Stretching and three-layer wiring planar layouts *

Teofilo Gonzalez and Si-Qing Zheng

Department of Computer Science, University of California, Santa Barbara, CA 93106, U.S.A.

Revised 18 April 1988

Abstract. It is well known that every planar layout is four-layer wirable. If we decrease the number of layers, there are planar layouts that cannot be wired in three layers and the problem of determining whether a planar layout is three-layer wirable is NP-complete. A planar layout may be stretched vertically (horizontally) by introducing empty rows (columns). Clearly, stretching a planar layout increases its area; however, if it is stretched in appropriate locations it can be wired in fewer than four layers. It is well known that every planar layout W can be stretched and two-layer wired in an area not larger than twice the area of W. The bound on the wiring area is best possible, i.e. there are planar layouts that need to be stretched by this factor of two. Every planar layout W can be stretched and three-layer wired in an area not larger than $(4/3)$ times the area of W. The bound on the wiring area is best possible for wirings obtained by partitioning diagonal diagrams, i.e. there are planar layouts that need to be stretched by the factor $4/3$ when the wiring is obtained by partitioning diagonal diagrams. In this paper we investigate the problem of stretching and then wiring in three layers a planar layout. We show that for a class of planar layouts, the additional area is no more than $1/4$ of the planar layout area. We develop an algorithm to construct an optimal area layout for the case when each of the sublayouts must be wirable by one of a given set of algorithms.

Keywords. Three-layer wirability, planar layout, knock-knee model, stretching.

1. Introduction

The *rectangle routing problem* (RRP) is a fundamental problem in VLSI design automation. As input we are given a rectangular *grid* R determined by the

* This research was supported in part by the National Science Foundation under Grant DCR 8503163. A preliminary version of this paper appears as UCSB Technical Report TRCS 86-28 and in the Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD '87), October 1987, pp. 278–281.

Elsevier Science Publishers B.V.
INTEGRATION, the VLSI journal 8 (1989) 111–141

horizontal lines with y-coordinate values i, $0 \leqslant i \leqslant m+1$ (called *tracks* or *rows*) and the vertical lines with x-coordinate values j, $0 \leqslant j \leqslant n+1$ (called *columns*). The horizontal lines with y-coordinate values 0 and $m+1$ and the vertical lines with x-coordinate values 0 and $n+1$ form the *boundary* of R. Let $N = \{N_1, N_2, \ldots, N_p\}$, where each N_i is a subset of grid points on the boundary of R (excluding the corners of R), such that $N_i \cap N_j = \emptyset$ for all $i \neq j$. Each set N_i is called a *net* and its grid points are called *terminals*. We assume that there are k conducting layers $L_1, L_2, \ldots, L_k$, each is a copy of the channel grid, and L_{i+1} is considered to be laid upon L_i, $1 \leqslant i \leqslant k-1$. Contacts between two layers (*vias*) can be introduced only at grid points. Under the *knock-knee* model a k-layer wiring (which is the final routing solution) is a three-dimensional structure which can be characterized by two mappings: wire layout and layer assignment. A *wire layout* for a RRP is a mapping that associates each net N_i to a subgraph W_i of the grid R connecting all terminals in N_i such that W_i does not share an edge with W_j for all $j \neq i$. This wire layout is called a *planar layout*, a *path disjoint layout*, or simply a *layout*. We use $W = \{W_1, W_2, \ldots, W_p\}$ to denote the wire layout. The *layer assignment* of a planar layout is a mapping that associates each edge in W to a layer in $\{L_1, L_2, \ldots, L_k\}$ in such a way that for any $i \neq j$, if edges (p_1, p_2) and (p_2, p_3) in W_i are assigned to L_s and L_t, respectively, and $(p_2, p_4) \in W_j$ is assigned to L_u, then $u > \max\{s, t\}$ or $u < \min\{s, t\}$. A solution for an RRP is a *k-layer wiring* formed by the composite mapping of wire layout and layer assignment. Obviously, in a wiring the segments of the same wire W_i can be connected through a via without sharing a grid point with a segment of another wire W_j in any layer. Physically speaking, in a wiring all terminals from the same net are made electrically common and no two distinct nets are connected. In

Teofilo F. Gonzalez received the B.S. degree in computer science from the Instituto Technologico de Monterrey and the Ph.D. degree in computer science from the University of Minnesota. Minneapolis, in 1972 and 1975, respectively.

He has been with the University of Oklahoma, the Pennsylvania State University, the Instituto Technologico de Monterrey and the University of Texas at Dallas. He is now Professor of Computer Science at the University of California, Santa Barbara. His major research interests are in the design and analysis of algorithms, computational aspects of computer-aided design, and scheduling theory.

Dr. Gonzalez is a member of the Association for Computing Machinery, and the Operations Research Society of America.

Si-Qing Zheng is an assistant professor in the Department of Computer Science at the Louisiana State University. His research interests include VLSI, computational geometry, combinatorial algorithms and parallel computing. Dr. Zheng received the B.S. degree from Jilin University, China, in 1973, the M.S. degree in computer science from the University of Texas at Dallas in 1982, and the Ph.D. in computer science from the University of California, Santa Barbara, in 1987.

practice knock-knee wirings minimize crosstalk between wires since the area of the grid shared by two different nets is limited to grid points. A variation of the rectangle routing problem is the *channel routing problem* (CRP) in which all terminals appear on the top and bottom boundaries of the grid R and the objective is finding a k-layer wiring with the least number of tracks.

The above characterization of the wiring provides an approach for constructing a wiring for a CRP by finding a planar layout with least number of tracks for the given CRP and then finding a layer assignment for the layout. This approach is used by the routing algorithm for the two-terminal net CRP (each net has exactly two terminals) by Preparata and Lipski [9]. The first phase of their algorithm finds a minimum-track planar layout that satisfies some special properties. In the second phase of their algorithm, the planar layout is transformed into a three-layer wiring by a powerful transformation (legal partition of the diagonal diagram induced by the planar layout). Their algorithm guarantees a three-layer optimal wiring. Several other routing algorithms for the CRP are also based on this approach (e.g. see the algorithms in [8,10,12]).

Brady and Brown [1] showed that every planar layout can be transformed into a four-layer wiring with dimensions identical to those in the planar layout. The implication of this result is that one can reduce the rectangle routing problem to the problem of finding a planar layout, since for any planar layout a four-layer wiring is always possible in the layer assignment phase. One may consider this two-phase four-layer routing approach as 'standard'. For example, necessary and sufficient conditions for the existence of a planar layout for the two-terminal net RRP are given in [3]. If these conditions are met, a planar layout can be found by the algorithms in [3] and [7]. A four-layer wiring for this planar layout can be found by applying the layer assignment algorithm given in [1]. It is not known whether the layouts generated by the algorithms in [3] and [7] are three-layer wirable.

By using the reduction given in Lemma 2.1 (refer to the next section) Lipski [6] gives a 19-row wire layout that is not three-layer wirable. He also shows that the problem of deciding whether a given planar layout is three-layer wirable is an NP-complete problem. However, as we show in Section 7, the problem of determining whether a k-row planar layout is three-layer wirable is polynomially solvable when k is bounded by some fixed constant.

A planar layout may be *stretched vertically* (*horizontally*) by introducing between a pair of adjacent rows (columns) an empty row (column) (see Figs. 3.1 and 3.5) without a horizontal (vertical) wire segment on it. Clearly, stretching a planar layout increases its area; however, if it is stretched in certain places it can be wired in less than four layers. Let us now investigate the trade-off between the routing area and the number of layers needed for wiring a planar layout. Let $A(W)$ denote the area of planar layout W. The simple stretching algorithm described in [7] generates a two-layer wiring with area not larger than $2 * A(W)$, by vertically stretching it between every pair of adjacent rows. An algorithm that constructs a minimum area wiring by vertically stretching the planar layout is given in [2]. In [5] it is shown that there exist planar layouts with area $A(W)$

which cannot be stretched and wired in an area less than $(2-\epsilon) * A(W)$, for all $\epsilon > 0$. The stretching algorithm described in [2] and [4] generates a three-layer wiring with area not larger than $(4/3) * A(W)$, by vertically stretching it between every three adjacent rows. In [5] it is shown that there exist planar layouts with area $A(W)$ which cannot be stretched and wired by partitioning diagonal diagrams in an area less than $(4/3-\epsilon) * A(W)$, for all $\epsilon > 0$. Therefore, the area bound for arbitrary planar layouts is tight for diagonal diagram methods when one is only allowed to vertically or horizontally stretch the planar layout. Gonzalez and Zheng [5] also present lower bounds for the area of the three-layer wiring generated by an algorithm (the algorithm is not required to construct the wiring by partitioning the corresponding diagonal diagram) that is only allowed to vertically or horizontally stretch a planar layout. In this paper we show that a class of planar layouts can be stretched and three-layer wired in an area not larger than $(5/4) * A(W)$. Our technique consists of vertically stretching a planar layout every four rows. We show that each of these four-row planar layouts is three-layer wirable. This area bound is close to its lower bound of $7/6$ given in [5] for algorithms that construct wirings by partitioning the corresponding diagonal diagram. We also develop an algorithm to construct an optimal area layout for the case when each of the sublayouts must be wired by one of a given set of algorithms. Other issues relating to three layer wirability of planar layouts are also addressed in this paper.

In Section 2 we present some preliminary definitions and results developed by Preparata and Lipski [9] for three-layer wiring planar layouts. Our stretching-wiring algorithm is given in Section 3. For completeness, in Section 4 we present an algorithm to wire in three layers any two-row planar layout and in Section 5 we present an algorithm to wire in three layers any three-row planar layout. The algorithm presented in Section 5 is obviously more general than the one in Section 4; however, the algorithm in Section 4 tends to generate wirings with fewer vias. In some applications these wirings are more desirable. The algorithm in Section 5 is different from the one in [2]. In Section 6 we present an algorithm that generates three-layer wirings for a special class of four-row planar layouts. In Section 7 we present an algorithm to find a three-layer wiring for any k-row planar layout, whenever such a wiring exists. The time complexity for this algorithm is linear; however, the constant associated with this bound is exponential on k. In Section 8 we discuss the implications of our results.

2. Preliminaries

In this section we review some definitions and results from [9]. Preparata and Lipski's algorithm for finding a wiring for a given planar layout W consists of the following steps:

(i) construct the core diagonal diagram D that corresponds to the layout W;
(ii) find a legal partition P of D that partitions D into two-colorable regions; and

(iii) find a layer assignment W' of W from P.

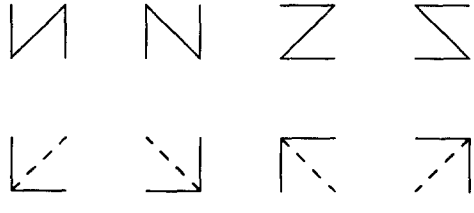
The diagonal diagram corresponding to the given layout is constructed as follows. At each grid point in R where W has a bend, a $(\sqrt{2}/2)$-length diagonal (called *half diagonal*) emanating from the grid point and internally bisecting the bend wire is introduced. Thus, in case there is a knock-knee at a grid point, a $\sqrt{2}$-length diagonal (called *full diagonal*), centered at the grid point, is formed. The resulting geometric structure from this transformation is called a *diagonal diagram*. The *core* diagonal diagram of a given layout is the diagonal diagram with the half-diagonals deleted (see Fig. 2.2). For the grid R, the partition grid $G(R)$ is defined as follows. The *grid points* of $G(R)$ are the points $(x + 1/2, y + 1/2)$, $0 \leqslant x \leqslant n$, $0 \leqslant y \leqslant m$. The grid points with $x = 0$, $x = n$, $y = 0$ or $y = m$ are called *boundary* points and the other grid points are called *internal* points. A vertical (horizontal) *grid line* in $G(R)$ is the line segment that includes all the grid points with the same y-coordinate (x-coordinate). The space between any two adjacent horizontal (vertical) grid lines is called a *row* (*column*) of $G(R)$. Note that a row (column) in R is a horizontal (vertical) grid line in R, but a row (column) in $G(R)$ is all the space between two adjacent horizontal (vertical) grid line in $G(R)$. The *edges* of $G(R)$ are the segments connecting each point with its immediate neighbors, vertically, horizontally, or at 45-degree angles. Let D denote the core diagonal diagram of layout W. It should be noted that a full diagonal in D lies on an edge of $G(R)$, the end points of a full diagonal are grid points of $G(R)$ and no two full diagonals in D cross. Each nonboundary grid point of $G(R)$ is called a *vertex*. We say that the vertex located at grid points (s, t) is D has degree i, $0 \leqslant i \leqslant 4$, if there are exactly i full diagonals with end points at (s, t). We say that D is of *degree* i, $0 \leqslant i \leqslant 4$, if the maximum degree of any vertex in D is i. A *legal partition* P of D (see Fig. 2.2) is any collection of edges in $G(R)$ satisfying the following conditions:

(a) the number of edges in P incident to any vertex of $G(R)$ is even;

(b) the diagonals in P are exactly those in D; and

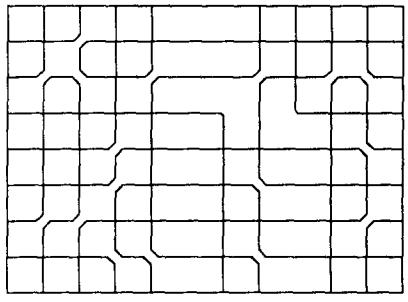
(c) P does not contain any of the patterns shown in Fig. 2.1.

The following lemma shows the significance of the notion of diagonal diagram D constructed from W and the legal partition P with respect to D.

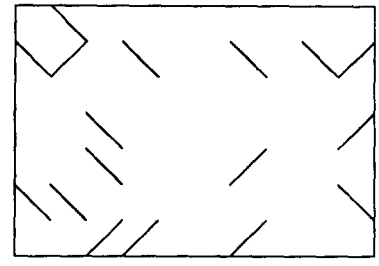


dashed diagonals must not be present

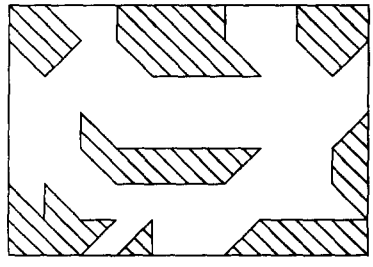
Fig. 2.1. Forbidden patterns.



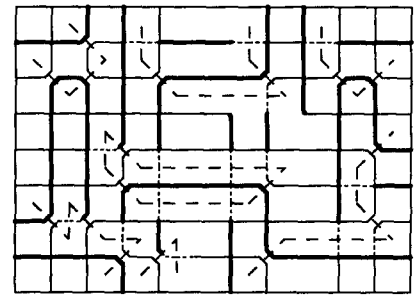
Planar Layout



Core Diagonal Diagram



Legal Partition



Three-Layer Wiring

— top layer (solid lines)
····· middle layer (dotted lines)
— bottom layer (dashed lines)

Fig. 2.2. Examples.

Lemma 2.1. [9]. *If there is a legal partition P of the core diagonal diagram D induced by the planar layout W, then there exists a three-layer assignment for W. Furthermore, a three-layer assignment for W can be easily constructed from P.*

We omit the description of the process for finding a layer assignment of W from a legal partition P of D. Interested readers should refer to [9] for details. In Fig. 2.2. we give a layout W, its corresponding core diagonal diagram D, the legal partition P of D and the three-layer wiring for W obtained from P. In the remaining portion of this paper we use the term diagonal diagram to refer to a core diagonal diagram.

3. Finding stretching positions

As mentioned in Section 1, it is always possible to find a three-layer wiring for a given planar layout by stretching it. Stretching a planar layout vertically is equivalent to inserting into the layout grid lines without any horizontal wires on it. Let us now briefly describe our approach for three-layer wiring. First we divide

a planar layout horizontally into three-layer wirable sublayouts. Then we find a three-layer wiring for each of these sublayouts and insert an empty grid line between every pair of adjacent sublayouts. In the final step, the vertical wires in every two adjacent sublayouts are joined at the newly introduced grid line between them by vias whenever necessary. Since the problem of determining whether or not a planar layout is three-layer wirable is NP-complete [6], it is simple to show that the problem of finding a minimum area three-layer wiring of a given layout W by vertically or horizontally stretching it is NP-hard. Our problem consists of dividing vertically a planar layout into a small number of sublayouts such that each of these sublayouts is wirable by one of a given set of algorithms.

The algorithms in [4] and [2] find a three-layer wiring for any three-row planar layout. Therefore any planar layout W that is vertically stretched every three rows can be wired in three layers. The resulting wiring has area not larger than $(4/3) * A(W)$. For completeness, we present our algorithm 3ROW-ASSIGN in Section 5. Using a similar approach, but with algorithm 2ROW-ASSIGN (see Section 4), the area bound becomes $(3/2) * A(W)$. Even though the area bound is larger, these wirings tend to have a smaller number of vias. In Section 6 we present algorithm 4ROW-ASSIGN that constructs a three-layer wiring for any four-row planar layout whose diagonal diagram is of degree 1. If the diagonal diagram of a given planar layout W is of degree 1, a three-layer wiring with area not larger than $(5/4) * A(W)$ can be obtained by following our general strategy.

Given a planar layout W whose diagonal diagram is of degree greater than 1, can we find a three-layer wiring with area smaller than $(4/3) * A(W)$? Consider the planar layout given in Fig. 3.1 whose diagonal diagram is shown in Fig. 3.2. This diagonal diagram is of degree 4. A legal partition is shown in Fig. 3.3. Finding this legal partition is difficult. If we divide the planar layout into sublayouts with no more than three rows each, then the minimum number of sublayouts is 7. On the other hand, if we divide it into sublayouts with diagonal diagram of degree 1, the minimum number of such sublayouts is also 7. However, if we divide it into the five subdiagrams delineated by the dashed lines in Fig. 3.2, we can find a legal partition by applying either procedure 3ROW-ASSIGN or 4ROW-ASSIGN on each subdiagram. The partitions constructed by these procedures are shown in Fig. 3.4. The final wiring is given in Fig. 3.5. In this case $A(W') \leqslant (23/19) * A(W)$. This example shows that when we divide a planar layout into three-layer wirable sublayouts we should be more interested in the local structure of adjacent rows in the layout instead of only considering the structure of the entire layout.

To obtain good wirings, we need to consider the case when a sublayout W has a trivial layout structure surrounding it. In this case the diagonal diagram of the planar layout consists of three regions (from top to bottom): a region whose rows are empty, a region of adjacent rows which can be legally partitioned and another region of empty rows. When one constructs a legal partition for the middle region, it can be constructed in such a way that there are no horizontal partitioning lines on the top and bottom boundary of this region. Now, for each

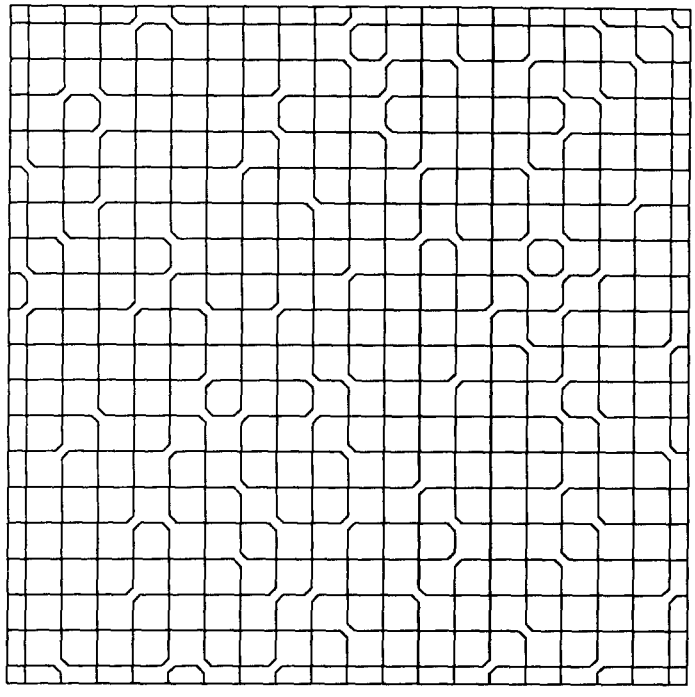


Fig. 3.1. Planar layout.

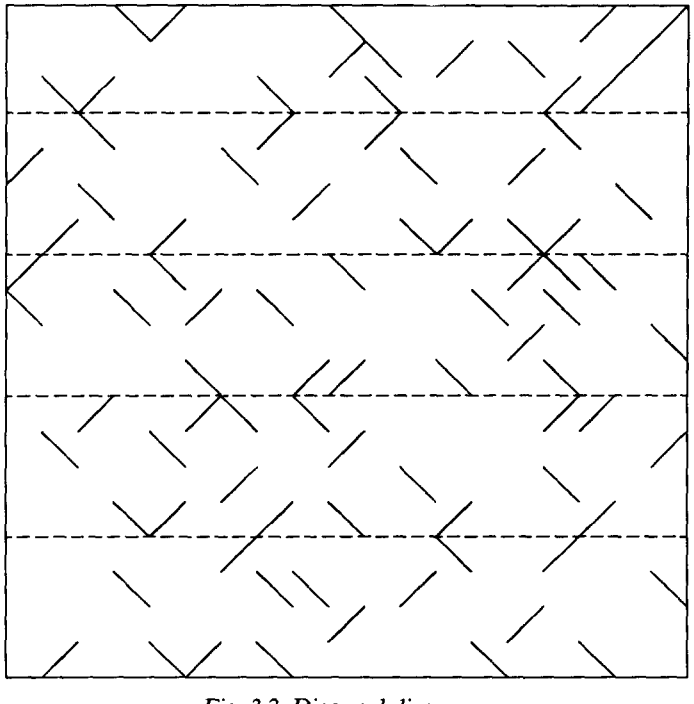


Fig. 3.2. Diagonal diagram.

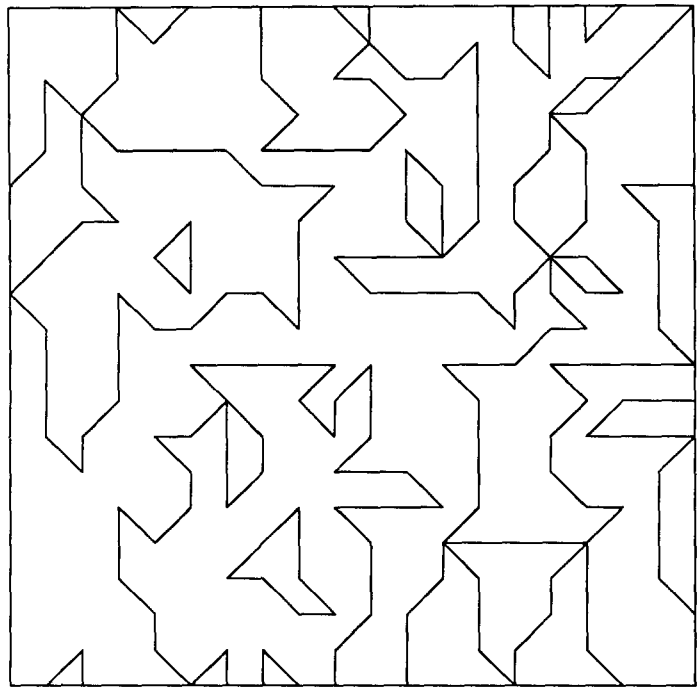


Fig. 3.3. Legal partition.

vertex on the top (bottom) boundary of the middle region incident with an odd number of partitioning line segments and diagonals, we add a vertical partitioning line segment emanating from it and finishing on the top (bottom) boundary of the top (bottom) region. It is easy to prove that the above procedure generates a legal partition of the diagonal diagram for the three regions from the legal partition of the diagonal diagram for the middle region.

A layer assignment algorithm that constructs a wiring for a planar layout by partitioning the corresponding diagonal diagram is said to be *consistent* if the following two conditions are satisfied:

(a) if the algorithm generates a three-layer wiring for a planar layout W, then the algorithm must also generate a three-layer wiring for a planar layout W', where the diagonal diagram for W' is the diagonal diagram for W surrounded by a trivial structure; and

(b) if the algorithm generates a three-layer wiring for a planar layout W, then the algorithm must also generate a three-layer wiring for W', where planar layout W' consists of contiguous rows in W.

From the discussion in the previous paragraph it is simple to see that any layer assignment algorithm which constructs a wiring for a planar layout by partitioning the corresponding diagonal diagram can be modified to satisfy condition (a). Modifying any layer assignment algorithm so that it satisfies property (b) is in general very difficult; however, in many cases it is simple. An example of a consistent layer assignment algorithm is a generalized version of algorithm

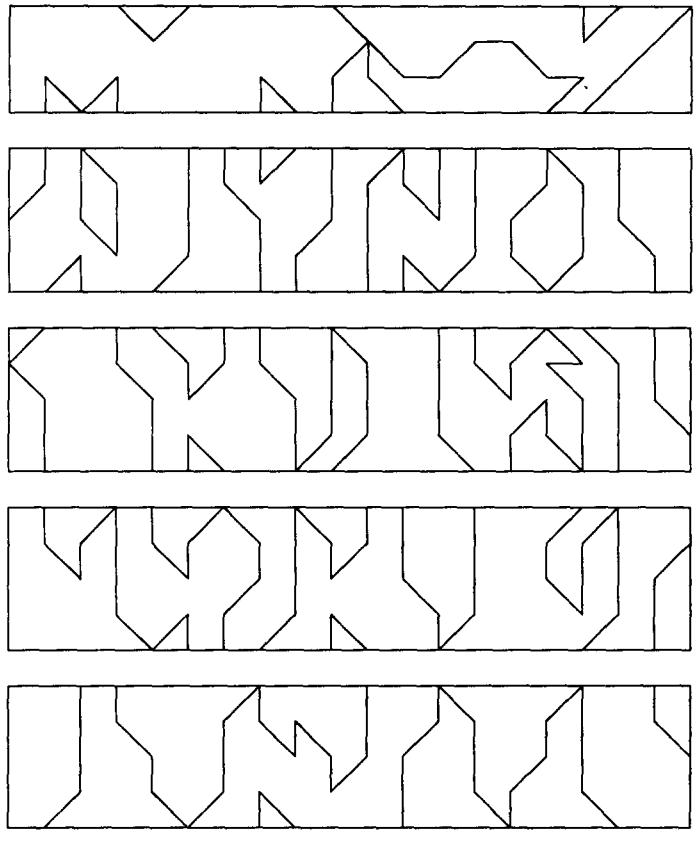


Fig. 3.4. Augmented legal partition.

3ROW-ASSIGN (see Section 5) which first checks if the diagonal diagram consists of three or less rows surrounded by a trivial structure. If so, it constructs a three-layer wiring for it by incorporating the techniques for handling trivial structures mentioned before; otherwise it does not generate a wiring. Note that the algorithm 2ROW-ASSIGN given in the next section does not construct a wiring from a legal partition of the diagonal diagram. Suppose we have a set of consistent layer assignment algorithms $A_1, A_2, \ldots, A_r$ that find three-layer wirings from a legal partition of a diagonal diagram. We say that $D^1, D^2, \ldots, D^t$ is a *vertical division* of a diagonal diagram D with respect to $A_1, A_2, \ldots, A_r$ if the concatenation of the D^js (D^j above D^{j+1}) is D, and D^j, $1 \leqslant j \leqslant t$, is wirable by one of the algorithms A_i. A vertical division $D^1, D^2, \ldots, D^t$ of D with respect to $A_1, A_2, \ldots, A_r$ is *optimal* if t is least possible among all vertical divisions of D. Note that optimality can also be defined with respect to horizontal divisions. Optimal divisions for this more general problem can be obtained by following a similar procedure. For simplicity, we did not define it this way. One can also define a more general stretching scheme by using algorithms that construct wirings of planar layouts rather than restricting the algorithms to constructing the wirings by partitioning diagonal diagrams. Since most of the known wiring

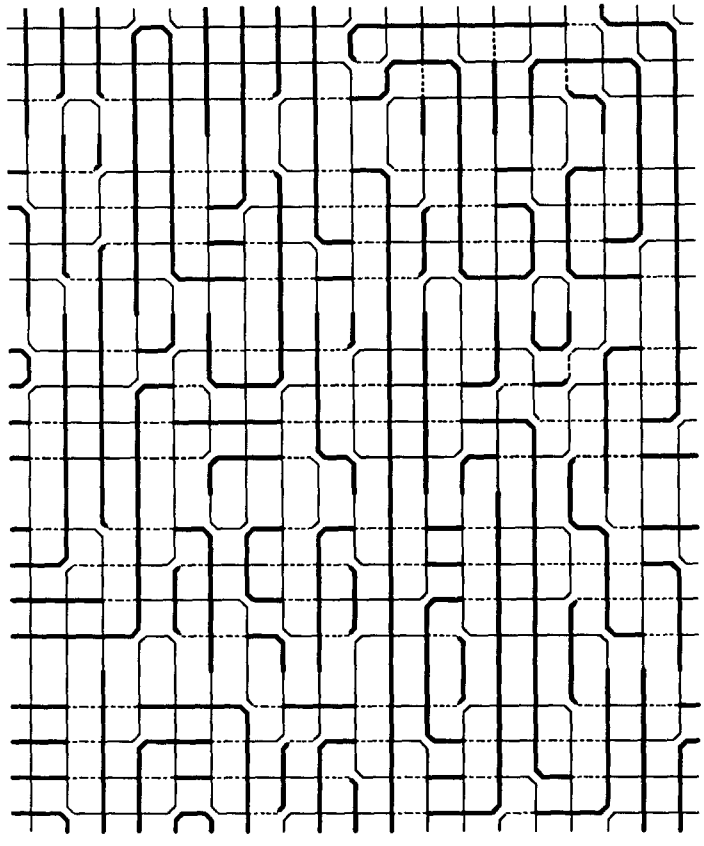


Fig. 3.5. Augmented three-layer wiring.

algorithms construct a wiring by partition diagonal diagrams, we decided on our definition.

Assuming that the bottom and top boundaries of $G(R)$ have a y-coordinate value 0 and m, respectively, procedure DIV is defined as follows.

procedure DIV
begin
  $t \leftarrow 1$;
  $j \leftarrow 0$;
  while $j < m$ **do**
    for $i = 1$ **to** r **do**
      Let k_i be the largest integer such that the diagonal diagram between the grid lines with y-coordinate values j and k_i is wirable by algorithm A_i;
    endfor
    $y_t \leftarrow$ **max**$\{k_i \mid 1 \leqslant i \leqslant r\}$;
    $j \leftarrow y_t$;
    $t \leftarrow t + 1$;
  endwhile
end of procedure DIV

Procedure DIV finds $t-1$ lines (with y-coordinates $y_1, y_2, \ldots, y_{t-1}$) that divide the diagonal diagram D into t subdiagrams. For each of these subdiagrams a legal partition can be found by one the algorithms $A_1, A_2, \ldots, A_r$. Since the problem of determining whether or not a planar layout is three-layer wirable is NP-complete [6], it is simple to show that the problem of finding a minimum area three-layer wiring of a given layout W by stretching it is NP-hard. However, if one restricts the above problem so that each of the subdiagrams must be legally partitioned by one of the algorithms $A_1, A_2, \ldots, A_r$, we have the following result.

Theorem 3.1. *With respect to the consistent layer assignment algorithms $A_1, A_2, \ldots, A_r$, procedure DIV finds a minimum number of division lines for diagonal diagram D.*

Proof. Suppose the diagonal diagram D can be divided into $t' < t$ subdiagrams each wirable by one of the algorithms A_i. Let $y_i' < y_{i+1}'$, $1 \leqslant i < t'-1$, be the y-coordinate values of the $t'-1$ division lines. Let $y_0 = y_0'$ denote the y-coordinate value of the bottom boundary of R and $y_t = y_{t'}'$ denote the y-coordinate values of the top boundary of R. Since $t' < t$ there exists an integer i such that $y_i' < y_i$ and $y_{i+1}' > y_{i+1}$. Let i be the smallest integer that satisfies this condition. Since the algorithms are consistent (i.e. satisfy properties (a) and (b)), it must have been that procedure DIV would have selected the horizontal line with y-coordinate value y_{i+1}' instead of the one whose y-coordinate value is y_{i+1}. Therefore, there is a contradiction. Hence t is minimum with respect to algorithms A_i, $1 \leqslant i \leqslant r$. □

Our complete scheme for the layer assignment problem is given below.

procedure STRETCHING-WIRING

Construct on grid $G(R)$ the diagonal diagram D corresponding to the given layout W on R;

Use procedure DIV to optimally divide D on $G(R)$ into $D^1, D^2, \ldots, D^t$ (accordingly the planar layout W on R is partitioned into sublayouts $W^1, W^2, \ldots, W^t$);

For each D^i, $1 \leqslant i \leqslant t$, construct a legal partition P^i by one of the consistent algorithms $A_1, A_2, \ldots, A_r$;

From the legal partition P^i of D^i, construct a three-layer wiring A^i for sublayout W^i;

Construct grid R' by inserting an empty grid line in R between W^i and W^{i+1}, $1 \leqslant i < t$, connect A^i and A^{i+1} at the newly inserted grid line and introduce vias at the grid points on the new grid lines if necessary;

end of procedure STRETCHING-WIRING

4. A simple layer assignment algorithm for two-row planar layouts

A three-layer wiring for a two-row layout can be easily constructed from a legal partition of the diagonal diagram corresponding to the given layout. The

main disadvantage of this approach is that the wiring constructed from a legal partition of the diagonal diagram could require a large number of vias. The algorithm that we present in this section constructs a layer assignment without using the diagonal diagram and the wiring does not contain a via at any internal grid point.

Let W be a two-row layout defined over the grid R determined by horizontal lines with y-coordinate values i, $0 \leqslant i \leqslant 3$, and vertical lines with x-coordinate values j, $0 \leqslant j \leqslant n+1$. Note that there is no horizontal wire with y-coordinate value 0 or 3. For simplicity, we assume that each net consists of two terminals, and each wire in W is a jogging path in R (possibly with loops, i.e. the wire crosses itself or it forms a knock-knee with itself). It is important to note that our result for this simplified case can be easily generalized to cover arbitrary two-row layouts, including multiterminal net layouts. In order to simplify our proofs we transform W into a full layout, i.e. every edge in R has a wire. After constructing a full layout, there might be *useless wires* (loop wires without terminal points), and *useful wires* (wires with terminal points). A *trivial wire* is a useful wire consisting of a single vertical line segment, i.e a vertical wire connecting two terminal points located on the same column.

We denote the three different layers by T (top), M (middle) and B (bottom). Our procedure constructs a mapping L with domain the set of useful wires in W and range {T, M, B} such that for any two useful wires W_i and W_j, $i \neq j$, if they share a grid point then $L(W_i) \neq L(W_j)$. Clearly, this mapping defines a valid wiring. Note that if the two-row planar layout contains trivial wires, one can remove the columns occupied by these wires and solve the remaining problem. The solution to the original problem can be obtained by assigning each trivial wire to a layer that is different from any layer assigned to the two tracks that the trivial wire intersects. A problem with useless wires can be broken into a set of disjoint problems (without useless wires) which can be solved almost independently. The only dependency is that one of the layers that is not used in the rightmost column of the first problem (i.e. the layer that will be assigned to the useless wire) should not be used in the first column of the second subproblem. Since this dependency can be easily handled, we may restrict our attention to the problem of three-layer wiring of two-terminal net planar layouts without trivial and useless wires. Let $x(t)$ denote the x-coordinate of terminal t. For any wire W_i, let $B(W_i)$ and $E(W_i)$, where $x(B(W_i)) \leqslant x(E(W_i))$, denote the *beginning* and *ending* terminals of wire W_i, respectively. Also, let $left(W_i)$ ($right(W_i)$), denote the smallest (largest) x-coordinate of any point in wire W_i.

In what follows we show how to assign layers to each of the wires in a two-row planar layout. The grid R is determined by horizontal lines with y-coordinate values i, $0 \leqslant i \leqslant 3$, and vertical lines with x-coordinate values j, $0 \leqslant j \leqslant n+1$.

procedure 2ROW-ASSIGN(W)
Let $W = \{W_1, W_2, \ldots, W_p\}$;
Let LAYER be defined as a first-in-first-out queue;
Add T, M, and B (in any order) to LAYER;

```
for j = 0 to n + 1 do /* j is column number */
  case
    for each beginning terminal point in column j do
      /* In case there are two beginning terminal points the one whose
         wire has the smaller left(W) value is considered first. In case of
         ties the order is immaterial */
      Let W_i be the wire to which this terminal belongs;
      L(W_i) ← delete(LAYER);
    endfor
    for each ending terminal point in column j do
      /* In case there are two ending terminal points the one whose
         wire has the larger right(W) value is considered first. In case of
         ties the order is immaterial */
      Let W_i be the wire to which this terminal belongs;
      Add L(W_i) to LAYER;
    endfor
  endcase
endfor
end of procedure 2ROW-ASSIGN
```

Lemma 4.1. *Procedure 2ROW-ASSIGN generates a three-layer assignment for any two-terminal two-row full planar layout without useless and trivial wires in $O(n)$ time, where n is the number of columns.*

Proof. Since it is obvious that the procedure takes $O(n)$ time, we only prove that the algorithm generates a three-layer assignment for any two-terminal two-row full planar layout without useless and trivial wires. The proof is by induction on j, the column number. We show that at the end of the jth iteration, all the wires with their beginning terminal located before the $j+1$st column have been assigned a layer and the wiring is valid. This statement is obviously true for $j = 0$. Assume it holds for column j and consider column $j+1$. There are three cases depending on the type of terminal points encountered at column $j+1$.

Case 1. Both terminal points are beginning terminal points (from wires W_a and W_b). Assume without loss of generality that the $left(W_a) \leqslant left(W_b)$. Since both terminal points are beginning points it must be that the wire segments occupying the two tracks between column j and $j+1$ belong to the same wire W_c. If $c \neq a$ (e.g. Fig. 4.1(a)), then wire W_c has its ending point in column j. From our algorithm it is simple to verify that $L(W_c)$ was the last element added to LAYER and that LAYER contains three elements at the beginning of the jth iteration. Therefore, the algorithm assigns layers to W_a and W_b that are different from $L(W_c)$ and we have a valid wiring. Thus, the induction hypothesis holds at step $j+1$. On the other hand if $a = c$ (e.g. Figs. 4.1(b) and (c)), it must have been that column j contained two ending terminal points and LAYER had one element when processing column j. Clearly, the algorithm selects such layer for W_a and

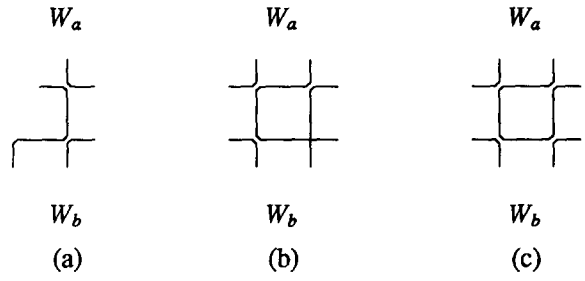


Fig. 4.1. Examples for case 1.

one of the other two layers for W_b. Therefore, we have a valid wiring and the induction hypothesis holds after the $j+1$st iteration. This completes the proof for this case.

Case 2. There is one beginning terminal point (from wire W_a) and one ending terminal point (from wire W_b). There are two subcases depending on whether $a = b$ or not.

Subcase 2.1: $a = b$. Clearly, the wire segments occupying the two tracks between column j and $j+1$ belong to the same wire W_c (e.g. Figs. 4.2(a) and (b)). If $c \neq a$ (e.g. Fig. 4.2(a)) then wire W_c has its ending point in column j. From our algorithm it is simple to verify that $L(W_c)$ was the last element added to LAYER and that LAYER contains three elements at the beginning of the $j+1$st iteration. Therefore, the algorithm assigns a layer to W_a that is different from $L(W_c)$ and we have a valid wiring. Thus, the induction hypothesis holds at step $j+1$. On the other hand, if $c = a$ (e.g. Fig. 4.2.(b)), it must be that column j contains two ending terminal points and LAYER had one element at the beginning of the jth iteration. Clearly, the algorithm selects such layer for W_a. Therefore, we have a valid wiring and the induction hypothesis holds after the $j+1$st iteration. This completes the proof for this subcase.

Subcase 2.2: $a \neq b$. For this case it is simple to verify that the two wire segments occupying the tracks between columns j and $j+1$st belong to different wires and that one of these wires is W_b (e.g. Fig. 4.3). The layer assigned to W_a is different from any of the layers assigned to these wires since

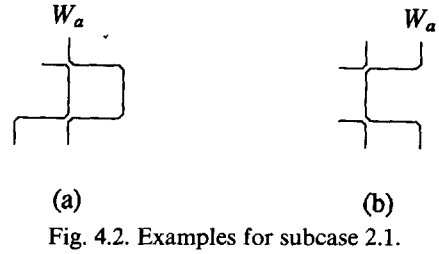


Fig. 4.2. Examples for subcase 2.1.

(a) (b)

Fig. 4.3. Examples for subcase 2.2.

the layer assigned to the two wires between columns j and $j+1$ is not added to LAYER until after an element is deleted from LAYER and assigned to W_a (note that the LAYER contains one element since there are only two active nets, i.e. nets for which we have processed the begin terminal but not its end terminal). Therefore we have a valid wiring and the induction hypothesis holds at step $j+1$st. This completes the proof of this subcase and case 2.

Case 3: Both terminal points are ending. The proof for this case follows from the fact that the algorithm only adds elements to LAYER and a valid wiring up to column $j+1$st already exists.

This completes the proof for this lemma. □

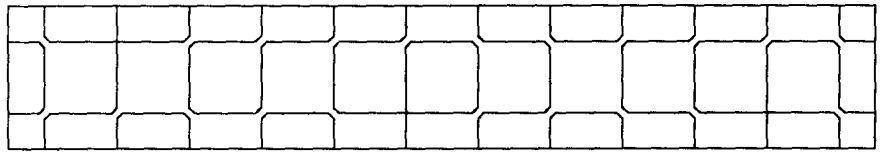
A two-row planar layout and its layer assignment constructed by the above algorithm are given in Fig. 4.4.

Theorem 4.1. *Every two-row planar layout can be three-layer wired without internal vias in $O(n)$ time, where n is the number of columns in R.*

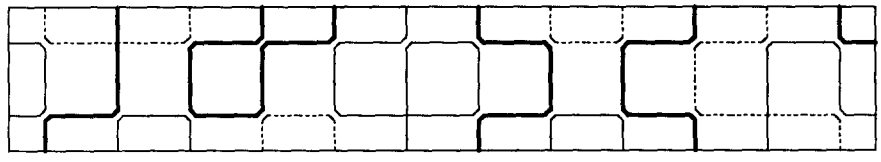
Proof. For the case of two-terminal net layouts, the wiring is obtained by following the rules mentioned just before procedure 2ROW-ASSIGN. The algorithm for multiterminal-net layouts is similar. Since the proof that the algorithms generate a three-layer wiring is straight forward, it will be omitted. □

5. A layer assignment algorithm for three-row planar layouts

Our layer assignment algorithm for a three-row planar layout W is based on finding a legal partition P of the diagonal diagram D. This legal partition is obtained from a legal connection C of the node diagram V induced by D. The *node diagram* V consists of the grid $G(R)$ and the nodes are defined from D as follows: there is a node in V at grid point (s, t) if the vertex at grid point (s, t) in D is of odd degree. A *legal connection* C of V is a set of horizontal and vertical grid line segments of $G(R)$ that satisfies the following conditions:



Planar Layout



Three-Layer Wiring

Fig. 4.4. Three-row planar layout.

(i) the end point of any line segment must either lie on the boundary of $G(R)$ or be incident to a node in V;
(ii) for every node v in V there is exactly one horizontal or one vertical line segment incident to v;
(iii) there are no two line segments l' and l'' in C lying on two adjacent vertical (horizontal) grid lines of $G(R)$ such that their projections to the y-axis (x-axis) share more than one point;
(iv) no two orthogonal line segments in C intersect.

From the definition of legal partition and legal connection, it is easy to prove the following lemma.

Lemma 5.1. *If C is a legal connection of node diagram V induced by the diagonal diagram D, then superimposing C on D yields a legal partition of D.* □

By this lemma, the problem of finding a legal partition P of D is reduced to the problem of finding a legal connection C of V induced by D. We graphically represent all the grid points in V without nodes by circles and represent all the nodes by filled-in circles. There are two internal horizontal grid lines in $G(R)$. If we partition the nodes in V into *columns* according to the vertical grid lines of $G(R)$ where they are located, we have four different types of columns. The legal connection C of V is constructed column by column in a left-to-right scan of V. When a column is being considered, we need to remember only the connection of the preceding column. Based on the configuration of this column and the type of the current column, the connection of the nodes in the current column is determined. The connection of the current column may modify the connection of the immediately preceding column; however, this modification guarantees that the resulting connection is legal. Without loss of generality we assume that the left and right boundary of $G(R)$ are columns with no nodes.

procedure 3ROW-ASSIGN
Construct the diagonal diagram D from W;
Construct the node diagram V corresponding to the diagonal diagram D;
Construct a legal connection C for V as follows:
for $c \leftarrow 1$ to n **do** /* left and right boundaries of $G(R)$ are columns 0 and $n+1$, respectively */
begin
Depending on the type of column c and the connection of column $c-1$ make a legal connection by using the actions given by the finite automata depicted in Fig. 5.1 (which is explained in the following paragraph) for the nodes in columns c and $c-1$. Note that this step may modify the connection of column $c-1$ obtained in the previous iteration.
end
Find the legal partition P of D from the legal connection C of V and construct a three-layer wiring of W from P;
end of procedure 3ROW-ASSIGN

The actions in step (3) of the algorithm can be visualized by means of a finite state automata. The transition diagram for this automata is illustrated in Fig. 5.1. Each state corresponds to the connection in the previous column. The state composed of two x's means that those grid points can either have a node or not, and the dotted lines mean that those lines are absent. Associated with each transition in the automaton we define an input–output pair. The first component of the pair, the input, specifies the type of the column being considered and the second component, the output, shows the final configuration for both the previous and the current columns. By starting in the state corresponding to the

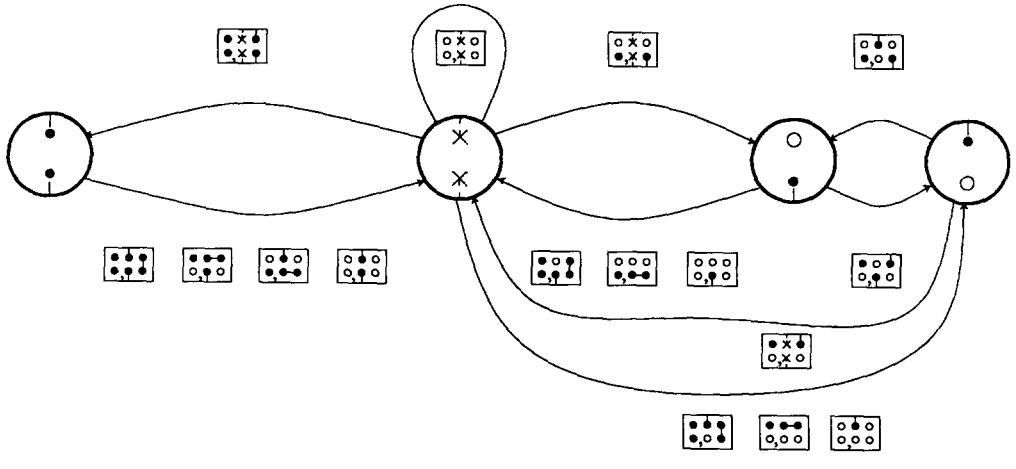


Fig. 5.1. Fine automation describing the actions for 3ROW-ASSIGN.

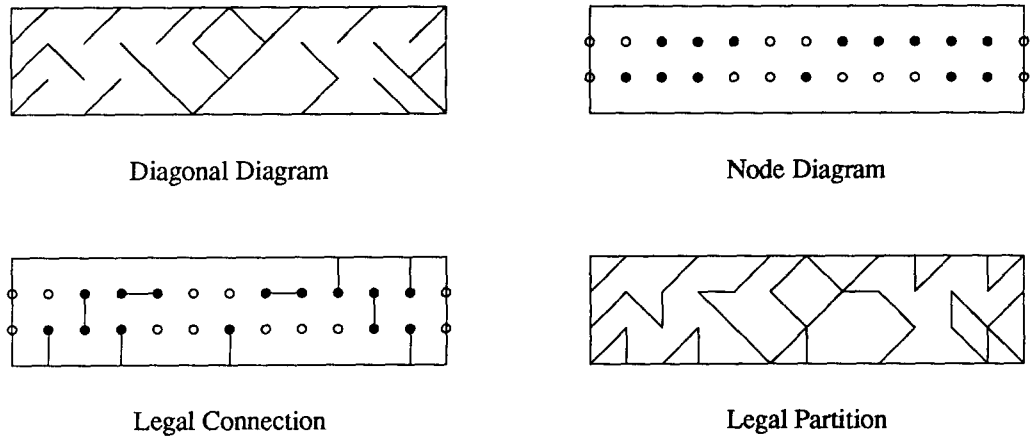


Fig. 5.2. Construction of legal partition from diagonal diagram.

left boundary of $G(R)$, it is easy to prove by induction that a legal connection C of V is constructed. Therefore, we have the following theorem.

Theorem 5.1. *Algorithm 3ROW-ASSIGN generates a legal connection for any three-row planar layout in $O(n)$ time, where n is the number of columns in R.*

Proof. The proof for the time complexity bound is straightforward. The correctness of our algorithm can be established by induction on the length of the path followed when traversing the finite automaton. □

An example of a diagonal diagram D with node diagram V is given in Fig. 5.2. A legal connection C obtained by our algorithm and the legal partition P of D by superimposing C on D is also illustrated in Fig. 5.2. Gonzalez and Zheng [5] show that there exists a four-row planar layout whose diagonal diagram cannot be legally partitioned. Hence, it is impossible to construct a four-row layer assignment algorithm for arbitrary planar layouts by partitioning diagonal diagrams.

6. A layer assignment algorithm for a class of four-row planar layouts

In this section we present an algorithm to construct a legal partition for any four-row diagonal diagram of degree 1. There are three internal grid lines in $G(R)$. Assign the y-coordinate one to the upper internal grid line, three to the middle internal grid line, and two to the lower internal grid line (this ordering facilitates the proof of correctness). Assign the x-coordinate zero to the left boundary of $G(R)$, one to the vertical grid line (column) immediately to the right of the left boundary of $G(R)$, and so on. Each vertex in the diagonal diagram of degree 1 will also be referred to as a *node* and it will be represented by its

location on the grid. Sort the nodes of D into an ordered list $v_1 = (x_1, y_1)$, $v_2 = (x_2, y_2), \ldots, v_q = (x_q, y_q)$ such that $x_i \leqslant x_{i+1}$ and if $x_i = x_{i+1}$, $y_i > y_{i+1}$. The nodes will be referred as *top*, *middle* or *bottom*, depending on which internal grid line in $G(R)$ they are located. Since the diagonal diagrams are of degree 1, in a legal partition every node has exactly one vertical or horizontal partitioning line incident to it. When a node has a vertical or horizontal partitioning line incident to it, we say that the node has been *connected*, otherwise we say that the node is *not connected*. After adding a set of partitioning line segments to a diagonal diagram we say that it forms a *partial legal partition* if there are no forbidden patterns (see Fig. 2.1) and no node has more than one partitioning line incident to it. Note that in a partial partition some nodes might not be connected. Unlike procedure 3ROW-ASSIGN, connections cannot be made by only concentrating on the structure of the node diagram and totally ignoring the diagonals. When the number of rows in the layout is greater than three, it seems unlikely that one can construct a legal partition in a left-to-right column by column fashion.

Our algorithm consists of three procedures; VERTICAL, MODIFY and HORIZONTAL. These procedures are invoked as follows. At each step we have a partial partition and we find the smallest positive integer c such that node v_c is not connected. Then our procedure construct a partial partition in which nodes $v_1, v_2, \ldots, v_c$ and possibly other nodes are connected. The first step invokes procedure VERTICAL. If procedure VERTICAL fails to connect node v_c, procedure MODIFY will introduce a set of partitioning line segments and all the nodes in column c will be connected. This operation is performed by modifying some previously introduced partitioning line segments. However, all the nodes that were connected before will remain connected. Whenever procedure MODIFY introduces a horizontal line segment to the right of the current column, x_c, procedure HORIZONTAL completes the connections of the nodes above and below this horizontal line segment. At this point we find the least value of c such that node v_c is not connected. This process continues until all nodes are connected by exactly one partitioning line and the entire configuration is a legal partition. In what follows we explain these procedures in more detail.

When a node v_c is considered (by procedure VERTICAL) we first check to see if it is possible to add a vertical line segment emanating from v_c and ending at the nearest neighbor node above v_c, or at the top boundary of $G(R)$ if such neighbor node does not exist. If the addition of this line segment does not form a forbidden pattern (see Fig. 2.1) and no node has more than one partitioning line segment incident to it, we add it and node v_c is connected. Otherwise we check to see if it is possible to add a vertical line segment emanating from v_c and ending at the nearest neighbor node below v_c or at the bottom boundary of $G(R)$ if such a node does not exist. If the addition of this line segment does not form a forbidden pattern (see Fig. 2.1) and no node has more than one partitioning line segment incident to it, we add it. There are situations when this step also fails to connect v_c. In Lemmas 6.1–6.3 we characterize the situations when both of these steps fail to connect node v_c. In these lemmas we show that the connection of a node fails only when obstructive configurations 1–18 surround node v_c (see Fig.

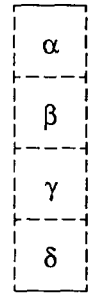


Fig. 6.1. Regions.

6.3). We say that obstructive configuration *i surrounds node* v_c *in a partial partition* P if node v_c is located on the right boundary of the obstructive configuration, and all the diagonals and partitioning line segments in the obstructive configuration are present in P. In Fig. 6.3 we use a small circle to indicate the absence of a node, and a filled-in circle for a node. Note that there might be some other diagonals or partitioning line segments in P that are not explicitly shown in the obstructive configuration. Before proving Lemmas 6.1–6.3, we make a couple of definitions. In what follows we shall refer to the the unit squares between columns x_{c-1} and x_c shown in Fig. 6.1 as *regions* α, β, γ, and δ. Also, when the patterns shown in Fig. 6.2 are in any of these regions, we shall refer to them as a type A or type B region (solid line segments indicate diagonals and partitioning lines that must be present). In the following lemmas we show that while procedure VERTICAL is processing the nodes in column x_c, a connection is always possible unless obstructive configurations 1–18 surround v_c. At this point it is important to remember that the nodes in a column are consider in the order top, bottom and then middle. This ordering facilitates our proof of correctness. In what follows when we refer to the top, middle and bottom nodes, we mean the top, middle and bottom node in column x_c.

Lemma 6.1. *Given a partial legal partition and the smallest positive integer c such that node* v_c *is not connected by a partitioning line segment and* v_c *is a top node, procedure* VERTICAL *connects* v_c *unless obstructive configurations* 1–4 *surround* v_c.

Proof. Suppose that procedure VERTICAL does not connect the top node (v_c). Since the top node (v_c) is not connected to the top boundary, it must be that

Fig. 6.2.Types A and B.

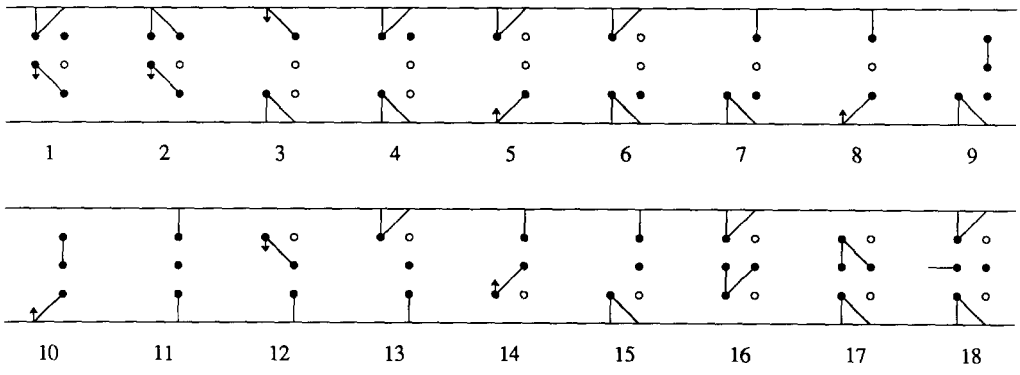


Fig. 6.3. Obstructive configurations.

region α is of type A or B. If there is a middle node, then since the top node v_c cannot be connected to it, we know that region β is either of type A or type B. But this implies that either two diagonals will be incident to the same node (in which case it is not a diagonal diagram of degree 1) or there is a point of degree greater than two on the left boundary of the region (in which case it contradicts the assumption that we start with a partial partition). Therefore, it must be that there is no middle node. There are two cases depending on whether there is bottom node or not.

Case 1: There is a bottom node. Region γ cannot be of type B because there is no middle node. Therefore, the top node cannot be connected to the bottom node only if γ is of type A. These are obstructive configurations 1 and 2.

Case 2: There is no bottom node. Region γ cannot be of type A or B since there is no middle nor bottom node. Similarly since there is no bottom node, region δ cannot be of type B. Therefore, region δ must be of type A. These cases correspond to obstructive configurations 3 and 4.

Therefore, procedure VERTICAL connects the top node, unless obstructive configurations 1-4 surround v_c. This completes the proof of Lemma 6.1. □

Lemma 6.2. *Given a partial legal partition and the smallest positive integer c such that node v_c is not connected by a partitioning line segment and node v_c is a bottom node, procedure VERTICAL connects node v_c unless obstructive configurations* 5–10 *surround v_c.*

Proof. Suppose the bottom node is not connected by procedure VERTICAL. There are two cases depending on whether there is a top node or not.

Case 1: There is no top node. Arguments similar to the ones used in Lemma 6.1 can be used to show that region δ is of type A or B, there is no middle node, and region γ cannot be type A or B. Region β cannot be of type A or B since there is

no middle nor top node. Similarly, since there is no top node, region α cannot be of type A. Therefore, region α must be of type B. These cases correspond to obstructive configurations 5 and 6.

Case 2: There is a top node. There are two cases depending on whether there is a middle node or not.

Subcase 2.1: There is no middle node. Clearly, the top node must be connected to the top boundary and region δ must be of type A or B. These are obstructive configurations 7 and 8.

Subcase 2.2: There is a middle node. There are two subcases depending on how the top node is connected.

Subcase 2.2.1: The top node is not connected to the middle node. Since the bottom node is not connected, it must be that each of the regions δ and γ are of type A or B. However, in each of these cases either two diagonals will be incident to the same node (in which case it is not a diagonal diagram of degree 1) or there is a point of degree greater than two on the left boundary of the region (in which case it contradicts the fact that we start with a partial partition).

Subcase 2.2.2: The top node is connected to the middle node. Clearly, region δ must be of type A or B. This corresponds to obstructive configurations 9 and 10.

Therefore, it is always possible to connect the top node, unless obstructive configurations 5–10 surround v_c. This completes the proof of Lemma 6.2. □

Lemma 6.3. *Given a partial legal partition and the smallest positive integer c such that node v_c is not connected by a partitioning line segment and v_c is a middle node, procedure* VERTICAL *connects v_c unless obstructive configurations* 11–18 *surround v_c.*

Proof. Suppose procedure VERTICAL does not connect the middle node. There are four cases.

Case 1: There is a top and bottom node. Clearly it must be that the top node is connected to the top boundary and the bottom node is connected to the bottom boundary. This is obstructive configuration 11.

Case 2: There is a bottom node, but there is no top node. Clearly, the bottom node must be connected to the bottom boundary. Since there is no top node, the middle node cannot be connected to the top boundary only when region α is of type B, or region β is of type A. These are obstructive configurations 12 and 13.

Case 3: There is a top node, but there is no bottom node. This is similar to case 3, but the obstructive configurations are 14–15.

Case 4: There is no top nor bottom node. Since there is no top nor bottom node and procedure VERTICAL fails to connect the middle node to the top or bottom boundary, it must be that region α is of type B or region β is of type A, and region γ is of type B or region δ is of type A. Since one of these four possibilities implies a diagonal diagram of degree greater than one, we are left with obstructive configurations 16–18.

This completes the proof of the lemma. □

It is simple to verify that when a connection is performed by procedure VERTICAL, we obtain a partial partition in which all nodes $v_1, v_2, \ldots, v_c$ are connected by either a vertical or horizontal line segment. In Fig. 6.4 we show how procedure MODIFY performs the connections when obstructive configurations 1–18 surround v_c. In this figure, we use a dotted line segment to indicate that such segment is not present, we use a small circle to indicate the absence of a node, and a filled-in circle for a node. The modifications are self-explanatory. Note that whenever we encounter a node v_c that procedure VERTICAL fails to connect, the connection for the entire column x_c is performed. In Lemma 6.4 we show that the connection performed by procedure MODIFY is legal, i.e. we obtain a partial partition in which each node $v_1, v_2, \ldots, v_c$ and every node in column x_c is connected. One should note that in certain cases the proposed modification introduces a horizontal line segment to the right of the current column. Later on we show how to deal with this case when processing the columns to the right of the current column.

Lemma 6.4. *The connections introduced by procedure* MODIFY *for obstructive configurations* 1–18 *are legal, i.e. we obtain a partial partition in which each node* $v_1, v_2, \ldots, v_c$ *and every node in column* x_c *are connected by a vertical or horizontal line segment.*

Proof. The proof of this lemma can be easily established by inspecting each of the modifications in Fig. 6.4 performed by procedure MODIFY. □

In case there is a horizontal line segment, only on the middle grid line of $G(R)$, that extends to the right of column x_c and ends at the nearest right neighbor node v_d of v_c, or the right boundary of $G(R)$, a call is made to procedure HORIZONTAL (see obstructive configurations 3, 5, 11, 13, 15 and 18 in Fig. 6.4). Let v_e be the middle node connected by this horizontal line.

In procedure HORIZONTAL, the nodes in the set CMPD $= \{v_i \mid x_c < x_i \leqslant x_d\}$ form two (possible empty) sublists of $v_1, \ldots, v_q$. These lists are $v_{i_1}, v_{i_2}, \ldots, v_{i_t}$, $(x_c < x_{i_1} < x_{i_2}, \ldots, < x_{i_t} \leqslant x_d,\ y_{i_k} = 1,\ 1 \leqslant k \leqslant g)$ and $v_{j_1}, v_{j_2}, \ldots, v_{j_k}$ $(x_c < x_{j_1} < x_{j_2}, \ldots, < x_{j_k} \leqslant x_d,\ y_{j_k} = 2,\ 1 \leqslant k \leqslant h)$. Any node $v = (x, y)$ whose x-coordinate is less than or equal to x_d has either already been connected or it belongs to one of these sublists. Without loss of generality we assume that $g,\ h \geqslant 1$. In what follows we informally describe how these two set of nodes are connected by

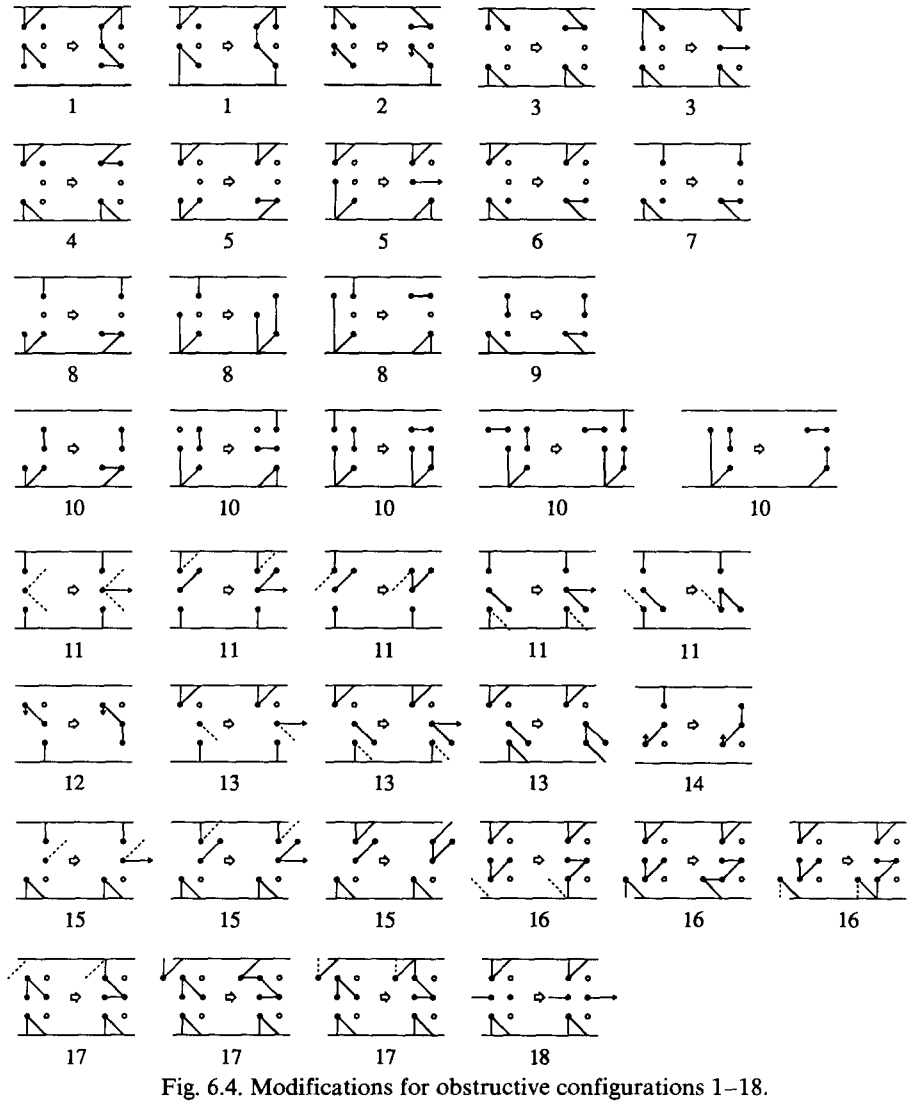


Fig. 6.4. Modifications for obstructive configurations 1–18.

procedure HORIZONTAL. First, nodes v_{i_1} and v_{j_k} are connected to the top and bottom boundary of $G(R)$ by a vertical line segment, respectively. If $g-1$ (or $h-1$) is a positive even integer, then we connect each pair of nodes $v_{i_{2k-1}}$ and $v_{i_{2k}}$, $1 \leqslant k \leqslant (g-1)/2$ (or $v_{i_{2k-1}}$ and $v_{i_{2k}}$, $1 \leqslant k \leqslant (h-1)/2$), by a horizontal line segment (see Fig. 6.5). Obviously such connections are legal. When $g-1$ $(h-1)$ is a positive odd number, we have three cases.

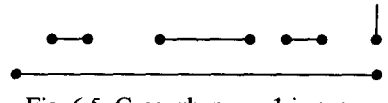


Fig. 6.5. Case when $g-1$ is even.

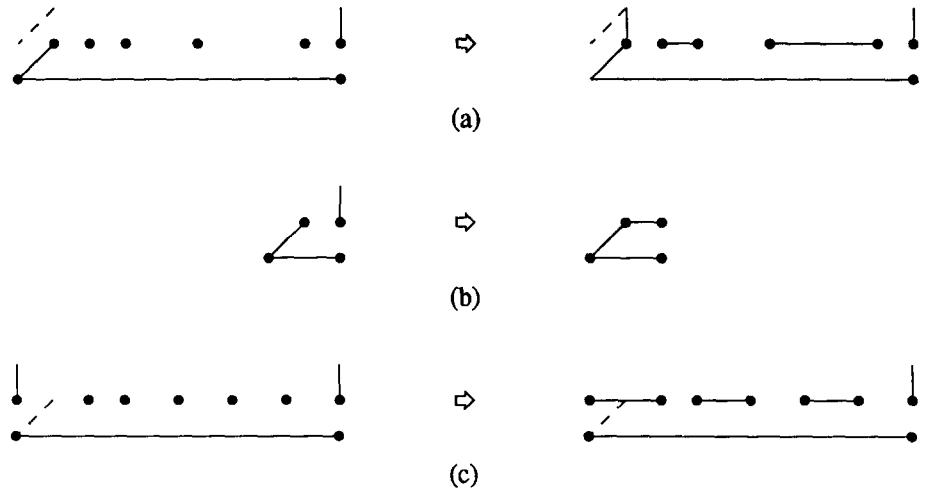


Fig. 6.6. Case when $g-1$ is odd.

Case 1: There is a diagonal form v_e to v_{t_1} (v_{J_1}) and $x_d - x_{t_1} > 1 (x_d - x_{J_1} > 1)$. From Fig. 6.4 one can verify that the diagonal that occupies the position of the dotted line in Fig. 6.6(a) cannot be part of the diagonal diagram. For this case we introduce a vertical line segment connecting v_{t_1} (v_{J_1}) to the top (bottom) boundary of $G(R)$ and for each pair of nodes $v_{t_{2k}}$ and $v_{t_{2k+1}}$ ($v_{J_{2k}}$ and $v_{J_{2k+1}}$), $1 \leqslant k \leqslant g/2 - 1 (1 \leqslant k \leqslant h/2 - 1)$ if $g - 1 \geqslant 3 (h - 1 \geqslant 3)$. See Fig. 6.6(a).

Case 2: There is a diagonal from v_e to v_{t_1} (v_{J_1}) and $x_d - x_{t_1} = 1$ ($x_d - x_{J_1} = 1$). In this case we replace the vertical line segment incident to v_{t_t} (v_{J_k}) by a horizontal line segment joining v_{t_1} and v_{t_t} (v_{J_1} and v_{J_k}). See Fig. 6.6(b).

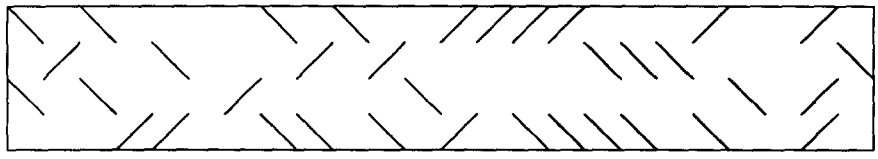
Case 3: Otherwise. Let v_{t_0} (v_{J_0}) be the nearest left neighbor of v_{t_1} (v_{J_1}). We eliminate the vertical line segment starting at the top (bottom) boundary and ending at node v_{t_0} (v_{J_0}) and connect each pair of nodes $v_{t_{2k}}$ and $v_{t_{2k+1}}$ ($v_{J_{2k}}$ and $v_{J_{2k+1}}$), $0 \leqslant k \leqslant g/2 - 1$ ($0 \leqslant k \leqslant h/2 - 1$), by a horizontal line segment. See Fig. 6.6(c).

Lemma 6.5. *The connections performed by procedure* HORIZONTAL *generate a partial legal partition, i.e. we obtain a partial partition in which all nodes that appear in columns* $1, 2, \ldots, x_d$ *are connected by a vertical or horizontal line segments.*

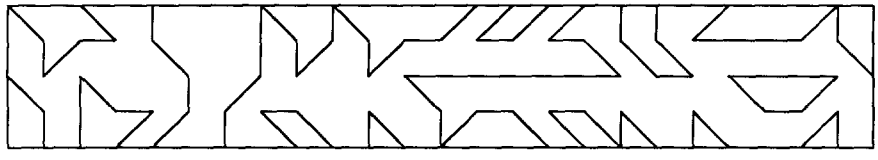
Proof. Since the proof is straightforward it will be omitted. □

Our algorithm is defined below.

procedure 4ROW-ASSIGN
Let $v_1, \ldots, v_q$ be the node list in the sorted order defined above;
let $c \leftarrow 1$;
while $c \leqslant q$ **do**



Diagonal Diagram



Legal Partition

Fig. 6.7. Legal partition by algorithm 4ROW-ASSIGN.

begin
invoke procedure VERTICAL to connect node v_c if possible;
if procedure VERTICAL fails to connect node v_c
then invoke procedure MODIFY to connect node v_c and all nodes in column x_c;
if node v_d with $x_d > x_c$ is connected by a horizontal line
then invoke procedure HORIZONTAL to connect the nodes in the set CMPD $= \{v_i \mid x_c < x_i \leqslant x_d\}$;
let c be the smallest integer such that v_c is not connected, if all nodes are connected let $c \leftarrow q + 1$;
end
end of procedure 4ROW-ASSIGN

Theorem 6.1. *Algorithm* 4ROW-ASSIGN *generates a three-layer assignment for any four-row planar layout in* $O(n)$ *time, where n is the number of columns in R.*

Proof. The proof for the time complexity bound is straightforward and the proof that our algorithm generates a three-layer assignment for any four-row planar layout follows from the above discussion. □

A diagonal diagram D and a legal partition P of D constructed by algorithm 4ROW-ASSIGN are shown in Fig. 6.7.

7. A linear layer assignment algorithm for layouts with a fixed number of rows

Even though the problem of determining whether a given layout is three-layer wirable is NP-complete, when the number of rows in a layout is bounded by some

fixed constant, the layer assignment problem can be solved in linear time. In this section we give a linear time dynamic programming layer assignment algorithm that generates a wiring for the given layout if one exists.

Consider an m-row layout W on R. If between two adjacent columns there are no horizontal wires, then the problem can be broken into a set of problem that can be solved independently of each other. Hereafter, we assume that between every two adjacent columns in W there is at least one horizontal wire. We denote the layout W restricted to the rectangle determined by the horizontal lines with y-coordinate values 0 and $m+1$ and the vertical lines with x-coordinate values $i-(1/2)$ and $i+(1/2)$, $1 \leqslant i \leqslant n$ as $W(i)$. We treat $W(i)$ as a 'switching box'. The horizontal wires entering a 'switching box' $W(i)$ from the left are called the *input wires of* $W(i)$ and the horizontal wires leaving $W(i)$ from the right of $W(i)$ are called the *output wires of* $W(i)$. Switching box $W(i)$ has $u \leqslant m$ input wires and $v \leqslant m$ output wires. A feasible layer assignment for $W(i)$ is a mapping of the edges in $W(i)$ to the layers such that if two edges (p_1, p_2) and (p_2, p_3) of the same wire W_j in $W(i)$ are assigned to layers L_s and L_t, respectively, and the edge (p_2, p_4) of another wire W_k in $W(i)$ is assigned to L_w, then $w > \max\{s, t\}$ or $w < \min\{s, t\}$. Let $X(i)$ represent all feasible wiring including columns $1, 2, \ldots, i-1$. We define the set STATES (i) as the set of all layer assignments for the input wires of $W(i)$ defined by the set of feasible wirings in $X(i)$. Clearly, since there are no more than m horizontal input wires just before $W(i)$, the number of elements in each set STATES(i) is at most 3^m. Our algorithm computes the sets STATES(i) for $i = 0, 1, \ldots, n$. If for some $1 \leqslant i < n$, STATES$(i) = \varnothing$, then we know there is no solution. On the other hand, if each of these sets contains at least one element, then there is at least one feasible wiring. A feasible wiring can be obtained from the sets STATES(i). Initially, STATES(0) consists of all possible layer assignments for the input wires for $W(1)$. Clearly, there are no more than 3^m of such layer assignments. From STATE(i) we compute STATE$(i+1)$ by testing for each element in STATE(i) and each possible element in STATE$(i+1)$ whether or not there is a layer assignment for each unit vertical wire segment in $W(i)$ such that wires from different nets are not electrically common. Note that there are most 3^m possible elements in STATES(i) and STATES $(i+1)$, respectively. Since the number of unit vertical wire segments in $W(i)$ is at most $m+1$, there are at most 3^{m+1} layer assignments for the vertical wires.

```
procedure mROW-ASSIGN
  let STATES(0) = {all possible layer assignments for the input wires to
  W(1)};
  for i = 1 to n do
    begin
      Compute the set STATES(i) from STATES(i − 1) and W(i);
      if STATES(i + 1) is empty then there is no three-layer wiring possi-
      ble for the layout W
        then stop, there is no wiring;
      end
```

Use STATES(0),...,STATES(n) to find and output the wiring solution;

end of procedure mROW-ASSIGN

From the above discussion it is simple to see that each iteration takes no more than $O(3^{2m} * 3^{m+1})$ time. Since m is bounded by a fixed constant, we know that the total time complexity is $O(n)$.

Theorem 7.1. *For m a fixed constant, procedure* mROW-ASSIGN *generates a feasible wiring for any m row planar layout in $O(n)$ time, where n is the number of columns in the planar layout.*

Proof. The proof follows from the above discussion. □

8. Discussion

It should be pointed out that the procedure 2ROW-ASSIGN cannot be modified to wire a sublayout with a simple structure surrounding it by following the procedure described in Section 3. Similarly, procedure mROW-ASSIGN also has this restriction. However, one can design a similar dynamic programming algorithm for partitioning a diagonal diagram with a fixed number of rows in linear time. Although the time complexity of mROW-ASSIGN (and for the new version of mROW-ASSIGN) is linear, the constant is exponential on m. Many speed-up techniques may be considered. For example, at each iteration of the procedure, there may be some elements meaningless or having less chance to yield a feasible wiring in the later stages of the wiring process. One can design rules to restrict the size of the set STATE[i] at each iteration so that the total time required by the procedure is smaller. To obtain a wiring of smaller area, one may combine procedure mROW-ASSIGN with other procedures based on legal partition of diagonal diagram.

As we mentioned in Section 1, the idea of stretching a planar layout to make it wirable in less than four layers is not new. In [7] it is shown that by inserting an empty track between every two adjacent tracks of R any planar layout is two-layer wirable. In fact, this idea can be traced back to the paper [11] where the channel routing problem is considered. In contrast to the previous methods, Brady and Sarrafzadeh [2] present an algorithm to find the minimum number of stretching points for wiring in two layers a planar layout. For three layers, this minimization problem is known to be NP-hard. In our approach we try to reduce the additional wiring area as much as possible. To achieve this, we present a stretching scheme that given a planar layout and layer assignment algorithms, it explores the best possible stretchings that allow a three-layer wiring. Our scheme has the advantage that if a planar layout has only a few sublayouts with complex structures, it can be wired by taking this information into account. To take

advantage of our approach, it is necessary to develop a few layer assignment algorithms for layouts with different structures.

Gonzalez and Zheng [5] propose a classification of planar layouts. They classify planar layouts into four classes depending on the properties of their diagonal diagram. It turns out that these classes form an interesting hierarchy. They show that these layout classes have different wirability properties. For example, they show that there exist four-row planar layouts in the general layout class and seven-row planar layouts in the simplest layout class that are not three-layer wirable by partitioning diagonal diagrams. By using their wirability results they give lower bounds for best possible approximation bounds for our stretching strategy.

The stretching and wiring problem can be generalized to two dimensions, that is, one is allowed to stretch a layout horizontally and vertically. This is equivalent to inserting empty grid lines into R in both directions. In this case the stretching scheme will not be as simple as the one given in Section 3. If the number of grid lines (horizontal and vertical) is small, then the total wiring area after stretching it can be expected to be small. In [5], lower bounds for the approximation factor for different layout classes are given. The analysis of these lower bounds provides evidence that two-dimensional stretching could be much better than one-dimensional stretching. The major problem in finding good wirings with two-dimensional stretching is to devise layer assignment algorithms for layouts large in both dimensions. We believe that solving this problem is difficult.

Our stretching scheme transforms a planar layout into another layout that is three-layer wirable. The wirings generated by our algorithm maintain the same topology as the original layout. Another approach one could take is to relax the requirement of preserving the topological structure. In practice, what we really care about is the connectivity of the nets. Given a planar layout, it may be possible to rearrange the wires so that the resulting layout has a simpler structure. Thus the layer assignment problem can be generalized to transforming a layout to a new layout which maintains the same connectivity as the original layout, but which is three-layer wirable in an area which is as small as possible.

References

[1] Brady, M.L. and D.J. Brown, VLSI routing: four layers suffice, *Ad. Computing Res.* **2** (1984).

[2] Brady, M.L., and M. Sarrafzadeh, Layout stretching to ensure wirability, Proceedings of the 21st Annual Conference on Information Sciences and Systems Johns Hopkins University, March 1987, pp. 25–28.

[3] Frank, A., Disjoint paths in rectilinear grid, *Combinatorica* **2** (4) (1982).

[4] Gonzalez, T. and S. Zheng, Stretching and three-layer wiring planar layouts, Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'87), October 1987, pp. 278–281.

[5] Gonzalez, T.F. and S.Q. Zheng, Wirability of planar layouts, Technical Report #87-11, UCSB (Aug. 1987).

[6] Lipski, W. Jr, An NP-complete problem related to three-layer channel routing, *Adv. Computing Res.* **2** (1984).

[7] Mehlhorn, K. and F.P. Preparata, Routing through a rectangle, *J. ACM* **33** (1) (1986).
[8] Mehlhorn, K., F.P. Preparata and M. Sarrafzadeh, Channel routing in knock-knee mode: simplified algorithms and proofs, *Algorithmica* **1** (2) (1986) 213–221.
[9] Preparata, F.P and W. Lipski, Jr, Optimal three-layer channel routing, *IEEE Trans. Comput.* **33** (5) (May 1984) 427–437.
[10] Preparata, F.P. and M. Sarrafzadeh, Channel routing of nets bounded degree, in: P. Bertolazzi and F. Luccio, eds., *VLSI: Algorithms and Architectures* (North-Holland, 1984).
[11] Rivest, R.L., A.E. Baratz and G. Miller, Provably good channel routing algorithms, in: H.T. Kung, R. Sproull and G. Steele, eds., *VLSI Systems and Computations,* (Computer Science Press, Rockville, Maryland, 1981).
[12] Sarrafzadeh, M. and F.P. Preparata, F.P., Compact channel routing of multiterminal nets, Tech. Rep. ACT-44, Coordinated Science Lab., Univ. of Illinois (1983).