The Complexity of Single Row Routing

RAGHUNATH RAGHAVAN AND SARTAJ K. SAHNI, MEMBER, IEEE

Abstract—This paper investigates a systematic suboptimal approach to multilayer rectilinear wire routing: the single row approach. The method involves decomposing the general rectilinear wiring problem into a number of conceptually simpler single row problems. The complexity of the decomposition process is first considered. Under interesting optimization criteria, each step in the decomposition process is shown to be NP-hard. Then the single row wiring problem itself is considered under a variety of optimization criteria. In some cases, either efficient or "usually good" algorithms are possible. In other cases, the problem turns out to be NP-hard.

I. Introduction

The fundamental *wire routing problem* occurs at all levels of the physical packaging hierarchy (e.g., within the IC chip, in the multichip package, on the PC card, in the backpanel and sidepanel, etc.). In this basic form, the problem may be stated as follows. We are given a set $W=\{(u_i, v_i)(x_i, y_i)|1 \leqslant i \leqslant n\}$ of n wires. (u_i, v_i) and (x_i, y_i) are the respective coordinates of the end points of wire i, $1 \leqslant i \leqslant n$. When more than one layer is available, the wire end-points are assumed to exist on all the layers.[1] The problem is to define the precise conductor paths for each of these wires, using as few layers as possible and subject to the following constraints:

(1) Each conductor path must be made up of horizontal and vertical segments only (the rectilinear constraint).

(2) The various conductor paths are not allowed to cross on any given layer.

This problem is NP-hard (for a proof, see [11]). So, it is not surprising that most approaches to the problem have been heuristic in nature (see, e.g., [5], [9], [6], [4] etc.) These approaches have the common characteristic that there is no guarantee of their finding optimal solutions efficiently. Furthermore, being very ad hoc in thier various approaches, they give no feel for the inherent difficulty (i.e., the routability") of the given instance. This makes it very hard to evaluate the suitability of any such heuristic approach for a given mix of routing problems.

This paper presents the results of a thorough investigation of a promising approach to the general wiring problem. This approach is called the *single row approach* [14]. It is a very systematic (but suboptimal) approach to wiring; it even allows a-priori estimation of the fraction of wires inherently routable. It works best for very large instances of the wiring problem (i.e., containing more than 1000 wires), which usually have reasonably "regular" geometries; i.e., there is a reasonable nontrivial rectangular grid, such that each wire end-point coincides with some grid point. (Of course, it is not necessary that every grid point be so occupied.) We will refer to this imaginary grid as the *point grid*.

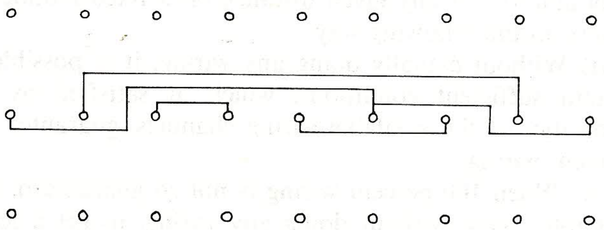

Fig. 1.

Often, there are restrictions as to where vias may exist. In that case, the point grid must be refined so that each legal via location also coincides with some grid point. In any case, the extent of the point grid is limited by the (fixed) area of the wiring surface.

The essence of the single row approach lies in three *strategic constraints* introduced by it:

A) Wires can only connect points (either pins or vias) that are either in the same row or in the same column. This amounts to decomposing the general wiring problem into a number of "single row" wiring problems, one for every row and every column of the point grid.

B) The layout of a wire connecting points in the same row (column) must be confined to the two wiring channels on either side of that row (column). Fig. 1 shows such a layout. So, if the wiring channels have sufficient capacities, the layout of each row (column) is independent of the layout of any other row (column).

C) On any given layer, only rows (columns) can be laid out. So, the layout of any row is independent of the layout of any column, since they will be on different layers. In conjunction with constraint B), this makes each "single row" wiring problem truly independent of any other. Another implication of this constraint is that at least two layers must be available for wiring. (Even when one-layer realizations are possible, the single row approach will use two or

[1] This assumption reflects certain limitations in early package fabrication technologies. It is a little simplistic in the modern context. However, we shall retain this assumption because: (i) it really impacts only the pre-routing phase of the single row approach; (ii) it represents a simplifying assumption; and (iii) we wish to add to a pool of complexity results that have been obtained using this assumption.

Manuscript received November 15, 1981; revised February 3, 1983. This work was supported in part by the National Science Foundation under Grant MCS 80-005856.

R. Raghavan is with Sperry-Univac, Roseville, MN 551153.
S. Sahni is with the Department of Computer Science, University of Minnesota, Minneapolis, MN 55455.

0098-4094/84/0500-0462$01.00 ©1984 IEEE

more layers, except in a few rare cases.) If the wiring channels have sufficient capacities, a two-layer realization is always possible, with one layer for all the rows and the other for all the columns [8].

The single row approach allows a systematic analysis of the degree of difficulty of a given wiring problem. Unlike earlier approaches, it allows *a prior* estimation of the "routability" of any given instance on a fixed number of layers, in the following way:

(i) Without actually doing any wiring, it is possible to obtain sufficient conditions, which, if satisfied by the capacities of the available wiring channels, guarantee 100 percent wiring.

(ii) When 100-percent wiring is not so guaranteed, it is possible, again without doing any wiring, to get a lower bound on the fraction of the wires that can actually be laid out. This lower bound compares very well with actual wiring results obtained using current wiring techniques. In a sense, it represents the inherent difficult of the given instance.

In addition to having this analytic ability, the single row approach reduces the general problem to conceptually simpler terms. All possible input configurations are mapped into the same basic "single row" configuration. This allows us to concentrate on solving the single row wiring problem "well." Arguably, this is the most attractive feature of this approach.

This paper surveys the results of an investigation of the single row approach. Section II considers the process by which the general wiring problem is decomposed into single row wiring problems. Following [15], this is a three-step process. Each step is explained in detail and its complexity analyzed. Then, Section III considers the complexity of single row wiring problems, under a variety of optimization measures. The motivation for considering each optimization measure is explained, and the complexity of the associated single row wiring problem analyzed. Where possible, efficient algorithms are presented.

II. The Complexity of the Decomposition Process

In decomposing a multilayer rectilinear wiring problem (with a regular geometry) into a number of independent single row wiring problems, there are three sequential steps [15]:

(1) via assignment,

(2) permutation of via columns,

(3) layering (this step is meaningful only when the number of available layers exceeds two).

As stated earlier, it is assumed that the wire end-points extend through all the layers. Further, the decomposition process to be described in this section assumes that any interlayer connection will require a hole that extends through all the wiring layers.

In this section, we investigate each of the above three decomposition steps. For each step, we present relevant constraints and optimization criteria. Then, we investigate the computational complexity of carrying out that step optimally.

2.1. *Via Assignment*

A *point* specifies the location of either a component pin/pad, a via or a site of a connection to the outside world (i.e., an I/O pin/pad). A *net* is a set of points to be made electrically common. Thus, a wire is a special case of a net. With the strategic constraint that wires (i.e., point-to-point connections) can exist only between points that are either in the same row or the same column, it may not be immediately possible to appropriately decompose each net into wires. A net is considered to be *decomposable* if, for any pair of points s and t belonging to the net, there exists a sequence s, $p_1, p_2, p_3, \cdots, p_k$, t of points such that (i) the points p_i, $1 \leqslant i \leqslant k$, belong to the net also; (ii) the points s and p_1 are either in the same row or the same column, and this is true for p_k and t also; and (iii) for any 1, $1 \leqslant i \leqslant k-1$, p_i and p_{i+1} are either in the same row or the same column. If a net is not initially decomposable, it can be made so by augmenting it with free (i.e., previously unassigned) vias in such a way that the augmented net satisfies the condition for decomposability mentioned above.

(Each grid location of the point grid that is not occupied by a component pin or an I/O pin/pad is a legal via location. Initially, all the legal via locations are called *free vias*. When a free via is assigned to a net, it becomes an *assigned via*, and is no longer free. It is important to realize that only assigned vias are actually fabricated.)

The object of the via assignment phase is to add free vias to nets as necessary to decompose all the nets. Optimization measures concern via usage. Two common ones are:

(1) Minimize the total number of vias used. There are a couple of reasons for this:

(a) For mechanical and thermal reasons, vias in printed circuit boards are unreliable, and should be eliminated wherever possible. Further, they increase the fabrication costs.

(b) In the context of IC fabrication, vias are known as contact cuts. Contact cuts are used to establish electrical connections between two layers. Contact cuts are functionally the same as vias. For a contact cut to be made successfully, the photomasks used in fabricating the two layers should be in exact alignment at the contact cut location. The greater the number of contact cuts, the harder it is to guarantee exact mask alignment at each contact cut location.

(2) Minimize the number of columns of vias used. When vias are allowed to appear only column-wise, minimizing the number of via columns needed to make all the nets decomposable minimizes the area needed for the final realization. It also minimizes the number of single row wiring problems generated.

The problem of minimizing the number of via columns has received quite a bit of attention already. Tsukiyama *et al.* [17] consider a restricted version of the problem where no net has more than one point in any given column. Ting *et al.* [16] consider another restricted version, where each net is forced to use only vias from the same via column. Both [17] and [16] have shown that, with their respective restrictions, deciding whether k via columns are sufficient to make each net in a net set decomposable is NP-complete.

We have obtained some important new results for the via assignment problem.

The assumptions used in [16] are somewhat less restrictive than the ones in [17]. With the basic restriction in [16] (i.e., that all vias assigned to a net must be from the same via column), we have obtained in Section 2.1.1 a strong result than has [16]. We show that the problem of deciding whether k via columns are sufficient to decompose all the nets is NP-complete even for *fixed* k ($k = 2$). The importance of this result lies in the following. Often, NP-complete problems reduce to polynomial complexity when some of the parameters are fixed (i.e., given constant values). Our findings imply that fixing k will not make the via column minimization problem any easier.

Next, we consider the via column minimization problem again, this time without any restrictions whatsoever on via selection. Thus, nets are free to choose their vias from anywhere. We show in Section 2.1.2 that the problem of via column minimization is still NP-complete.

Finally, in Section 2.1.3, we consider the other via assignment optimization problem, that of minimizing via usage. We show that this problem is NP-complete also.

2.1.1. Via Column Minimization: Restricted Version: The decision problem corresponding to the via column minimization problem is the following:

VIACOL (Via column minimization)

Input: A set $N = \{N_1, N_2, \cdots, N_m\}$ of m nets, and a positive integer $k \leqslant m$.

Output: "Yes" if the m nets can all be made decomposable using at most k via columns; "no" otherwise.

We assume that each net is forced to get all its vias from the same via column. We shall refer to this problem as *restricted VIACOL*. It has been shown that restricted VIACOL is NP-complete for general k [16]. We shall prove that restricted VIACOL is NP-complete even for (fixed) $k = 2$. We shall refer to this problem as *restricted VIACOL* ($k = 2$).

First, we need to introduce some notation and terminology for the problem VIACOL. A *g-node* is a maximal decomposable subset of a net. Each net can be (conceptually) partitioned into one or more g-nodes. Thus a given net $N_a = \{(1,1),(1,4),(2,3),(4,3)\}$ can be partitioned into two g-nodes $G_1 = \{(1,1),(1,4)\}$ and $G_2 = \{(2,3),(4,3)\}$. See Fig. 2.

If a net contains more than one g-node, it cannot be immediately decomposable. For restricted VIACOL ($k = 2$), we need to assign exactly one via to each g-node in a net to make that net decomposable. Not all vias are assignable to a given g-node. A row is *reachable* from a g-node only if there is a point belonging to that g-node in that row. A via is assignable to a g-node only if it is in a reachable row. Fig. 2(b) shows a symbolic representation for the net of Fig. 2(a). The net is represented in terms of its g-nodes. For each g-node, the rows reachable from it are indicated by the arrows emanating from it. The broken closed path indicates that G_1 and G_2 together constitute a net (N_a).

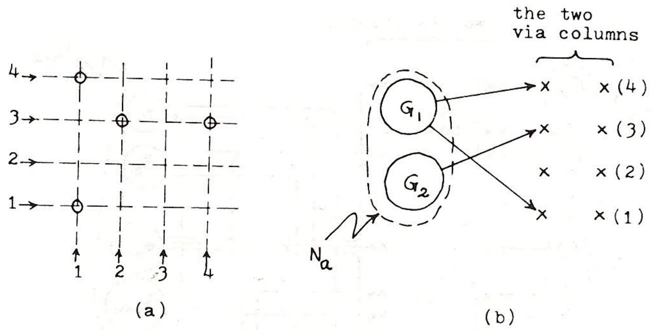

Fig. 2.

To establish the NP-completeness of restricted VIACOL ($k = 2$), we shall use a new variation of the known NP-complete problem Monotone 3-Sat. So, before launching into Theorem 1, we shall present this variation, which we shall term the restricted version, of Monotone 3-Sat and show that, in spite of this variation, it remains NP-complete.

In an arbitrary instance of the usual form of Monotone 3-Sat, we are given $U = \{x_1, x_2, \cdots, x_n\}$: a set of n Boolean variables, and a collection $C = \{C_1, C_2, \cdots, C_m\}$ of clauses over U. Each clause is the disjunct of exactly 3 literals. The literals in any given clause are either all negated or all unnegated. The question is whether there is a truth assignment for U that satisfies all the m clauses.

In the restricted version (actually, a variation) of Monotone 3-Sat, for each Boolean variable $x_i \in U$, we shall require (additionally) that the number of occurrences (over all the clauses) of the literal x_i equal the number of occurrences of the literal $-x_i$. Furthermore, we shall relax the restriction that there be *exactly* 3 literals per clause, requiring instead that each clause be the disjunct of *at most* 3 literals. This variation of Monotone 3-Sat is also NP-complete. To see this, suppose that x_i occurs r times more often than does $-x_i$. We can equalize the number of occurrences by adding new clauses $(-x_i \vee -a_j^i \vee -b_j^i)$, $1 \leqslant j \leqslant r$, and $(a_j^i \vee b_j^i)$, $1 \leqslant j \leqslant r$. The variables a_1^i, b_j^i, $1 \leqslant j \leqslant r$, are newly defined Boolean variables. Note that now, x_i occurs as often as $-x_i$, a_j^i as often as $-a_j^i$ and b_j^i as often as $-b_j^i$, $1 \leqslant j \leqslant r$. The trueth assignment $[a_j^i \leftarrow \text{true}, b_j^i \leftarrow \text{false}, 1 \leqslant j \leqslant r]$ satisfies (i.e., makes true) all these newly added clauses, without impacting the original problem in any way. Hence, the restricted version of Monotone 3-Sat is also NP-complete

Theorem 1. Restricted VIACOL ($k = 2$) is NP-complete.

Proof: Since the problem is in the class NP, all we need to show is that the restricted version of Monotone 3-Sat α restricted VIACOL ($k = 2$).

Let literals x_i and $-x_i$ occur f_i times each, $1 \leqslant i \leqslant n$. Without loss of generality, we assume that the first r clauses C_i, $1 \leqslant i \leqslant r$, contain only unnegated variables, and the remaining $m - r$ clauses C_j, $r+1 \leqslant j \leqslant m$, contain only negatived variables. We shall refer to the first r clauses as

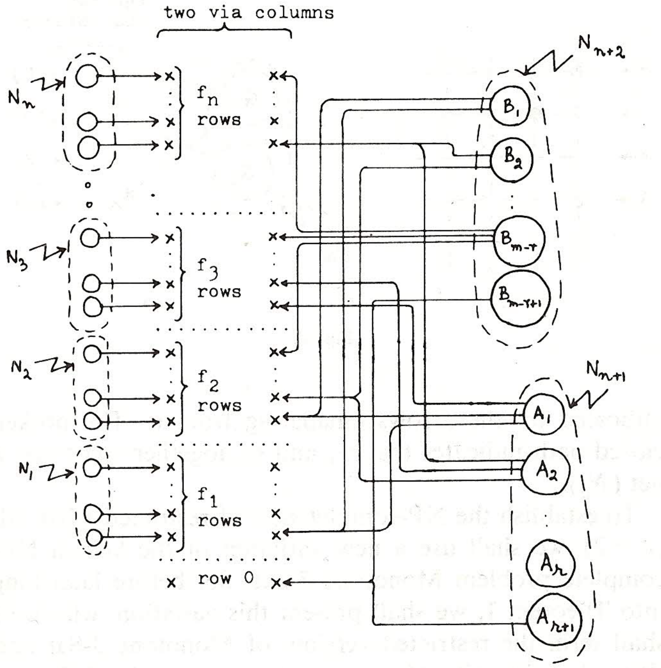

Fig. 3.

"unnegated" clauses, and to the last $m-r$ clauses as "negated" clauses.

A corresponding instance of restricted VIACOL ($k=2$) is constructed as follows. There are two via columns, each with

$$\left(\sum_{i=1}^{n} f_i + 1\right)$$

vias. We shall now construct $n+2$ nets as follows.

The nets N_i, $1 \leqslant i \leqslant n$, correspond respectively to the Boolean variables x_i, $1 \leqslant i \leqslant n$. The net N_i contains f_i g-nodes. Each of these g-nodes can reach just one row; the pth g-node in N_i can reach row

$$\left(\sum_{j=1}^{i-1} f_j + p\right), 1 \leqslant p \leqslant f_i, 1 \leqslant i \leqslant n.$$

Se Fig. 3.

Now, to decompose N_i, we need to use f_i vias. All these f_i vias have to be from the same column, because of the nature of our restriction. Picking them from one or the other column will be seen to correspond to setting the associated variable x_i to true or false.

The net N_{n+1} corresponds to the set of "unnegated" clauses C_i, $1 \leqslant i \leqslant r$. It contains $r+1$ g-nodes A_i, $1 \leqslant i \leqslant r+1$. The only row reachable from A_{r+1} is row 0. The remaining g-nodes A_i, $1 \leqslant i \leqslant r$, correspond respectively to the r "unnegated" clauses C_i, $1 \leqslant i \leqslant r$. The rows reachable from each A_i, $1 \leqslant i \leqslant r$, are determined as follows. Suppose that C_i, the clause corresponding to A_i, is $(x_a \vee x_b \vee x_c)$. Then, three rows are reachable from A_i. (In general, if C_q contains s literals, s rows are reachable from A_q). Let $z \in \{a, b, c\}$. Then, A_i can reach one of the rows reachable from a g-node in N_z. Accordingly, an arrow is drawn form A_i to the appropriate row. Also, we shall stipulate that no row should be reachable from two distinct g-nodes A_j and A_k. (Clearly, there are enough rows, and each net N_i, $1 \leqslant i \leqslant n$, has enough g-nodes, so that this stipulation does not cause any difficulties). Figure 3 illustrates the case where $C_1 = (x_1 \vee x_2 \vee x_3)$ and $C_2 = (x_2 \vee x_3 \vee x_n)$. Note that, after the net N_{n+1} has been constructed as specified above, each row other than row 0 is reachable from exactly two g-nodes, one of which belongs to one of the nets N_i, $1 \leqslant i \leqslant n$, and the other to the net N_{n+1}.

The net N_{n+2} corresponds to the set of "negated" clauses C_j, $r+1 \leqslant j \leqslant m$. It contains $m-r+1$ g-nodes B_i, $1 \leqslant i \leqslant m-r+1$. The only row reachable from B_{m-r+1} is row 0. The remaining $m-r$ g-nodes correspond, respectively, to the $m-r$ "negated" clauses. The rows reachable from each B_i, $1 \leqslant i \leqslant m-r$, are determined exactly as described earlier for the g-nodes, A_j, $1 \leqslant j \leqslant r$; the fact that the variables are negated now, rather than unnegated, makes no difference in determining the reachable rows. Fig. 3 depicts the case where $C_{r+1} = (-x_1 \vee -x_2)$, $C_{r+2} = (-x_2 \vee -x_n)$ and $C_m = (-x_2 \vee -x_3 \vee -x_n)$. After the net N_{n+2} has been constructed, each row q, $1 \leqslant q \leqslant (\Sigma_{i=1}^{n} f_i)$, is reachable from exactly three g-nodes. One of these belongs to one of N_i, $1 \leqslant i \leqslant n$, the second to N_{n+1} and the third to N_{n+2}.

Row 0 is reachable from A_{r+1} and B_{m-r+1}. These two g-nodes have to be assigned vias from row 0, as there is no other choice. So, these two vias have to be from different columns. It is clear that this forces N_{n+1} to get its $r+1$ vias from a different via column from the one from which N_{n+2} gets its $m-r+1$ vias.

Now, suppose that all the $n+2$ nets are decomposable using just these two via columns. A satisfying truth assignment for $U = \{x_1, x_2, \cdots, x_n\}$ can be obtained from the optimal via assignment as follows. Without loss of generality, assume that the $r+1$ vias for N_{n+1} are from the right column. So the $m-r+1$ vias for N_{n+2} are from the left column. Set x_i to true (false) if N_i gets its f vias from the left (right) column, $1 \leqslant i \leqslant n$.

This truth assignment will satisfy all the m clauses. To see this, suppose an "unnegated" clause $C_q = (x_a \vee x_b \vee x_c)$ is not satisfied. Then it must be that $x_a = x_b = x_c =$ false. So each of the nets N_a, N_b, and N_c use vias from the right column. However, one of these very vias has to have been assigned to A_q, if all the $r+1$ vias for N_{n+1} are from the right column. Since two nets cannot use the same via, this is a contradiction. So every "unnegated" clause must be satisfied. A similar argument can be used to show that every "negated" clause must also be satisfied.

Next, suppose that there is a satisfying truth assignment for U. A decomposition of $N_1, N_2, \cdots, N_{n+2}$ using just two via columns is obtained as follows. For $1 \leqslant i \leqslant n$, if $x_i =$ true(false), use the vias in the left (right) column to decompose N_i. N_{n+1} can be decomposed using vias in the right column only, as follows. A_{r+1} is assigned the right-column via in row 0. For each "unnegated" clause C_j, $1 \leqslant j \leqslant r$, there has to be a variable $x_t \in C_j$ such that x_t is true. From the way A_j was defined, it can reach a row p such that row p is reachable from N_t too. Since $x_t =$ true, N_t uses only vias from the left column. So, A_j can be assigned the right-column via in row p. In this manner, every g-node A_i, $1 \leqslant i \leqslant r+1$, can be assigned a via from the right via column. So, N_{n+1} is decomposable, and it uses vias from the right column only.

Using a similar argument, we can show that every g-node B_i, $1 \leqslant i \leqslant m-r+1$, can be assigned a via from the left via column. So, N_{n+2} is also decomposable.

Thus the constructed restricted VIACOL ($k=2$) instance has answer "yes" if and only if there is a satisfying truth assignment for the restricted Monotone 3-Sat instance. Hence, this restricted version of Monotone 3-Sat α restricted VIACOL ($k=2$). ■

2.1.2. Via Column Minimization: General Version:

We now remove the restriction that all the vias assigned to a net must be from the same via column. The problem VIACOL ($k=2$) is the decision problem concerning the feasibility of decomposing a net using just 2 via columns. There are now no restrictions on how the vias assigned to each net may be chosen. So this is a less constrained problem than the one considered in [16].

Theorem 2. VIACOL ($k=2$) is NP-complete.

Proof: Clearly, the problem is in the class NP. To establish that it is NP-hard, we shall extend the proof of Theorem 1.

First of all, note that if a net uses vias from both via columns, one or more "extra" (i.e., otherwise unused) vias is needed to establish a connection between the two via columns. We shall make use of the fact that if such "extra" vias are not available, every net will be forced to choose all its vias from the same via column.

Now, consider the constructed restricted VIACOL instance in the proof of Theorem 1. There are

$$2\Big(\sum_i f_i + 1\Big)$$

vias arranged in two via columns. We shall refer to these vias as "old" vias, as we shall be adding some more "new" vias. To decompose the nets N_i, $1 \leqslant i \leqslant n+2$, we need at least

$$\Big(\sum_j f_j\Big) + (r+1) + (m-r+1) = \sum_j f_j + m + 2$$

"old" vias. This leaves at most

$$2\Big(\sum_j f_j + 1\Big) - \Big(\sum_j f_j + m + 2\Big) = \sum_j f_j - m$$

"old" vias unused. There are two possible situations where this many "old" vias are in fact unused after decomposing the nets N_i, $1 \leqslant i \leqslant n+2$:

(i) Each net is assigned all its vias from the same via column; or

(ii) For each net that is assigned vias from different columns, there is one extra row of "new" vias (to be discussed next) available to establish a connection between the two via columns.

We now add $\Sigma_j f_j - m$ new nets N_j^*, $1 \leqslant j \leqslant (\Sigma_i f_i - m)$, to sweep up some of the old vias. With each new net N_j^*, we add a row of "new" vias. N_j^* contains two g-nodes α_j and β_j. The only row reachable from α_j is the corresponding newly added via row. From β_j, all the "old" via rows are reachable. Figure 4 illustrates this construction for $\Sigma_j f_j - m = 2$.

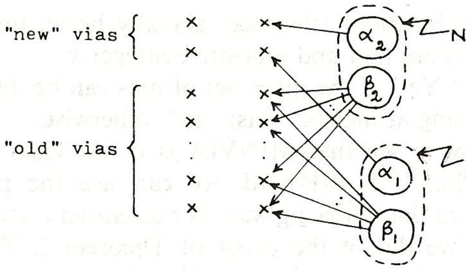

Fig. 4.

Clearly, the nets N_j^*, $1 \leqslant j \leqslant (\Sigma_i f_i - m)$, can be decomposed if and only if at least $\Sigma_i f_i - m$ "old" vias are not needed for decomposing the nets N_i, $1 \leqslant i \leqslant n+2$. In conjunction with the fact that at most $\Sigma_i f_i - m$ "old" vias can be left unused upon decomposing the nets N_i, $1 \leqslant i \leqslant n+2$, this means that there can be no unused "old" vias. Also, it should be clear that there is no "extra" row among the newly added vias, as one via from each "new" via row is assigned to some net N_a^*. In light of our earlier discussion, this means that each net N_i, $1 \leqslant i \leqslant n+2$, is forced to choose all its vias from the same via column.

The proof is now very simply completed. If the restricted Monotone 3-Sat instance has a satisfying truth assignment, the nets N_i, $1 \leqslant i \leqslant n+2$, can be decomposed using just two via columns as described in the proof of Theorem 1. As the nets N_i, $1 \leqslant i \leqslant n+2$, need just $\Sigma_i f_i + m + 2$ "old" vias, the nets N_j^*, $1 \leqslant j \leqslant (\Sigma_i f_i - m)$, can also be decomposed with two via columns, since $\Sigma_i f_i - m$ "old" vias are still available after decomposing all the N_i's.

On the other hand, if the nets N_j, $1 \leqslant j \leqslant n+2$, and N_j^*, $1 \leqslant j \leqslant (\Sigma_i f_i - m)$, can be decomposed using just the two via columns, it must be the case that each of the nets N_i, $1 \leqslant i \leqslant n+2$, has all its vias from the same via column. So, following the proof of Theorem 1, there is a satisfying truth assignment for U.

Thus, the restricted version of Monotone 3-Sat α unrestricted VIACOL ($k=2$). Therefore, this latter problem is NP-hard, and also NP-complete. ■

2.1.3. Via Minimization:

As before, we are given a set of nets and a set of legal via locations, located within the wiring surface whose extent has been predetermined. Once again, we wish to obtain a suitable decomposition into wires for each net. In order to do so, we will augment some or all nets with free vias.

Typically, the number of legal via locations will far exceed the actual number of vias required to decompose all the nets. Since only the vias that are actually assigned get fabricated, it is desirable to minimize the number of vias assigned. Minimizing the number of vias that have to be fabricated serves to reduce the manufacturing time and cost, at the same time increasing reliability.

The associated decision problem is the following:

MINVIA (Minimize the total number of vias used)

Input: A set $N = \{N_i | 1 \leqslant i \leqslant m\}$ of m nets, a set of legal via locations (this may actually be an implicit specification) and and a positive integer k'.

Output: "Yes" if the given set of nets can be decomposed using at most k' vias; "no" otherwise.

It is easy to see that MINVIA is in the class NP. To establish that it is NP-hard, we can use the proof of Theorem 2 in the following way. We construct a set of nets, exactly as we did in the proof of Theorem 2. Then, we construct the legal locations for the vias as appearing in two columns. The resultant instance of MINVIA is then identical to the problem instance constructed in proving Theorem 2.

We see that with k' set to

$$\sum_i f_i + m + 2 + 2\Big(\sum_i f_i - m\Big) = 3\sum_i f_i - m + 2$$

the value of k' is small enough that each net is forced to choose all its vias from the same via column.

Theorem 3. MINVIA is NP-complete.

Proof: Follows from the above discussion. ■

2.2. Permutation of Via Columns

This phase of the decomposition process is meaningful only when the vias appear column-wise. The locations of the pins are probably unalterable, since they have been determined by the fixed placement procedure. This, however, is not so for the via columns. At the end of the via assignment phase, the number of via columns needed has been minimized. However, the locations of the via columns have not been fixed. In particular, we can interchange the locations of two via columns while preserving all the interconnections; such an interchange will not violate the validity of the decomposition of any net. An interchange could actually improve the final layout.

The *thickness* of a decomposition is defined as max{number of wires that cross the interval between an adjacent pair of columns of points}. The thickness is a useful norm is evaluating a decomposition. Minimizing the thickness results in obvious improvements.

The problem of finding a permutation of the via columns that minimizes the thickness is NP-hard. The problem is identical to the NP-hard problem One-Dimensional Placement [3]. Each via column corresponds to a component b_i of One-Dimensional Placement. The rest of the correspondence between the two problems follows easily.

2.3. Layering

When the number of layers available for wiring exceeds two, layering, the third phase in the decomposition process, is used to effect an even distribution of the conductors among the various layers. Usually, half the available layers are used for realizing the row problems, and half for the column problems.

The layering problem to be considered is the following. We are given a set $V = \{1,2,\cdots,n\}$ of n collinear points, a set $L = \{N_1, N_2,\cdots,N_m\}$ of m nets over V, and integers c_V and c_L. c_V is the capacity of one of the adjacent wiring channels and c_L the capacity of the other. We wish to decompose L into r disjoint subsets $L_1, L_2,\cdots,L_r$ such that:

(1) All the nets in each L_i, $1 \leqslant i \leqslant r$, can be laid out on a single layer, subject to all the usual constraints, without causing an overflow in either of the two adjacent wiring channels.

(2) r is a minimum over all choices of the L_i.

This single row layering problem is NP-hard. For a proof, see [13].

Reference [20] considers a restricted version of the single row layering problem, in which $c_V = c_L = 2$; a heuristic algorithm for this restricted version is proposed therein.

III. The Complexity of Single Row Wiring

In the single row wiring problem, we are given n points (they may be pins and/or vias) $V = \{1,2,\cdots,n\}$ evenly spaced along a line, and a set of nets $L = \{N_1, N_2,\cdots,N_m\}$ over V. Without loss of generality, we may assume that the line of points is oriented horizontally. The nets satisfy the following conditions:

(i) $N_i \cap N_j = \phi$, $i \neq j$

(ii) $\cup N_i = \{1,2,\cdots,n\}$.

Laying out a single row wiring problem consists of defining conductor paths in order to realize all the nets, subject to the following constraints:

(i) Each conductor path is made up of horizontal and vertical segments only.

(ii) Conductor paths do not cross.

(iii) The layout of each net should be free of "backward moves". In other words, it should not be possible to draw a vertical line anywhere, that intersects more than one conductor segment of any given net.

A layout that satisfies the above constraints is called a *realization*.

We now introduce some terminology. The wiring channel above the line of points is called the *upper street*, and the one below, the *lower street*. For any given realization, the number of wiring tracks needed in the upper street is called the *upper street congestion* of the realization, and is denoted by c_V. Similarly, we can define a *lower street congestion* c_L. The quantity $\max\{c_V, c_L\}$ is called the *width* of the realization. When the layout of a conductor path contains a vertical segment from the upper street to the lower street (or vice versa), the path is said to make an *interstreet crossing*. The *between-nodes congestion* c_B of a realization is max{number of interstreet crossings between an adjacent pair of points}. In laying out a wire the number of right-angled *bends* in the layout of that wire is related to the number of interstreet crossing made by it, as follows: the number of bends = 2 (number of interstreet crossings)+2. In general, $p-1$ wires are required to realize a net with p points. Let p_i be the number of points in net N_i, $1 \leqslant i \leqslant m$. Then, for the complete realization, the total number of bends = 2 (number of interstreet crossings)+ $2\Sigma_i(p_i - 1)$. (A T connection counts as two-right-angled bends.)

In what follows, we consider the single row wiring problem under a variety of constraints and optimization measures. In each case, the motivation for considering that particular constraint or optimization measure is explained. This is followed by a complexity analysis of the resulting single row wiring problem. Where possible, efficient algorithms are presented.

3.1. Minimizing the Width

Minimizing the width of the layout results in the most compact realization possible. A consideration of the discussion in Section I establishes that minimizing the width of the realization of each single row wiring problem improves the chances of overall success. This is because adjacent single row wiring problems on a given layer are less likely to interfere with each other. The corresponding decision problem is the following:

MINWIDTH (minimize the width of the realization)
Input: A set of m nets over n points, and a positive integer k
Output: "Yes" if there is a realization with width at most k; "no" otherwise.

It has recently come to our attention that is problem is NP-complete [1]. (The transformation is from the known NP-complete problem bandwidth). Since the problem is of great practical importance, it has continued to attract quite a bit of attention.

The minimization version problem has been studied extensively by Kuh *et al.* [8]. They have obtained necessary and sufficient conditions that characterize optimal layouts. However, they have not succeeded in presenting an efficient algorithm for the problem.

In [12], among other things, we have developed an $O((2k)!kn\log k)$ algorithm for MINWIDTH. (A description of the algorithm also appears in [12]. The algorithm is named POSSIBLE. If the algorithm outputs "yes," the layout can also be obtained. This algorithm is in the nature of a "usually good" approach to the problem of finding the minimum width realization. In most applications, k is small (about 3 or 4), so that the complexity is not unbearable.

It is worth mentioning that, for the special case $k=2$, Tsukiyama *et al.* [19] have developed an $O(mn)$ algorithm for MINWIDTH. This algorithm is not generalizable for $k>2$. Furthermore, it is asymptotically inferior to (i.e., slower than) algorithm POSSIBLE. For any fixed k, the latter algorithm has a complexity of just $O(n)$.

There is a more generalized version of the problem MINWIDTH, in which the conductors used in realizing the nets may have nonuniform widths. Specifically, in this generalized MINWIDTH, we associate with each net N_i a positive integer t_i, $1\leqslant i\leqslant m$. t_i is the width of the various conductor paths to be used in laying out N_i, $1\leqslant i\leqslant m$. The basic objective is to obtain a minimum width layout. (The width is now a weighted conductor count.)

This is an NP-hard problem, and is easily shown to be so. Consider an arbitrary instance of the known NP-complete problem Partition. We are given a multiset $A=\{a_1,a_2\cdots a_n\}$ of positive integers. The question is whether there exists a partition of A into A_1 and A_2 such that

$$\sum_{a_i\in A_1} a_i=\sum_{a_j\in A_2} a_j.$$

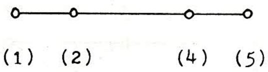

Fig. 5.

An equivalent instance of the generalized MINWIDTH problem is as follows:
(i) $V=\{1,2\cdots 2n\}$: a set of $2n$ vertices;
(ii) nets $N_1,N_2\cdots N_n$, where $N_i=\{i,2n+1-i\}$;
(iii) net widths $t_i=a_i$, $1\le i\le n$.

Clearly, there exists a layout with upper street usage = lower street usage = $(\Sigma a_i)/2$ if and only if the corresponding partition instance has the answer "yes."

3.2. Minimizing the Between-Nodes Congestion

In dual in-line IC packages, the separation between the two rows of pins is much greater than the separation between adjacent pins in a row. In view of this, it is quite possible that we could encounter situations in which adjacent single row wiring problems are relatively far apart; however, in a given single row wiring problem, the points could be quite closely spaced. In such situations, it is more important to limit the between-nodes congestion of a realization, than it is to limit the width.

The decision problem to be considered is the following:

MINCB (Minimize the between-nodes congestion).
Input: A set of m nodes over n points, and a positive integer k.
Output: "Yes" if there is a realization with between-nodes congestion c_B at most k; "no" otherwise.

We shall show that MINCB is NP-complete (see Theorem 4). In doing so, we need to use the interval graphical formulation of [8] for the single row wiring problem. So, we now introduce this rather elegant formulation.

An *interval line* is a horizontal straight line with a left end-point, a right-end point and intermediate points. Each point in the interval line can be characterized by its x-coordinate. Fig. 5 shows an interval line with 4 points. Their x-coordinate values are indicated in parentheses.

Any subset S of the set $\{1,2,\cdots n\}$ can be represented by an interval line. Suppose $S=\{1,3,4\}$. S can be represented by an interval line with 3 points. The left end-point would be at $x=1$, the right end-point at $x=4$ and the one intermediate point at $x=3$. Thus the interval line in Fig. 5 can be thought of as representing the subset $\{1,2,4,5\}$.

Since each net N_i, $1\le i\le m$, is a subset of $\{1,2,\cdots n\}$, it can be represented by an interval line. With this representation for the net list, it is possible (as will be explained soon) to associate a realization with each ordering of the m interval lines. Such a representation for a realization is called a *conceptual realization*. Fig. 6(a) shows a conceptual realization for the net set $N=\{N_1,N_2,N_3,N_4\}$, where $N_1=\{2,5,7\}$, $N_2=\{3,6\}$, $N_3=\{4,8\}$ and $N_4=\{1,9\}$. The

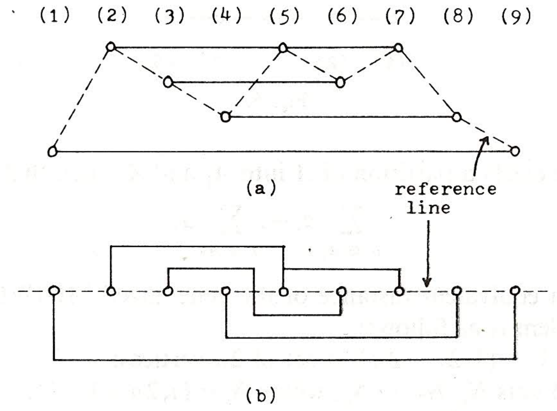

Fig. 6.

dashed line is called the *reference line*, and is obtained by connecting the ith and $(i+1)$th points, $1 \le i \le n-1$. (The ith point is the point with the ith smallest x-coordinate value. Note that this does not necessarily mean that the point is at $x = i$.

The actual realization corresponding to a conceptual realization is obtained by (conceptually) straightening out the reference line to form the line of points, and then topologically mapping each interval line into paths in the upper and lower streets. Those portions of an interval line that are above (below) the reference line are mapped into paths in the upper (lower) street. Fig. 6(b) is an actual realization corresponding to the conceptual realization in Fig. 6(a).

It is easy to see that a different ordering of the m interval lines will usually result in a different actual realization. (It is not always necessary that a different actual realization result. For instance, suppose there is an ordering in which there is a pair of adjacent interval lines that do not overlap. Exchanging the positions of the two lines gives a new ordering, which has the same actual realization as the original ordering.)

The reference line is made up of $n-1$ segments. The segment connecting the ith and $(i+1)$th points will be denoted by $\langle i, i+1 \rangle$, $1 \le i < n$. Given a conceptual realization, the number of interval lines that cross any given segment $\langle j, j+1 \rangle$ is the number of interstreet crossings between the jth and $(j+1)$th points in the corresponding actual realization. In terms of the conceptual realization, the between-nodes congestion c_B is max {the number of interval lines that cross the reference line segment $\langle j, j+1 \rangle | 1 \le j < n$}.

We shall now prove that MINCB is NP-complete. We shall use the conceptual representation, and calculate c_B as described above. It is easy to see that the decision problem MINCB is in the class NP. To establish its NP-completeness, we shall show that the known NP-complete problem Bandwidth α MINCB. Some aspects of the reduction used are similar to one used in [18].

Theorem 4. MINCB is NP-complete.

Proof: All we need to show is that bandwidth α MINCB.

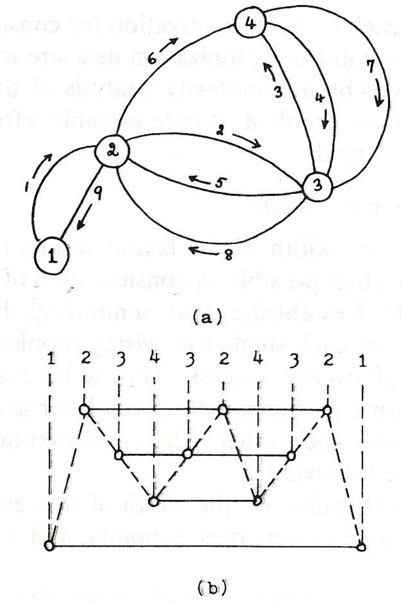

Fig. 7.

An arbitrary instance of bandwidth is as follows. We are given an undirected connected graph $G = (V, E)$, and a positive integer $K \le |V|$. We wish to find a linear ordering $f\colon V \to \{1, 2, \cdots |V|\}$ of the vertices in V such that for each edge $\{u, v\} \in E$, $|f(u) - f(v)| \le K$. In order to construct a corresponding instance of MINCB, we proceed as follows.

If all the vertices in a graph are of even degree, then the graph has an Euler walk (i.e., a closed path that goes through each edge in the graph exactly once). If, in the Bandwidth instance, the given graph G is not Eulerian, we can create an Eulerian multigraph $G' = (V, E')$ from the graph G by duplicating each edge in E; i.e., for each edge $\{u, v\} \in E$, there are two edges $\{u, v\}, \{u, v\} \in E'$. Now, each vertex in G' is of even degree, and so there exists an Euler walk through G'.

We now find an Euler walk through G (or through G', if G is not Eulerian). In general, the Euler walk that is found will not be unique. However, this has no effect on the proof.

An Euler walk can be conveniently represented by the vertex sequence encountered in making that walk. Fig. 7(a) shows an undirected Eulerian multigraph. The numbered arrows indicate the order in which edges are traversed in an Euler walk beginning and ending at vertex 1. The corresponding vertex sequence is $A = \{1, 2, 3, 4, 3, 2, 4, 3, 2, 1\}$.

From such a vertex sequence, we construct a set of interval lines as shown in Fig. 7(b). This results in one interval line being constructed for each vertex in the graph. Note that the vertex sequence does not determine the (vertical) ordering of the interval lines. (In fact, this ordering will be seen to correspond to the required ordering f of the vertices). Also, note that there is a one-to-one correspondence between edges in G and segments of the reference line. These $|E|$ segments of the reference line are called "old" segments, to differentiate them from "new"

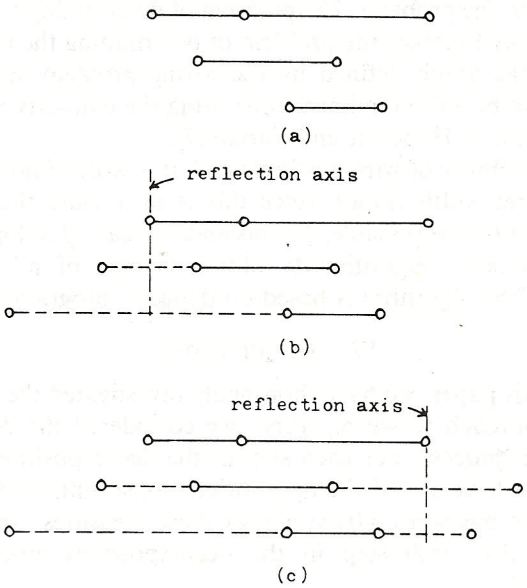

Fig. 8.

segments created by the additional construction discussed next.

Each interval line is extended in both directions using what we shall term a *reflection extension*. A leftward reflection extension is done as follows. A vertical *reflection axis* is drawn through the leftmost point, i.e., the point with the smallest x-coordinate value. To each interval line whose left end-point is to the right of the reflection axis, we add a new point that is as far to the left of the reflection axis as the left end-point is to the right of it. This new point becomes the new left end-point. Fig. 8 illustrates the reflection extension. Fig. 8(a) shows a set of interval lines, such as might be obtained from a vertex sequence of an Euler walk. Fig. 8(b) shows a leftward reflection extension of the interval lines. A rightward reflection extension can be defined in an analogous manner. Fig. 8(c) shows the set of interval lines of Fig. 8(a) after a leftward reflection extension and a rightward reflection extension.

The effects of these two extensions are as follows. In a given ordering R of the $|V|$ interval lines, let $M(i)$ denote the position in R of the interval line to which the ith point belongs. (We shall sometimes refer to the ith point simply as point i). Then,

(1) For any "old" segment $\langle p, p+1\rangle$ of the reference line, the number of interval lines that cross $\langle p, p+1\rangle$ equals $|M(p)-M(p+1)|-1$. Obviously, the number cannot be greater than this. For the number to be less than this, it must be that there is a point d such that:

(i) either $M(p)<M(d)<M(p+1)$ or $M(p)>M(d)>M(p+1)$; and

(ii) the left (right) end-point of the interval line to which d belongs is to the right (left) of the point $p+1(p)$. The nature of the leftward and rightward reflection extensions makes (ii) impossible. Thus the number of interval lines crossing $\langle p, p+1\rangle$ equals $|M(p)-M(p+1)|-1$.

(2) For each "new" segment $\langle q, q+1\rangle$, there is an "old" segment $\langle r, r+1\rangle$ such that $M(q)=M(r+1)$ and $M(q+1)=M(r)$. Furthermore, the number of interval lines that cross $\langle q, q+1\rangle$ is less than or equal to the number of interval lines that cross $\langle r, r+1\rangle$. Consequently, it is not necessary to consider the contributions of the "new" segments of the reference line when calculating the between-nodes congestion c_B for a conceptual realization.

The proof is now easily completed. Given an arbitrary instance of Bandwidth, we construct a corresponding set of interval lines as described above. We then set k (of the MINCB instance) to $K-1\{K\in$ bandwidth). Clearly, this construction takes only a polynomial amount of time.

Now, suppose that there is a conceptual realization with $c_B\le k$. An ordering f of V is obtained as follows. For each $v\in V$, set $f(v)$ to x if the corresponding interval line is the xth one in the conceptual realization. To see why this yields the proper f, consider the following argument. For each edge $\{u,v\}\in E$, there exists a corresponding "old" segment $\langle z, z+1\rangle$. Now, it must be the case that $|M(z)-M(z+1)|-1\le k$, since $c_B\le k$. Thus $|M(z)-M(z+1)|\le k+1=K$. Also, we know that $|M(z)-M(z+1)|=|f(u)-f(v)|$. Hence, for each $\{u,v\}\in V$, $|f(u)-f(v)|\le K$.

Next, suppose that there is a function f such that $|f(u)-f(v)|\le K$ for each $\{u,v\}\in E$. A conceptual realization with $c_B\le k$ can be obtained as follows. For each vertex $v\in V=\{1,2,\cdots\}$, if $f(v)=j$, the interval line corresponding to v is the jth one in the conceptual realization. That this yields a realization with $c_B\le k$ can be justified as follows. Each "old" segment $\langle z, z-1\rangle$ corresponds to some edge $\{u,v\}\in E$. Thus, the ordering procedure suggested above makes $|f(u)-f(v)|=|M(z)-M(z+1)|$. Since $|f(u)-f(v)\le K$, $|M(z)-M(z-1)\le K$. Hence $|M(z)-M(z+1)|-1\le K-1=k$. Since this holds true for every "old" segment, $c_B\le k$.

Hence, the constructed MINCB instance has an answer "yes" if and only if the (arbitrary) Bandwidth instance does too. So, Bandwidth α MINCB, and MINCB is NP-hard (and also NP-complete).

3.3 Minimizing the Total Number of Bends

Here, we are concerned with finding a realization that minimizes the total number of bends (in all the wires). This optimization measure is important in at least three contexts.

(1) In fabricating microwave and millimetric wave integrated circuits, the conductor paths in the metallization layer act as waveguides, rather than as simple electrical wires. These paths are called microstrip lines. The effect of a bend in a microstrip line is the creation of reflections, and this reduces the transmission efficiency of the line. This forces the drivers to send out stronger signals in order to effect a proper transmission. So, in this context, counting the total number of bends in a realization can help form a rough estimate of the power requirements of the chip. Minimizing the total number of bends will help reduce the power consumption of the integrated system.

(2) As mentioned earlier, the total number of bends in a realization is intimately related to the total number of interstreet crossings. If external circumstances (e.g., a fixed

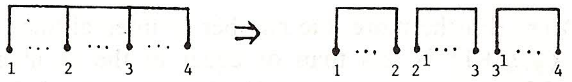

Fig. 9.

placement procedure) do not force the point locations to be fixed, a realization that minimizes the total number of interstreet crossings can be followed by a suitable relocation of the point positions along the lines of points. The result is a "minimum length" realization, i.e., a realization in which the separation between the two extreme points is minimized.

The corresponding decision problem is MINBENDS: Given a single row wiring problem, is there a realization with at most k bends totally? Note that it is essential to have $k \geqslant n$ (the number of points), since the geometry of the single row wiring problem dictates that any realization must have at least n bends.

Tsukiyama *et al.* [18] have shown that finding a realization that minimizes the total number of interstreet crossings is NP-complete. This establishes the NP-completeness of MINBENDS.

3.4. Limiting the Number of Bends in Each Wire

In fabricating high-frequency IC's, each bend in a microstrip line results in reflections of the electrical signal. So the greater the number of bends in a microstrip line, the lower is its transmission efficiency. This forces the transmitting device to output more power in order to accomplish the transmission. Limiting the maximum number of bends in the layout of any microstrip line helps alleviate this problem.

The decision problem to be considered here is KBENDWIRE: Given an instance of the single row wiring problem, is there a realization with at most k bends in the layout of each wire? Because of the single row geometry, we require that $k \geqslant 2$.

It is not known whether KBENDWIRE is NP-complete for general k. However, for the special case $k = 2$, it is easy to find a polynomial time algorithm for KBENDWIRE. Note that with $k = 2$, it is not possible for (the layout of) any wire to make interstreet crossings. This problem has been considered before, by [2], but no satisfactory solution was proposed therein. The problem was of much interest earlier, when design rules did not permit any etch paths between adjacent pins of dual-in-line IC packages.

The restriction that wires cannot make interstreet crossings allows one to decompose nets that contain more than two nodes into a set of nets, each containing exactly two nodes. See Figure 9.

It is not too difficult to formulate a linear time algorithm to determine whether or not a set of nets can be wired with no interstreet crossings. This algorithm assumes that nets have already been decomposed such that each net contains exactly two nodes.

Let us assume without loss of generality that the n points are numbered $1, 2, \cdots, $n from left to right. We now add the following "new" nets: $\{1,2\}$, $\{2,3\}$, $\{3,4\}, \cdots, \{n-1,n\}$, $\{n,1\}$ to the problem. The problem of determining feasibility simply becomes the problem of determining the planarity of the graph defined by the wiring problem instance. This can be solved in linear time using the planarity testing algorithm of Hopcroft and Tarjan [7].

If feasibility of wiring is indicated, it is worth finding the minimum width layout, since this is in a sense the most compact layout possible. To this end, we have developed in [10] an $0(n^2)$ algorithm to obtain layouts of minimum width. The algorithm is based on dynamic programming.

IV. Conclusions

In this paper, we have thoroughly investigated the single row approach to wiring. First, we considered the decomposition process. For each step in the decomposition process, we have stated the appropriate constraints and optimization measures. Given any of these measures, we have shown that each step in the decomposition process is NP-hard.

Then, we considered the complexity of single row wiring. The most reasonable optimization measure here is the width of the realization. While it is known that finding the minimum width realization is NP-hard, we have obtained (elsewhere, in [10], [12]) a "usually good" algorithm for this problem. Other important optimization measures are the between-nodes congestion and the total number of bends. For each of these measures, the single row wiring problem is NP-hard. Then, we consider the single row wiring problem when each wire is restricted to have at most k bends in its layout. For the special case $k = 2$, we have outlined an efficient algorithm for the single row wiring problem.

From the findings in this paper, one may conclude that the single row approach allows a conceptual simplification of the general rectilinear wiring problem. However, this conceptual simplification is not accompanied by a reduction in the computational complexity.

References

[1] P. B. Arnold, "Complexity results for circuit layout on double-sided printed circuit boards," Undergraduate thesis submitted to the Dept. of Applied Mathematics, Harvard University, May 1982.

[2] J. C. Foster, "A lookahead router for multilayer printed wiring boards," in *Proc. 16th Des. Automation Conf.*, pp. 486–493, 1979.

[3] S. Goto, I. Cederbaum and B. S. Ting, "Suboptimum solution of the backboard ordering with channel capacity constraint," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 645–652, Nov. 1977.

[4] A. Hashimoto and J. Stevens, "Wire routing by optimizing channel assignment within large apertures," in *Proc. 6th Des. Automation Conf.*, pp. 155–169, 1971.

[5] D. W. Hightower, "A solution to line routing problems in the continuous plane," in *Proc. 6th Des. Automation Conf.*, pp. 1–24, 1969.

[6] R. B. Hitchcock, "Cellular wiring and the cellular modelling technique," in *Proc. 6th Des. Automation Conf.*, pp. 25–41, 1969.

[7] J. Hopcroft and R. Tarjan, "Efficient planarity testing," *J. Ass. Comput. Mach.*, vol. 21, no. 4, pp. 549–568, Oct. 1974.

[8] E. S. Kuh, T. Kashiwabara and T. Fujisawa, "On optimum single row routing," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 361–368, June 1979.

[9] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Trans. Electronic Computers*, vol. EC-10, no. 3, pp. 346–365, Sept. 1961.

[10] R. Raghavan and S. Sahni, "Single row routing," Univ. of Minnesota, MN, Tech. Rep. 80-22, 1980.

[11] R. Raghavan, J. P. Cohoon, and S. Sahni, "Manhattan and rectilinear wiring," Univ. of Minnesota, MN, Tech. Rep. 81-5, 1981.

[12] R. Raghavan and S. Sahni, "Optimal single row router," *Proc. 19th ACM IEEE Des. Automation Conf.*, pp. 38–45, June 1982.

[13] S. Sahni, A. Bhatt, and R. Raghavan, "Complexity of design automation problems," Univ. of Minnesota, MN, Tech. Rep. 80-23.

[14] H. C. So, "Some theoretical results on the routing of multilayer printed wiring boards," in *Proc. IEEE Symp. on Circuits and Systems*, pp. 296–303, 1974.

[15] B. S. Ting and E. S. Kuh, "An approach to the routing of multilayer printed circuit boards," in *Proc. IEEE Symp. on Circuits and Systems*, pp. 902–911, 1978.

[16] B. S. Ting, E. S. Kuh and A. Sangiovanni-Vincentelli, "Via assignment problem in multilayer printed circuit board," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 261–271, Apr. 1979.

[17] S. Tsukiyama, I. Shirakawa and S. Asahara, "An algorithm for the via assignment problem in multilayer backboard wiring," *IEEE Trans. Circuits Syst.*, vol. CAS-26, pp. 369–377, June 1979.

[18] S. Tsukiyama and E. S. Kuh, "Double-row planar routing and permutation layout," Univ. California, Berkeley, Electronics Res. Lab., Memo. UCB/ERL M80/46, July 1980.

[19] S. Tsukiyama, E. S. Kuh, and I. Shirakawa, "An algorithm for single-row routing with prescribed street congestions," *IEEE Trans. Circuits Syst.*, vol. CAS-27, Sept. 1980, pp. 765–772.

[20] ____, "On the layering problem of multilayer PWB wiring," in *Proc. 18th Des. Automation Conf.*, June 1981.

Raghunath Raghavan was born in Bangalore, India, on October 12, 1955. He received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, India, in 1977, and the M.S. and Ph.D. degrees in computer science from the University of Minnesota, Minneapolis, both in 1981.

He is currently with the Major Systems Division of Sperry Univac, Roseville, MN, where he is engaged in research in the area of computer-aided design. He is also on the adjunct faculty of the Computer Science Department with the University of Minnesota, Minneapolis, MN. His current research interests are computer-aided design, complexity theory and the theory of computer algorithms.

Sartaj Sahni (M'79) received the Ph.D. degree in computer science from Cornell University, Ithaca, NY, in 1973.

Currently, he is a Professor with the Department of Computer Science, University of Minnesota, Minneapolis, MN. He is coauthor of the texts *Fundamentals of Data Structures*, and *Fundamentals of Computer Algorithms*, as well as the author of the text *Concepts in Discrete Mathematics*. He has published in the *Journal of the Association for Computing Machinery*, *JCSS SICOMP*, *IEEE TRANSACTIONS ON COMPUTERS*, *ACM TOMS*, *Operations Research*, the *International Journal on Theoretical Computer Science*, *Mathematics of Operations Research*, and the *Journal of Statistical Computation and Simulation*. His publications are concerned with the design and analysis of efficient algorithms for sequential and parallel computers, as well as design automation.

0098-4094/84/0500-0472$01.00 ©1984 IEEE