem, three basic steps are involved.[3,4] First, vias have to be assigned to form connections with pins on the board. The connection patterns are then assigned to various layers. Finally all the connections are realized on a single-line, single-layer basis.

There are two main objectives in assigning vias to form connections with pins, namely: the minimization of vias columns and the minimization of vias usage. In fact:

a) Each addition of vias columns adds to the size of the board or alternatively if the size of the board is kept constant then the area for routing is decreased. It is also expensive to make extra vias columns than it is necessary.

b) Each via is consisted of a plated-through hole which is made at a different period using a different process from those of printed wires. This creates realiability problem at the contacts of a via, especially when under mechanical or thermal stress.[1] Less vias usage will enhance the realiability of the board.

In this paper, the vias assignment problem is discussed and some efficient algorithms for its solution are proposed. In Section II, a graph theoretical formulation of the problem is presented. The necessary and sufficient conditions for proper connections using k vias columns are given. Moreover, Section II discusses the complexities of the optimization problems involved in the vias assignment. The problems are shown to belong to a well-known class of hard computation problems, the class of NP-complete problems.[5,6] In Section III, an alternate sufficient condition is presented, the complexity even for this sufficient condition is still NP-complete but the alternate condition has the advantage to suggest a heuristic algorithm which is locally optimal. The computational complexity of the algorithm is then discussed.

In Section IV, further research directions are suggested and some concluding remarks are discussed.

## II. Formulation and Analysis

Given a board with fixed rectangular array of pins and vias, each pin and via can be identified by its locations, i.e., row and column. For example, a pin $p_i$ in row A and column 1 is indicated as $p_i = (A,1)$.

A signal net or simply a net is a set of pins $N_i = \{p_{u_1}, \ldots, p_{u_i}\}$ to be connected together by a continuous wire.

Figure 1a shows a board with one vias column and 4 nets defined over three pins columns.

Some partial connections can be made according to the assumptions mentioned in the Introduction without using any vias (e.g. pins of net 2 on row C and pins of net 1 on column 1). The set of pins of a net which can be connected without using vias are called generalized pins. In the degenerate case, a generalized pin is simply a singleton pin. Both net 3 and net 4 contain one generalized pin each, thus no via is needed for either net. We call such a net a trivial net. Nets 1 and 2 are of a different kind in that each net has at least two distinct generalized pins. In order to connect either net, some vias will have to be selected. It is in dealing with this kind of non-trivial nets that the vias selection process becomes important. Figure 1b shows a case where the vias on row C and row D are selected by net 1. Net 2 can not be connected because it requires a via on row C while there is none available. However, an alternative selection shown in Figure 1c reveals that both nets can be connected.

We begin our analysis of the vias selection problem for the non-trivial nets with a simple case with one vias column on the board.

In particular, we introduce a bipartite graph representation of the problem as follows:

Definition II.1:

Let $L = \{N_1,\ldots,N_p\}$ be the set of nets to be connected. Let $N_i^{(j_1,j_2,\ldots,j_p)}$ represent the set of pins on rows $j_1,j_2,\ldots,j_p$ associated with net $N_i$ forming one generalized pin. Let $v_i^k$ represent the via on row k belonging to vias column i. Let $G_1 = (S,T,E)$ be a bipartite graph[7] associated with L and one vias column as follows:

(i) Each S-node in $G_1$ repsents a generalized pin $N_i^{(j_1,\ldots,j_p)}$ associated with nontrivial net $N_i$.

(ii) Each T-node in $G_1$ represents a via $v_1^k$ on the vias column.

(iii) $e = (N_i^{(j_1,\ldots,j_p)}, v_1^k)$ is an edge in graph $G_1$ if there is a pin in $N_i^{(j_1,\ldots,j_p)}$ on the same row as the via $v_1^k$ (i.e. there is $j_m \in (j_1,\ldots,j_p)$ such that $j_m = k$).

Remark II.1:

Graph $G_1$ actually contains all possible connection patterns between generalized pins and the vias. To connect all nets is equivalent to finding a matching in $G_1$ such that all S-nodes in $G_1$ are covered. Figure 2a shows such a bipartite graph associated with the example in Figure 1a. Figures 2b, 2c show bipartite matchings representing the connection patterns of the nontrivial nets in Figures 1b, 1c respectively.

Theorem II.1:

Given a set of nets L, called the net list, and one vias column on the board, then the net list can be properly connected using one vias column if and only if in $G_1$ corresponding to L and one vias column, there is a matching $M_1$ which covers all S-nodes of $G_1$.

Proof:

It is direct to check that if all nets can be properly connected then a matching $M_1$ exists. On the other hand, if a matching $M_1$ exists, since the vias are all on the same column, connections among vias are direct and all nets can be connected. Q.E.D.

As a corollary of Theorem II.1, we can state a sufficient condition as follows.

Corollary II.1:

A net list L can be connected using k vias columns if two conditions are satisfied.

(i) $L = \bigcup_{i=1}^{k} L_i$ such that $L_i \cap L_j = \phi$ for all $i \neq j$.

(ii) For each $G_1^i$ corresponding to $L_i$ and one vias column, there exists a matching $M_1^i$ covering all S-nodes in $G_1^i$ for $i \in (1,\ldots,k)$.

Proof:

By Theorem II.1, if a matching $M_1^i$ exists in $G_1^i$, then the nets in $L_i$ can be connected using one vias column. Since this is true for all $i \in \{1,\ldots,k\}$, k vias columns are sufficient to connect all nets in L.

Q.E.D.

The crucial part of Corollary II.1 is to find a proper partition of net list L into k disjoint parts. Since the number of vias columns are to be minimized, k should be minimized. However, finding a partition which minimized k in Corollary II.1 is an NP complete problem. This problem is a hard computational problem in the sense that no present known algorithms with polynomial-time can be used to solve it.

Karp[8] has listed a few of the known NP complete problems.† In order to show that a problem A is NP complete, two criteria are to be satisfied:

(NP-1) A is NP.

(NP-2) Satisfiability is polynomial-time transformable to A.

The first (NP-1) criterion is generally easy to show. Either the Backtrack Criterion or the Existence Criterion in [8] suffices to check whether a problem is NP. In [8] Karp also indicated that it suffices to show that any NP complete problem is transformable to A instead of showing (NP-2). We shall follow the above discussion in the proof of NP completeness of a problem.

Theorem II.2:

Finding minimum k in Corollary II.1 is NP complete.

Proof:

We can show (NP-1) using the Backtrack Criterion. For each partition of k sets of nets we can verify whether each set of nets along with one vias column contains a matching in $O(p^{5/2})$ steps[9] where p is the number of generalized pins in the set of nets plus the vias

†In a fundamental paper[8], Karp showed that many important combinatorial problems, such as finding the chromatic number of graph, determining the solution of an integer programming problem, finding the maximum independent set in a graph, testing if a graph admits an Hamiltonian circuit, are "equivalent" from a computational complexity point of view.

Every solution method so far obtained for these problems is computationally explosive in the sense that its computational complexity (computation time and storage requirements) is bounded by an exponential in the size of the intput. The fundamental contribution due to Karp is that the existence of an algorithm for the solution of any of them with a complexity bounded by a polynomial in the size of the input would imply the existence of a polynomial algorithm for each of them. Thus there is a strong conjecture that none of these problems could be solved with an algorithm whose complexity is bounded by a polynomial in the size of the input. These "equivalent" problems are called NP-complete.

from one vias column. Thus our problem here is NP. It remains to show (NP-2) in order to prove that our problem is NP complete. We shall use the graph coloring problem[10] as the known NP complete problem and show that it is transformable to our problem.

Let G be an arbitrary undirected graph and $M_G$ the associated incidence matrix. We define a simple transformation as follows: For every node $n_i$ in G, there corresponds a net $N_i$ and for every edge $e_j$ in G, there corresponds a row $R_j$ on the board. A net $N_i$ has a generalized pin on row $R_j$ if node $n_i$ is incident with edge $e_j$ in G. Furthermore, we may increase the size of some of the generalized pins by adding pins to those generalized pins in an arbitrary manner.

We now claim that the chromatic number of graph G is k if and only if the corresponding set of nets can be connected according to Corollary II.1 using a minimum of k vias columns.

Suppose that the chromatic number of G is less than k, e.g. k-1. Then the nodes of G can be partitioned into k-1 disjoining parts such that none of the nodes in each part has any edge in common. By transformation introduced above, the corresponding nets of each partitioned group of nodes has the property that a matching satisfying Theorem II.1 exists between the generalized pins of the nets and one vias column. By Theorem II.1, this set of nets can be connected using only one vias column. Since there are k-1 such disjoint groups of nets, k-1 vias columns are sufficient to connect all the nets instead of k vias columns. Conversely, if the chromatic number of G is k, then a minimum of k vias columns is sufficient to connect all the nets associated with G. Otherwise, suppose k-1 vias columns are sufficient to connect all the nets, then the nets can be partitioned

into k-1 disjoint groups such that each group of nets can be connected using one vias column. In graph G, each set of nodes corresponding to each partitioned group of nets has the property that none of the nodes share an edge in common. Since there are k-1 such disjoint groups, the graph G can be k-1 colored. This is a contradiction. Thus we conclude that our problem is NP complete. Q.E.D.

Remark II.2:

In [8] Karp stated the list of NP complete problems as decision problems. It is worthwhile for us to state the problem of Theorem II.2 in a similar fashion:

Net List Partitioning.

Input: Net list L, integer k.

Property: There is a partition of nets in L into k or less disjoint parts such that the nets in each part can be connected using one vias column.

If we allow all possible connections as long as the unidirectional connection constraints are satisfied, a solution for the problem in Theorem II.2 may not give the minimum number of vias columns. We have excluded the possibility that a net may be connected using vias from more than one vias column. As a consequence, a solution for the problem in Theorem II.2 is only sufficient. Figure 3 shows an example where net 1 is connected using vias from both vias columns. In fast, if we insist on the formulation of Theorem II.2, we can readily check that minimum number of vias columns is 3 instead of 2 in Figure 3. Thus further considerations must be made when we allow selection of vias from more than on column in connecting one net.

To have a better appreciation of the problem at hand, let us consider a connection pattern associated with a net in Figure 4a. If all the vias are available for selection, then the choice of connection pattern for net 1 is not unique. The graph in Figure 4b contains all possible connections for net 1 in that any tree which contains all the pins of net 1 is a permissible connection pattern. If there are other nets present on the board, the selection of trees for all the nets is more complicated. Each tree must be disjoint from others and each tree must contain all the pins of the net it associates with. A procedure capable of finding trees for all the nets must have the ability of avoiding two nets competing for the same via.

We shall introduce the graph in Figure 4b as follows:

Definition II.2:

We let $\bar{G}_k$ be a forest bipartite graph associated with a set of nets and k vias columns as follows:

(i) Excluding trivial nets, for every generalized pin of a net, there is an S-node in $\bar{G}_k$, denoted by $N_i^{(j_1,\ldots,j_p)}$.

(ii) For every via in any of the k columns, there is a T-node in $\bar{G}_k$ denoted by $v_i^j$.

(iii) All the T-nodes in $\bar{G}_k$ associated with a vias column form a clique (i.e. $(v_i^q, v_i^r)$ is an edge in $\bar{G}_k$ for all $i = 1,2,\ldots,k$; $q,r = 1,\ldots,m$ where m is the number of rows on the board $q \neq r$). All the T-nodes in $\bar{G}_k$ associated with a vias row form a clique (i.e. $(v_i^s, v_j^s)$ is an edge in $\bar{G}_k$ for all $s = 1,\ldots,m$ and $i,j = 1,\ldots,k$ where $i \neq j$).

iv) $e = (N_i^{(j_1,\ldots,j_p)}, v_q^r)$ is an edge in $\bar{G}_k$ if the corresponding generalized pin has a pin on the same row as the via in question (i.e. there exists $j_t$ in $(j_1,\ldots,j_p)$ such that $j_t = r$).

Figure 4b is a graph $\bar{G}_2$ associated with one net and two vias columns. The graph $\bar{G}_k$ is important because it is both necessary and sufficient to set up conditions for a set of nets to be properly connected. We first state a lemma, then we shall give the necessary and sufficient condition under which nets can be properly connected.

Lemma II.1:

A net N can be connected using vias from k vias columns if and only if there is a tree $\mathcal{T}$ containing all S-nodes associated with net N in graph $\bar{G}_k$ representing the net N and k vias columns.

Proof:

Each edge in $\bar{G}_k$ represents a permissible connection on the board. Thus a tree $\mathcal{T}$ containing all the S-nodes associated with net N forms a proper connection pattern for net N. Conversely, a proper connection pattern for net N is simply a tree in $\bar{G}_k$. Q.E.D.

Theorem II.3:

A net list L can be connected using k vias columns if and only if there is a forest with $|L|$ disjoint trees in $\bar{G}_k$ such that each tree contains all S-nodes associated with exactly one net in L.

Proof:

If we have a forest as stated, then by Lemma II.1, each tree represents a proper connection pattern for one distinct net. Since there are $|L|$ disjoint trees in the forest, each one representing a net in L, thus all nets can be connected. Conversely, if all nets are

connected, it is easy to check that there is a forest as stated in $\bar{G}_k$.

Q.E.D.

Remark II.3:

Let us digress for a moment to Theorem II.2. In the context of graph $\bar{G}_k$ of Definition II.2, we can see that the connection patterns desired by the problem formulation in Theorem II.2 is a forest discussed in Theorem II.3 with a special property. Namely that none of the nets selected vias from more than one vias column to make connection. To see more clearly the relations between the forest of Theorem II.3 and the "special" forest implied in Theorem II.2 we shall introduce the following definition.

Definition II.3:

An overlap function OV associated with a forest F in $\bar{G}_k$ satisfying Theorem II.3, OV(F), is the sum of the number of vias that each tree in F is associated with, minus the number of S-nodes in the tree.

Remark II.4:

It is now easy to interpret Theorem II.2. The solution for the problem of Theorem II.2 is nothing but a forest satisfying Theorem II.3 such that the value of the overlap function, OV(F), is zero. In view of the discussion in Remark II.2, we shall restate Theorem II.2 as follows.

Simple Forest Problem.

Input:

$\bar{G}_k$ associated with net list L and k vias columns.

Property:

A forest F satisfying Theorem II.3 exists in $\bar{G}_k$ such that OV(F) = 0.

We have shown that the problem of Theorem II.2 is NP complete, we may expect that if we want to minimize the vias in the solution for the problem of Theorem II.3, the minimization problem will be NP complete. For the sake of clarity, we shall state this minimization problem as a decision problem.[8]

Complex Forest Problem.

Input:

$\bar{G}_k$ associated with net list L and k vias columns, integer p.

Property:

A forest F satisfying Theorem II.3 exists in $\bar{G}_k$ such that OV(F) $\leq$ p.

Theorem II.4:

The Complex Forest Problem in Remark II.4 is NP complete.

Proof:

The proof of this theorem is now simple. Since we have shown that the Simple Forest Problem in Remark II.4 is NP complete because it is just a restatement of Net List Partitioning Problem. The Simple Forest Problem is directly transformable to the Complex Forest Problem in that if we set integer p of the Complex Forest Problem equal to zero, then the two problems are identical.

Q.E.D.

The objectives in the vias assignment problem are minimization of vias columns and minimization of vias usage as stated in Section I. In light of Theorem II.4, we can easily guess that the problem is NP-complete. In fact, we can state the problem as a decision problem.

Minimum Forest Problem.

Input:

$\bar{G}_k$ associated with net list L and k vias columns, integer P.

Property:

A forest F satisfying Theorem II.3 exists in $\bar{G}_q$ such that $q \leq k$ and $OV(F) \leq p$.

We have investigated into the nature of connection patterns for a set of nets and explored the vias requirement condition within the framework of our problem formulation. We have shown that the problem we encountered is NP complete. It is not computationally feasible to solve the problem as it stands. Instead, some sufficient condition should be developed to offer a more computationally efficient approach to solve the vias assignment problem.

## III. Solutions: Preprocessing Criteria and Heuristics

Due to the complexity of the vias assignment problem, we have to be contented with a less ambitious approach to a solution of our problem. One intuitive idea is to construct or select a tree in graph $\bar{G}_k$ for one net at a time. Instead of anticipating future vias selections by other nets as we must in the global formulation, we may limit the scope to anticipate vias selection for the next net only. This strategy has the advantage that we can localize the selection process. Hence we can expect an algorithm with a more acceptable computation time. The price we have to pay is to require more vias columns on the board and use more vias for making connections.

The size of the graph $\bar{G}_k$ is extremely large especially when there is a large number of vias columns. There are some redundancies in the structure of $\bar{G}_k$ with respect to the cliques associated with each of the vias columns on the board. It is necessary to reduce the size of the graph $\bar{G}_k$ in our (local) problem consideration to have a more computationally economical algorithm for the tree selection.

We can make two observations. If there are some nets already connected, then all the edges and nodes associated with those connection patterns in $\bar{G}_k$ can no longer be a part for future tree construction. The second observation is that in selecting vias for the current net the S-nodes associated with other nets in $\bar{G}_k$ can never be a part of the tree being constructed. Thus none of the edges incident with those S-nodes can be a part of the tree either. However, the size of the remaining graph is still large due to the structure of T-nodes associated with the vias. We shall attempt to further reduce the size of $\bar{G}_k$.

There are two types of vias columns in the tree selection process in $\bar{G}_k$. A vias column in which some vias are already used is called an old vias column. A vias column in which none of the vias is yet chosen is called a new vias column. In selecting a tree for a net in $\bar{G}_k$, the choice of vias is not unique. Vias can be chosen from a new vias column, an old vias column or even from both types of columns. In order to consider all the possibilities, we have to look at a large portion of the graph $\bar{G}_k$. Large storage and computation will be required to do this. We shall now introduce two criteria to reduce the size of $\bar{G}_k$ we have to consider. Figure 5a shows a board with two nets, two old vias columns and one new vias column. If net 1 is

connected using vias from the new vias column, then net 2 can not be connected as shown in Figure 5a. Figure 5b shows a case where if net 1 is connected using vias from both the old and the new vias columns, net 2 can not be connected. On the other hand, Figure 5c shows a case where both nets can be connected if net 1 is connected using vias from old vias columns only. The crux of this example if that in selecting a tree, it is always better to select vias from the old vias columns so far as the next net is concerned. We shall state this idea as an observation.

Observation III.1:

If a net can be connected using available vias from old vias columns, then it is always better or as good to do so.

When it is determined that a net can not be connected using vias from the old vias columns, we will give the next selection criterion as an observation.

Observation III.2:

If a net can not be connected using vias from old vias columns alone, then it is always better or as good to use a new vias column for the connection.

Figure 6 shows that using a new vias column yields a better result in the sense that the next net has a better chance of getting connected.

Remark III.1:

The two seemingly simple observations provide some extremely effective guidelines in the local selection of a tree in $\bar{G}_k$. Instead of looking at graph $\bar{G}_k$ as a whole, Observation III.1 says we look at a small subgraph of $\bar{G}_k$ induced[7] by the S-nodes associated with the net

and the T-nodes associated with the unused vias from the old vias columns. If the tree selection is not unique, we may select a tree according to some optimization criteria. If it is determined that a tree does not exist in the induced graph discussed above, then Observation III.2 says that we can simply use a new vias column to connect the net. These two guidelines give two locally optimum criteria in the sense that the selection rules enhance the possibility of tree selection for the next net. The added advantage of these schemes is the reduction of the size of $\bar{G}_k$ to a small induced graph.

For clarity purpose, we shall denote by $G_{\mathcal{L} \cup S_i}$ as the induced subgraph of $\bar{G}_k$ associated with net $N_i$. $S_i$ denotes the set of S-nodes associated with the generalized pins of net $N_i$ and $\mathcal{L}$ denotes the T nodes associated with the set of unused vias from the old vias columns. Thus $G_{\mathcal{L} \cup S_i}$ is the induced subgraph of $\bar{G}_k$ by nodes in $\mathcal{L}$ and $S_i$. The problem then is to find a tree $\mathcal{T}_i$ in $G_{\mathcal{L} \cup S_i}$ such that all S-nodes are covered. If no such tree exists, then we use a new vias column to connect the net.

Suppose that a tree $\mathcal{T}_i$ exists in $G_{\mathcal{L} \cup S_i}$, we still have to decide the many choices we have facing us. Let us recall for a moment the formulation in Remark II.4 for the Complex Forest Problem. Apparently the function value OV(F) is to be minimized with respect to the forest F. In the context of present discussion, we are only dealing with the selection of a single tree instead of a forest. However, the function OV is still well defined with respect to a tree. Minimizing the value of function OV means to minimize the number of vias selected in a tree and is equivalent to minimize the number of vias columns associated with the tree. However, this problem will be shown

to be NP complete. We shall state the problem as a decision problem as in [8], then we shall show that the problem is NP complete.

Minimum Tree Problem.

Input:

$G_{\mathcal{L} \cup S_i}$ associated with $S_i$ and a set of vias columns $\mathcal{L}$, integer p.

Property:

There is a tree $\mathcal{T}_i$ containing $S_i$ in $G_{\mathcal{L} \cup S_i}$ such that $OV(\mathcal{T}_i) \leq p$.

Theorem III.1:

The Minimum Tree Problem is NP complete.

Proof:

Checking whether a tree $\mathcal{T}_i$ covers $S_i$ and calculating $OV(\mathcal{T}_i)$ can be done in O(p) steps where p is the size of the graph $G_{\mathcal{L} \cup S_i}$. Thus we can easily verify that the problem is NP.

We shall use the set covering problem [10] to show that our problem is NP complete. We state the set covering problem as a decision problem:

Set Covering Problem.

Input:

S is a set, $F = \{S_1, S_2, \ldots, S_q\}$ is a family of sets; integer k.

Property:

There is a subset $F' \subset F$ such that $S \subset \bigcup_{T \in F'} T$ and cardinality of $F' \leq k$.

In order to show that our problem is NP complete, we need to show that the set covering problem is transformable to our problem.

Let $S_i^*$ denote a set to be covered by a family of sets $\mathcal{L}^*$ in a set covering problem. Hence we are looking for a subset $\mathcal{L}^{*'} \subset \mathcal{L}^*$ such that $S_i^* \subset \bigcup_{c_j^* \in \mathcal{L}^{*'}} c_j^*$ and cardinality of $\mathcal{L}^{*'} \leq k$. If we insert element {a} to every $c_j^* \in \mathcal{L}^*$ where $\{a\} \notin S_i^*$ and we then add {a} to set $S_i^*$, the solution of the set covering problem will remain unchanged. We then identify $S_i^*$ with $S_i$, $\mathcal{L}^*$ with $\mathcal{L}$, and each $c_j^*$ with $c_j$ as an old vias column, graph $G_{\mathcal{L} \cup S_i}$ can be constructed. If we identify k as equal to p + 1 then we see that an input of the set covering problem can be transformed into an input of our problem. It remains to show that with this transformation, the properties in both problems are equivalent. We shall prove it by contradiction.

Suppose that a solution exists for the set covering problem, but in the corresponding minimum tree problem we have $OV(\mathcal{T}_i) > p = k-1$. This is impossible because if the cardinality of the cover for set $S_i^*$ is $\leq k$, at most $|S_i| + k-1$ vias are sufficient to connect the net represented by $S_i$. Hence the value $OV(\mathcal{T}_i) \leq k-1 = p$ which is a contradiction. Conversely, if $OV(\mathcal{T}_i) \leq p$ is satisfied in our problem, then the cardinality of the cover is at most p+1 = k. Thus we conclude that our problem is NP complete.

Q.E.D.

We have seen that even a local problem dealing with $G_{\mathcal{L} \cup S_i}$, a small induced subgraph of $\bar{G}_k$, is an NP complete problem. A sensible approach at this point is to look for a simple heuristic which can simulate the objective of Minimum Tree Problem.

As a first step, we may try to connect the most number of S-nodes in $S_i$ by selecting one old vias column (the T-nodes associated with the old vias column) in $G_{\mathcal{L} \cup S_i}$. We may repeat this selection procedure

for the remaining S-nodes in $S_i$ until a tree $\mathcal{T}_i$ covering $S_i$ is constructed. This procedure will locally minimize the function OV because at each selection step, vias are chosen from one column with the maximum connection to the remaining S-nodes in $S_i$. We shall use an example in Figure 7 to illustrate the selection process.

Figure 7a shows a net with 6 generalized pins and 5 old vias columns on a board. Vias column $C_3$ is the best initial choice because the net has 4 generalized pins in common rows with vias column $C_3$. The others have at most 3 such nonempty intersections. With $C_3$ as the initial selection, the pins of net 1 on rows $R_1, R_2, R_3, R_4$ and $R_{10}$ can be connected using 4 vias from $C_3$. The connection is shown in Figure 7b. The next step is to select another vias column which can connect the most number of remaining generalized pins of the net. Considering the partially connected pattern in Figure 7b, the most economical way of reaching a new pin is through a shortest path from the set of already connected points (pins and vias) to one of the reamining pins. Dijkstra's breadth first expansion method [11] examplifies such a scheme. There are two possible shortest paths from the set of connected points to a net pin:

(i) via($R_1$,$C_3$)-via($R_1$,$C_5$)-via($R_5$,$C_5$)-new pin on row $R_5$,

and

(ii) via($R_3$,$C_3$)-via($R_3$,$C_1$)-via($R_5$,$C_1$)-new pin on row $R_5$.

As a consequence, two vias columns are available for selection, namely $C_1$ and $C_5$. $C_1$ can connect only one generalized pin (pins on $R_5$, $R_6$, and $R_7$) as compared to $C_5$ connecting two generalized pins (pins on rows $R_5, R_6, R_7$ and pins on row $R_8$). Local greedy selection rule prefers the selection of $C_5$ in this case. Figure 7c shows a tree thus constructed. Essentially the strategies can be summarized into three parts. First

an initial selection of old vias column is performed such that the most number of generalized pins can be connected using just one vias column. The second and the third parts form an iterative process. A breadth first search is performed to find a shortest path from the set of already connected points to a new pin. A backtrack procedure is then performed to select a vias column which can connect most number of remaining generalized pins. The iterative process continues till all pins of the net are connected. In case a tree $\mathcal{T}_i$ can not be constructed using the old vias columns alone, a new vias column is used to construct the tree.

Remark III.2:

In spite of the simplicity of the procedure described above, both Observation III.1 and Observation III.2 are adopted. Moreover, the minimization of OV function is done implicitly on a local scale.

We shall first introduce some notations then we shall formalize the tree construction algorithm.

Definition III.1:

We denote by $\Lambda(c_j,S)$ an operation between the set of generalized pins in S, a subset of $S_i$, and an old vias column $c_j$ in $\mathcal{L}$. The value of $\Lambda(c_j,S)$ is the number of generalized pins S has in common rows with the available vias in $c_j$. We denote by $\Gamma(c_j,S)$ the set of generalized pins associated with the operation $\Lambda(c_j,S)$ and $\theta(c_j,S)$ the set of vias associated with the operation $\Lambda(c_j,S)$.

The cardinalities of $\Gamma(c_j,S)$ and $\theta(c_j,S)$ are identical and are equal to $\Lambda(c_j,S)$. Referring to Figure 7a, if we let $S = S_1$ and $c_j = C_5$, then $\Gamma(C_5,S_i)$ = [(pins on $R_1,R_2$), (pins on $R_5,R_6,R_7$), (pin on $R_8$)]

and $\theta(C_5,S_i)$ = [(vias on $R_1,R_2$), (vias on $R_5,R_7$), (via on $R_8$)] where $\Lambda(c_5,S_i)$ is simply equal to 3. We shall now present the algorithm.

Algorithm Tree.

Initial Selection:

Step 1: Select an old vias column $c \in \mathcal{L}$ such that $\Lambda(c,S_i) = \max_{c_j \in \mathcal{L}} \Lambda(c_j,S_i)$.

Step 2: $S \leftarrow S_i - \Gamma(c,S_i)$, $V(\mathcal{T}_i) \leftarrow \Gamma(c,S_i) \cup \theta(c,S_i)$,

Mark(x) = 0 for all x in $V(\mathcal{T}_i)$,

$E(\mathcal{T}_i) \leftarrow \{(u,v)$: u in $\Gamma(c,S_i)$, v in $\theta(c,S_i)$ such that u,v share a row on the board$\} \cup \{(w_p,w_{p+1})$: $\theta(c,S_i) = (w_1,\ldots,w_k)$, $p = 1,\ldots,k-1\}$,

$T \leftarrow V(\mathcal{T}_i)$, Lab(x) for all x in $V(\mathcal{T}_i)$.

Phase 1:

Step 1: $j \leftarrow 0$.

Step 2: If $V(\mathcal{T}_i) \supset S_i$, go to Step 7.

Else,

Step 3: $j \leftarrow j+1$, If Adj(T) = $\phi$, go to Step 8.

Else,

*Adj(T) is the set of nodes adjacent to T in $G_{\mathcal{L} \cup S_i}$.

Step 4: Mark(x) = j for all x in Adj(T).

Step 5: If Adj(T) $\cap$ S $\neq \phi$, then pick s in Adj(T) $\cap$ S, go to Phase 2.

Else,

Step 6: $T \leftarrow T \cup$ Adj(T), go to Step 2.

Step 7: End of Algorithm.

Step 8: Tree $\mathcal{T}_i$ can not be constructed using old vias columns. Connect the net using a new vias column, end of algorithm

Phase 2:

Step 1: $T^* \leftarrow \{s\}$, $k \leftarrow j$, Mark(s) = 0,

$T \leftarrow T^*$, Lab(s), $y \leftarrow s$.

Step 2: If $k = 0$, go to Step 6.

Else,

Step 3: $k \leftarrow k - 1$.

Step 4: Pick n in $Adj(T^*)$ such that Mark(n) = k and n in vias column c such that $\Lambda(c,S) = \max_{c_p \in \mathcal{L}} \Lambda(c_p,S)$.

$Mark(n) \leftarrow 0$, $T \leftarrow T \cup \{n\}$, Lab(n),

$E(\mathcal{T}_i) \leftarrow E(\mathcal{T}_i) \cup \{(y,n)\}$, $y \leftarrow n$.

Step 5: $T^* \leftarrow \{n\}$, go to Step 2.

Step 6: Go to Phase 1.

Note:

There are several notations which need to be explained in the algorithm above. $V(\mathcal{T}_i)$ is the vertex set of the tree $\mathcal{T}_i$ under construction, it contains both pins and vias. $E(\mathcal{T}_i)$ is the edge set for the tree. Mark(.) is used to denote distance of a node from the set of already connected points. Lab(.) indicates the node is in the tree.

The complexity of the algorithm can be analyzed as follows. In the Initial Selection, operation $\Lambda(\ ,\ )$ is simply a logical "and" operation. It is linear with respect to the number of nodes in $G_{\mathcal{L} \cup S_i}$. In Phase 1, it takes at most m steps to reach a new target where m is the number of edges in $G_{\mathcal{L} \cup S_i}$. In Phase 2, $q|S|$ comparisons in Step 4 are required to select a vias column where q is the number of vias columns having vias adjacent to the target pin determined in Phase 1 and S is the remaining generalized pins in $S_i$. It then takes at

most n-1 steps to backtrack where n is bounded by the number of nodes in $G_{\mathcal{L} \cup S_i}$. At most $|S_i| - 1$ iterations are needed for Phase 1 and Phase 2. Thus the complexity is in the order of $O[n + (|S_i| - 1)(m + (n-1) + q|S_i|)]$. In general q is bounded by $|\mathcal{L}|$ and is small; both $|S_i|$ and n are small compared to m. Thus we may say roughly that the complexity of the algorithm is linear with respect to the size of the graph $G_{\mathcal{L} \cup S_i}$. We shall now analyze how the algorithm performs with respect to the function OV.

Definition III.2:

Let c* be the number of old vias columns in $G_{\mathcal{L} \cup S_i}$ in tree $\mathcal{T}_i^*$ according to Theorem III.1. Let the total number of vias $\mathcal{T}_i^*$ be $V^*$ and let k* be the number of vias rows associated with $\mathcal{T}_i^*$ such that net $N_i$ has no pins on those rows. We denote similarly, c,V,k with respect to tree $\mathcal{T}_i$ constructed by the tree algorithm.

Remark III.3:

The example in Figure 8 shows a case where $|S_i| = 4$, $c = 3$ and $k = 2$. There are a total of 8 vias in the tree thus $V = 8$. In the next theorem we shall give the relations among c,k,$S_i$ and V.

Theorem III.2:

$$V^* = c^* + k^* + |S_i| - 1 \qquad \text{(III.1)}$$

$$V = c + k + |S_i| - 1 \qquad \text{(III.2)}$$

Proof: It suffices to show just one case, the other one is similar. We shall show Eq. (III.2).

In tree $\mathcal{T}_i$, all the generalized pins are represented by $S_i$. In order to connect all the nodes in $S_i$ at least $|S_i|$ vias, one for each generalized pin are necessary. Moreover, there are $OV(\mathcal{T}_i)$ additional vias. The value $OV(\mathcal{T}_i)$ can be accounted as follows: There are k rows on the board where no pin of the net is present and vias are chosen into the tree from those rows. Thus an additional k vias are in tree $\mathcal{T}_i$. Furthermore, c vias columns are involved in $\mathcal{T}_i$. In order to connect the vias from c columns, an additional c-1 vias are needed. Thus we have $OV(\mathcal{T}_i) = c + k - 1$ and $V = c + k + |S_i| - 1$.

Q.E.D.

Johnson[10] showed that the set covering problem is NP complete, moreover, there is no heuristic that can give an approximated solution within a constant bound. Therefore, we can expect that our tree algorithm behaves similarly with respect to Minimum Tree Problem. We shall briefly sketch this aspect as an observation.

Observation III.3:

Use the notation of Definition II.5, we can easily check that

$$|S_i| + k > c \tag{III.3}$$

Since the total number of rows tree $\mathcal{T}_i$ containing vias is $|S_i| + k$ and by algorithm tree, c is at most $|S_i| + k - 1$. Hence we have $|S_i| + k-1 \geq c$ which is equivalent to Eq. (III.3). Now if we consider the ratio $r = V/V^*$, then by Eqs. (III.1), (III.2) and (III.3) we have

$$r = \frac{V}{V^*} = \frac{|S_i| + c - 1 + k}{|S_i| + c^* - 1 + k^*} < \frac{2|S_i| + 2k - 1}{|S_i| + k^* + c^* - 1} \tag{III.4}$$

We see that the ratio of Eq. (III.4) is not bounded by any constant because there is no relation between k and k*. Instead, it is problem structure dependent. However, there is a special case when k = 0. In this case, Eq. (III.4) is bounded by 2 because $c < |S_i|$.

## IV. Concluding Remarks

In this paper, we have discussed the formulation and complexity of the vias assignment problem in the routing of multilayer printed circuit boards. The formulation of the problem has been given using graph theory. The complexity of the problem has been derived using the NP complete problem theory. The discussion of the complexity of of the vias assignment procedure has led naturally to the introduction of a heuristic algorithm.

The performances of this algorithm have been evaluated. The obtained results can be incorporated in a general procedure for the routing process of a multilayer printed circuit board. In fact at the end of this assignment stage, all the connections are defined unidirectionally in the sense that no direct connection exists for two points which are neither on the same row nor on the same column. This effectively reduces the routing problem into several single line routing problems. Hence we can realize each single line connections with the algorithms presented in [3]. Much work remains to be done in the vias assignment problem.

In particular, the complexity of the optimization problem involving the minimization of the via columns used for the connections of a net list L, regardless of the number of vias used is not known, even though it is strongly conjectured that it belongs to the NP-complete problem class.

Another open problem is the formulation and analysis of the following extension of the vias assignment problem. Suppose we are given a board with 17 equal sized chips and some input-output (I/O) terminals on the edge of the board (Figure 9a). If the vias appear only columnwise as shown in Figure 9b, then a very low routing space utilization will result in area A and in the shaded areas because very few pins are present on the rows in those areas. Vias distribution is unbalanced and connections will be congested in the unshaded areas besides area A. To have a more balanced vias distribution, vias must appear in both area A and shaded areas of Figure 9b. Figure 9c shows a scheme where vias appear columnwise as well as row wise. Since the procedure we developed can not be applied to this case, the problem is: How to assign vias when they appear rowwise as well as columnwise.

A possible approach is to divide the board into several sections as in Figure 9c. Then the vias assignment procedure presented in this paper can be applied to each section independently. In fact within each section the only change required is the orientation of the vias and of the pins (for example from vertical to horizontal in Section I).

The assigned vias are then treated as pins, and the iteration loop can be applied again to the whole board where this time the vias will appear columnwise. However, the sectioning of the boards requires human judgement at the present time and it is not clear whether this approach is effective in dealing with very large sized problems.

References

[1] Printed circuit and multilayer board design and production, Milton S. Kiver Publication Inc., Chicago (1972).

[2] H. So: Some theoretical results on the routing of multilayer, printed-wiring boards; Proc. 1974 IEEE Int. Symp. on Circuits and Systems, pp. 296-303.

[3] B. S. Ting, E. S. Kuh and I. Shirakawa: The multilayer routing problem: algorithms and necessary and sufficient conditions for the single-row, single-layer case; IEEE Trans. on Circuits and Systems, Vol. CAS-23, No. 12, pp. 768-778, Dec. 1976.

[4] B. S. Ting: A new approach to the interconnection problem of multilayer printed circuit board, Ph.D. Thesis, EECS, University of California, Oct. 1976.

[5] S. A. Cook: The complexity of theorem proving procedures; Proc. 3rd Annual ACM Symp. on Theory of Computing, pp. 151-158 (1971).

[6] R. M. Karp: Reducibility among combinatorial problems; in Complexity of Computer Computations. (R. E. Miller and J. W. Thatcher, Eds.) Plenum Press, New York, (1972).

[7] F. Harary: Graph Theory; Addison-Wesley Publishing Company (1972).

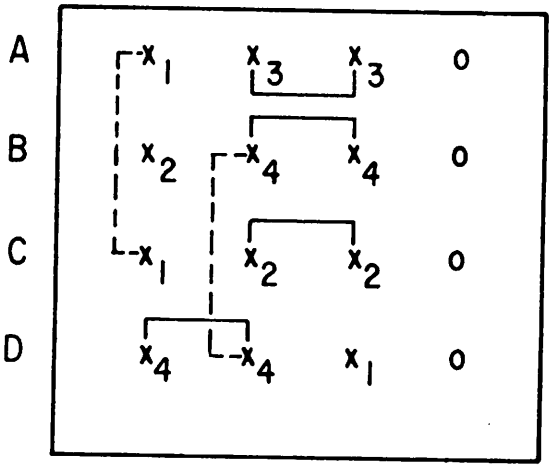
[8] R. M. Karp: Complexity of combinatorial problems; Networks, Vol. 5, No. 1, pp. 45-68 (1975).

[9] J. E. Hopcroft and R. M. Karp: An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs; Siam J. Comp., Vol. 2, No. 4, pp. 225-231 (1973-12).

[10] D. S. Johnson: Approximation algorithms for combinatorial problems; Proc. of 5th Annual ACM Symp. on Theory of Computing, pp. 38-49 (1973).

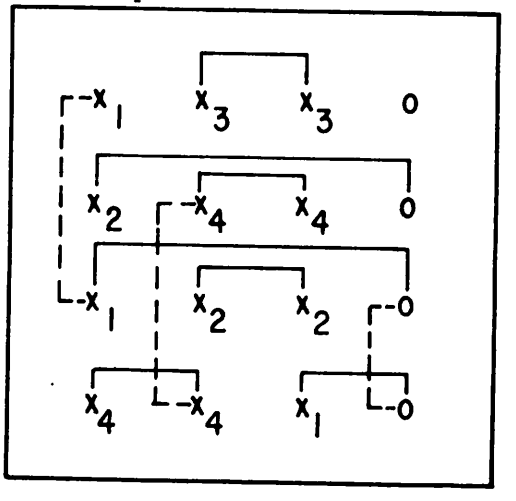
[11] E. W. Dijkstra: A note on two problems on connection with graphs; Numerishe Mathematik, 1, p. 269 (1959).

[12] S. Goto, I. Cederbaum and B. S. Ting: Sub-optimum solution of the back-board ordering with channel capacity constraint: Memo. ERL-M597, Electronics Research Laboratory, University of California, Berkeley, Cal. 94720 (1976).



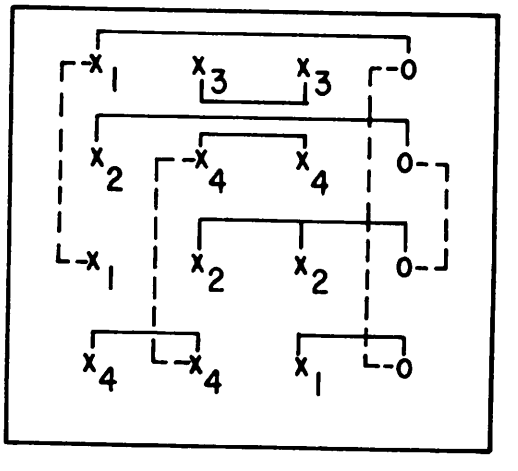

(a)

Figure 1a. 4 nets defined on a board, each generalized pin is a connected part.



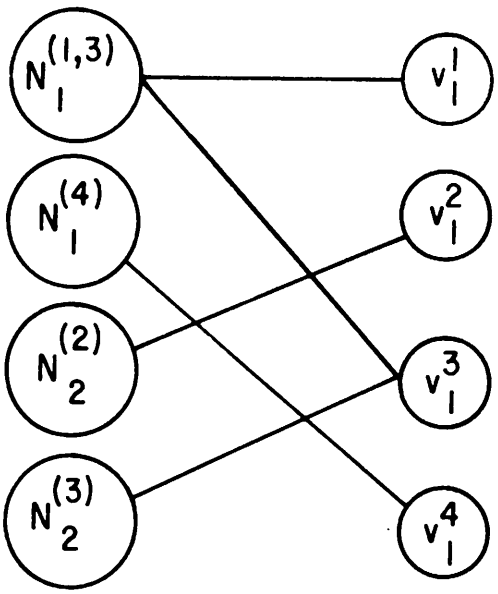

(b)

Figure 1b. Net 2 left unconnected.



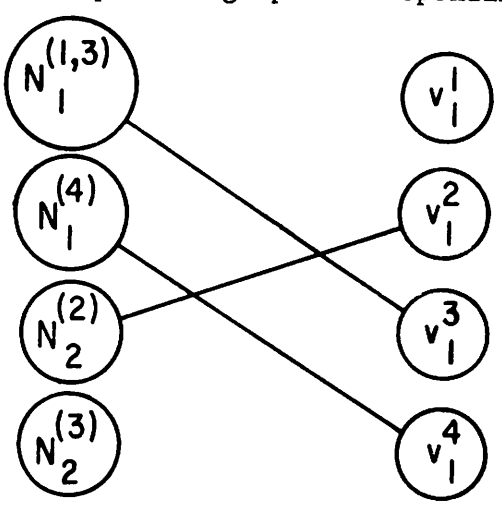

(c)

Figure 1c. All nets are properly connected.



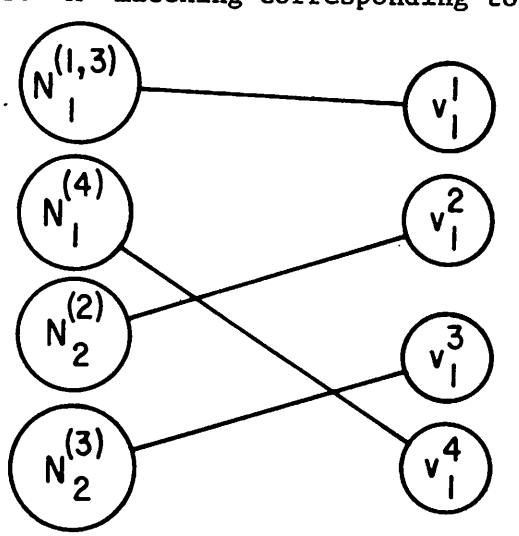

(a)

Figure 2a. Bipartite graph corresponding to Figure II.1a.




(b)

Figure 2b. A matching corresponding to Figure II.1b.




(c)

Figure 2c. A matching corresponding to Figure II.1c.

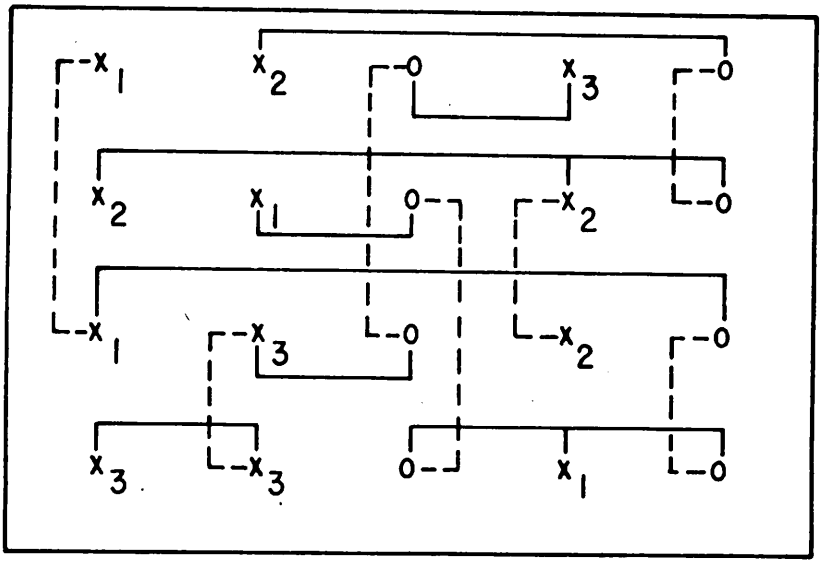



Figure 3. Net 1 using vias from both vias columns.

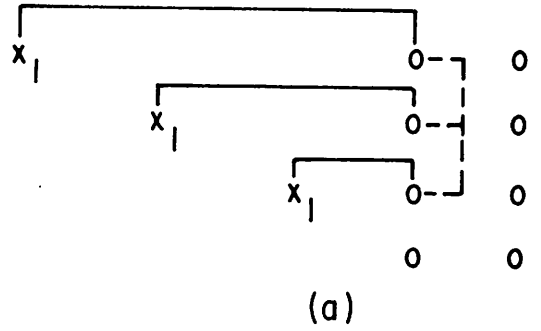



Figure 4a. A tree associated with net 1.

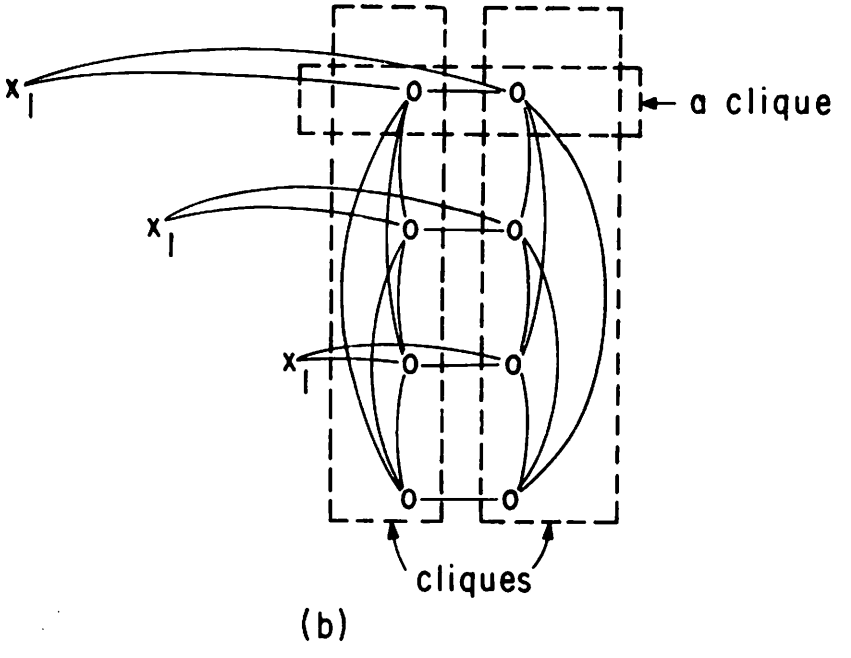



Figure 4b. A graph emcompasses all possible trees associated with net 1.

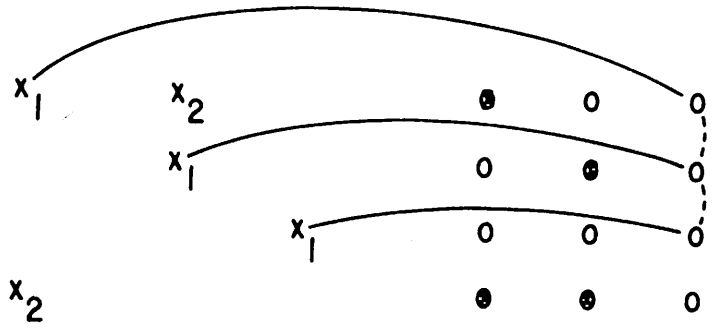



Figure 5a. Net 2 can not be connected.

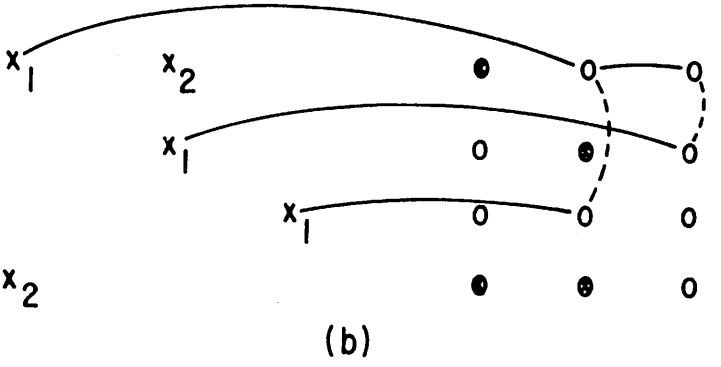



Figure 5b. Net 2 can not be connected.

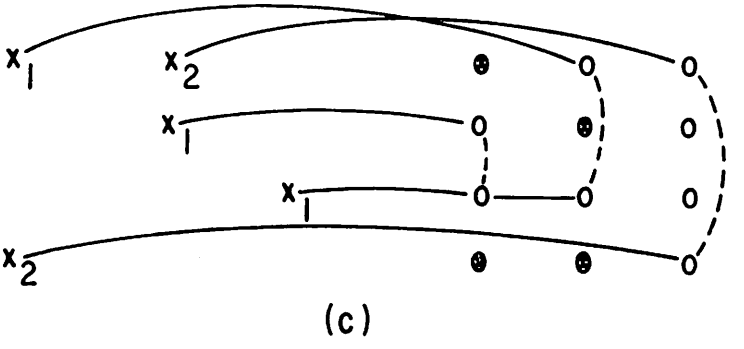



Figure 5c. Both nets are connected.

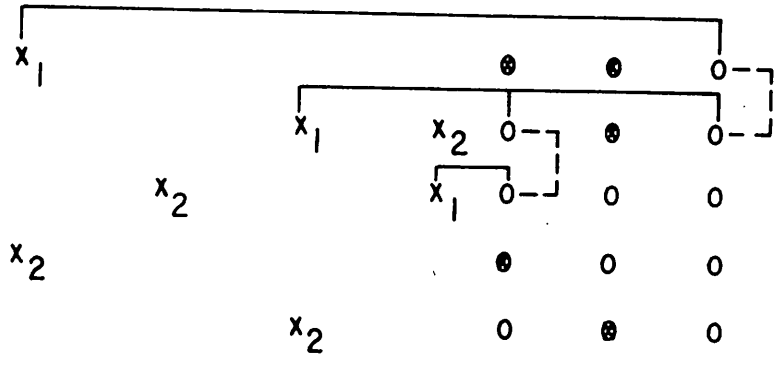



Figure 6a. Net 2 can not be connected.

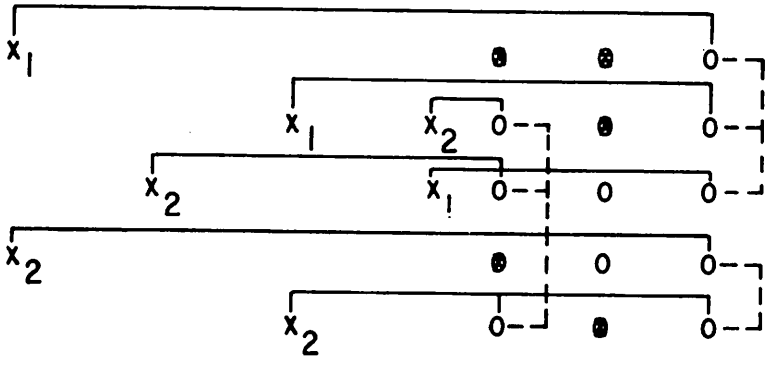
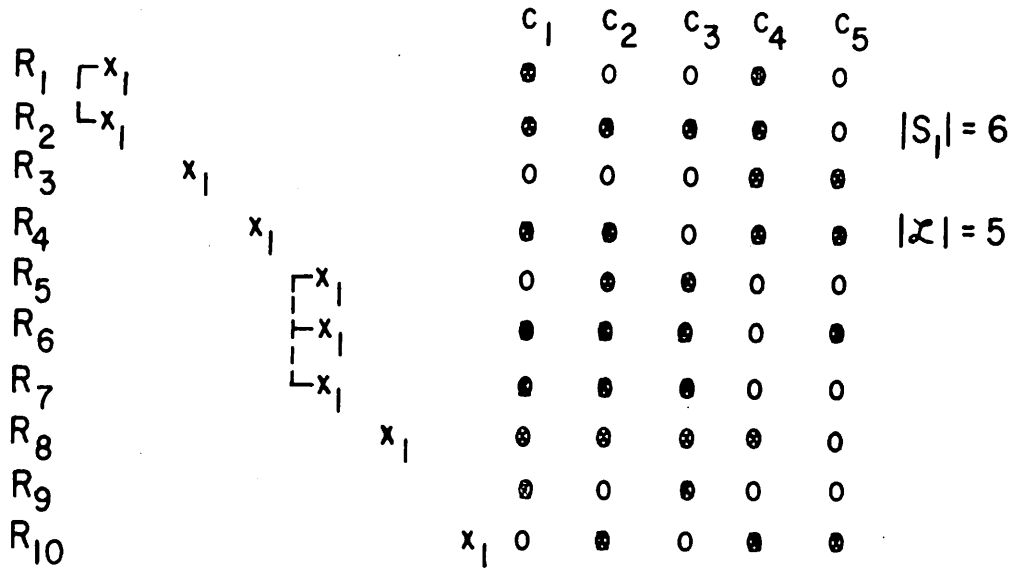
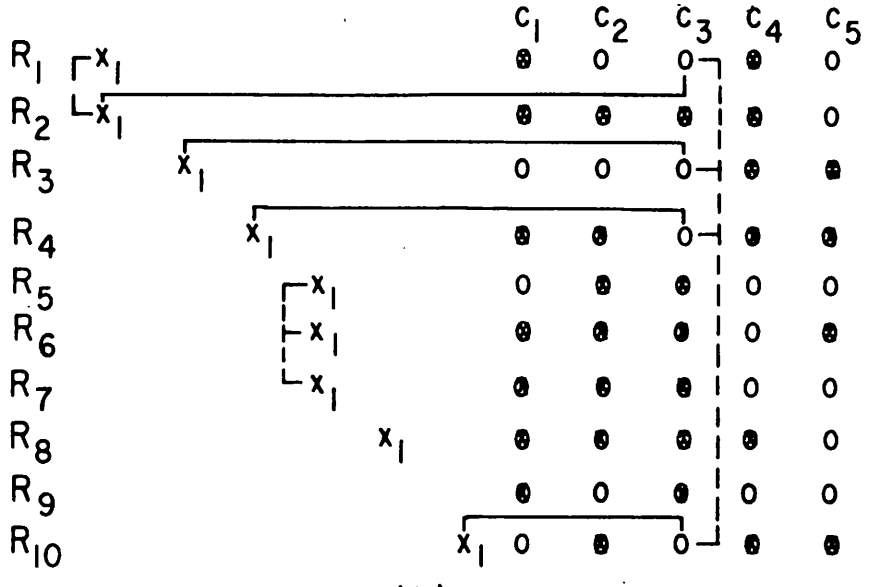



Figure 6b. Both nets can be connected.



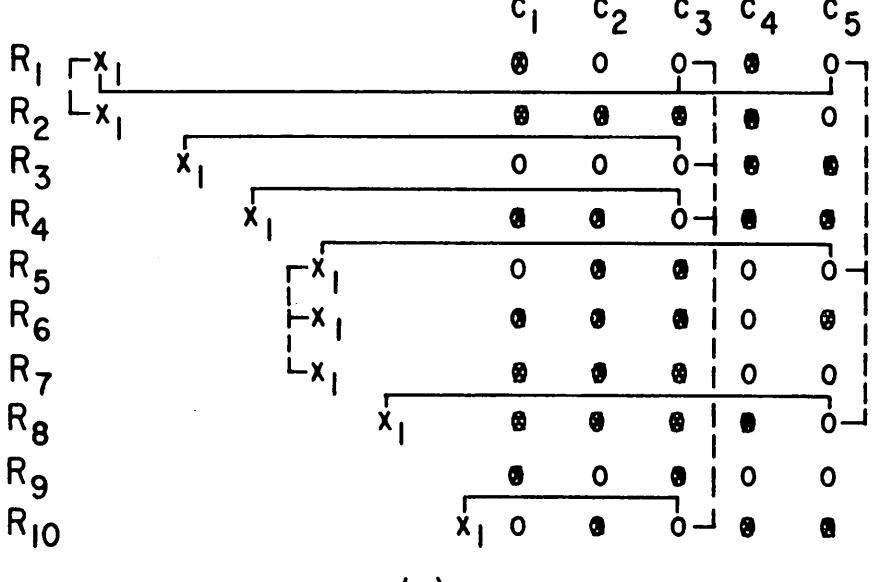

(a)

Figure 7a. 6 generalized pins and 5 old vias columns are present.




(b)

Figure 7b. A partial connection of net 1 using vias column $C_3$.




(c)

Figure 7c. All pins are connected using 2 old vias columns.

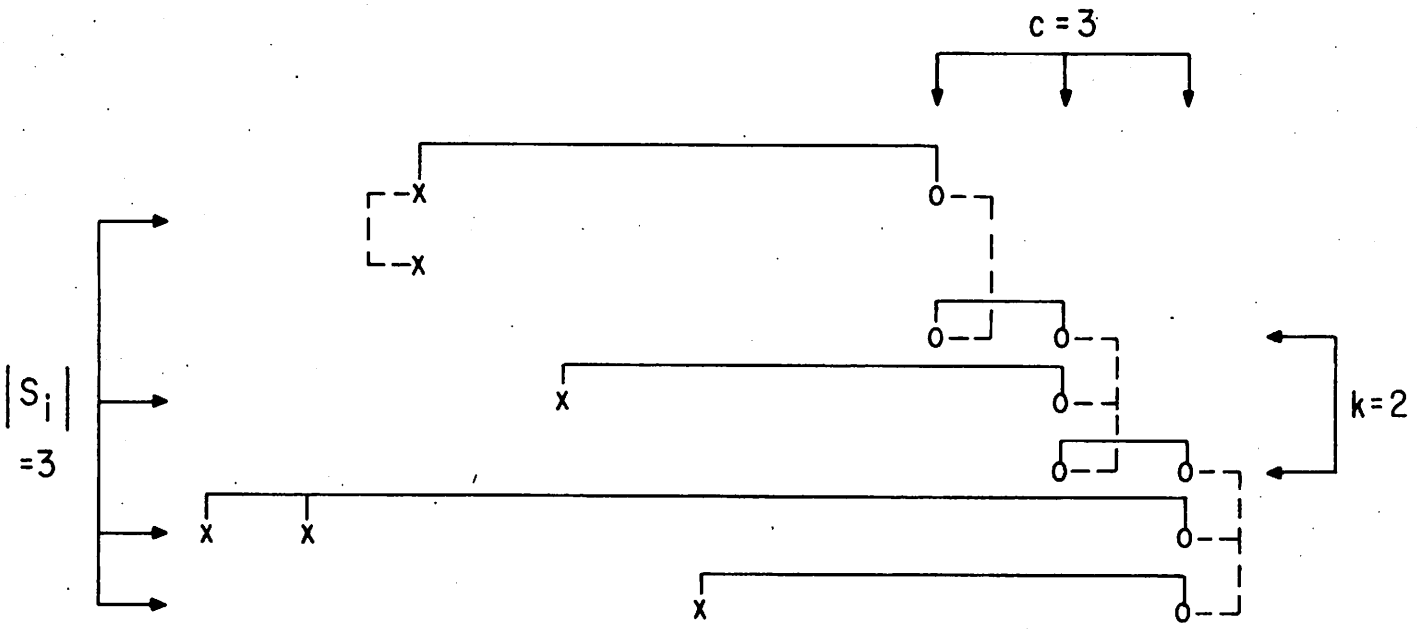
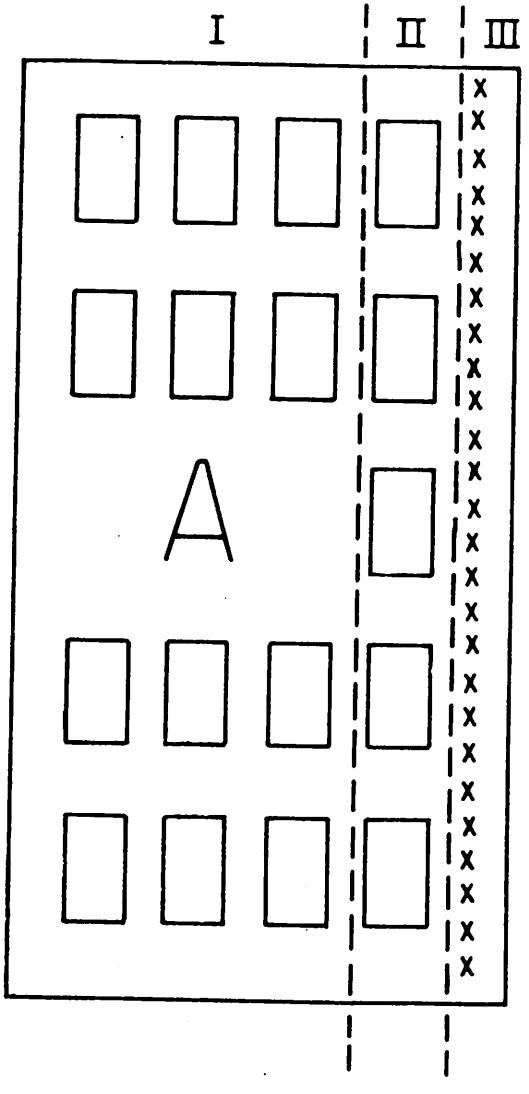
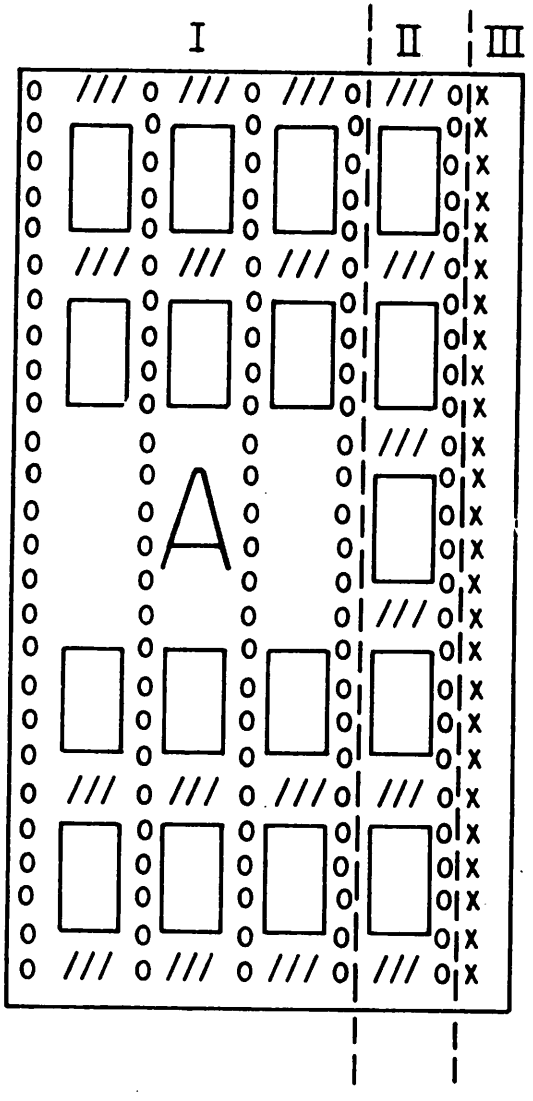



Figure 8. A net connected using 8 vias.



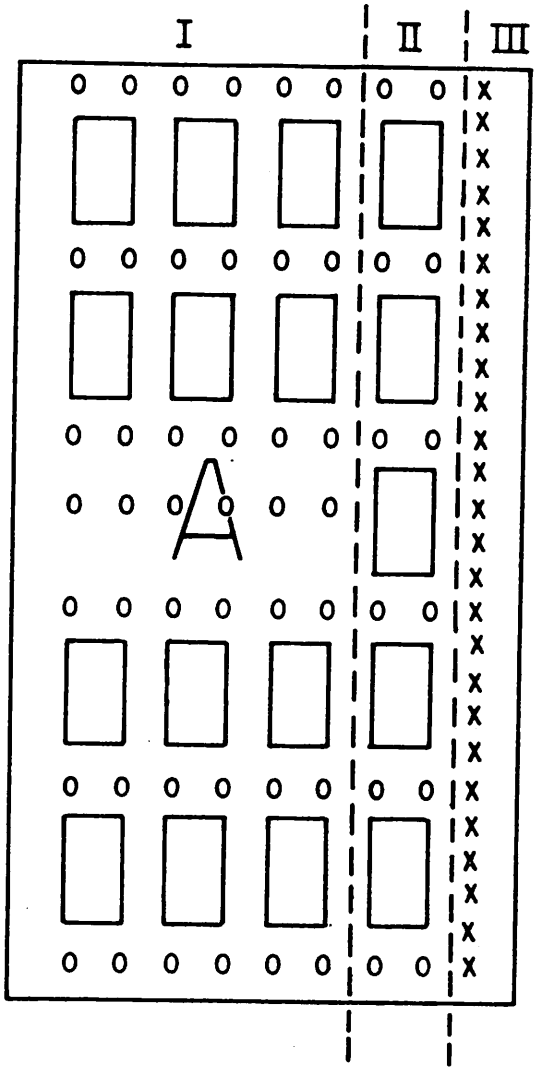

(a)

Figure 9a. A board with 17 chips.




(b)

Figure 9b. All vias appear columnwise.




(c)

Figure 9c. Board divided into 3 sections, vias can appear rowwise.