Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

A PROVABLY GOOD MULTILAYER TOPOLOGICAL PLANAR ROUTING ALGORITHM FOR MCM AND DENSE PCB DESIGNS

Jason Cong

April 1991
CSD-910014

A Provably Good Multilayer Topological Planar Routing Algorithm For MCM and Dense PCB Designs

Jason Cong

Department of Computer Science
University of California at Los Angeles
Los Angeles, CA 90024, U. S. A.

Abstract

Given a number of routing layers, the multilayer topological planar routing problem is to choose a maximum (weighted) set of nets so that each net in the set can be topologically routed entirely in one of the given layers without crossing other nets. This problem has important application in the layout design of VLSI circuits, multichip modules (MCMs) and high-density printed circuit boards (PCBs). In this paper, we present a provably-good approximation algorithm for the multilayer topological planar routing problem. Our algorithm, called the iterative-peeling algorithm, can always find a solution whose weight is at least $1 - \frac{1}{e} \approx 63.2\%$ of the weight of an optimal solution. The algorithm works for multi-terminal nets and arbitrary number of routing layers. When the number of routing layers is fixed, we have even tighter performance bounds. In particular, the performance-ratio of the iterative-peeling algorithm is at least 75% if there are two routing layers and is at least 70.4% if there are three routing layers. Experimental results confirm that our algorithm can always route majority of the nets without using vias even when the number of routing layers is fairly small.

1. Introduction

Advances in VLSI fabrication technology have made it possible to use more than two routing layers for interconnections. Many VLSI chips have been designed using three or four metal layers for routing. Multichip modules (MCMs) and high density printed circuit boards (PCBs) may use even more layers for interconnections. For example, the multi-chip module developed for the IBM 3081 mainframe has 33 layers of molybdenum conductors (including 1 bonding layer, 5 distribution layers, 16 interconnection layers, 8 voltage reference layers, and 3 power distribution layers [BlBa82, Bl83]). Fujitsu's latest supercomputer, the VP-2000, uses ceramic PCBs with over 50 interconnection layers [HaYY90]. These recent developments in VLSI fabrication and packaging technology raise many interesting and important multilayer interconnection problems.

In this paper, we shall study the multilayer topological planar routing problem. The objective is to choose a maximum (weighted) set of nets so that each net in the set can be topologically routed entirely in one of the given layers without crossing other nets. Our research on the multilayer topological planar routing problem is motivated by a number of applications:

(1) In the design of VLSI circuits, we usually want to route each of the critical nets (such as the power and ground nets and the clock nets) in one of the 'preferred' layers (i.e. metal layers);

(2) In the design of MCMs and dense PCBs, the large number of interconnection layers offers an excellent opportunity for planar routing. Thus, we want to route most nets each in a single layer without using vias. For high-performance MCMs or PCBs, vias not only increase the manufacture cost but also degrade the system performance since they form inductive and capacitive discontinuities and cause reflections when the wires have to be modeled as transmission lines [Ba90];

(3) If we can topologically route all (or most) of the nets each in a single layer, the detailed routing problem is greatly simplified. We can carry out planar routing for each layer independently and several effective methods (such as rubber-band routing [LeMa85, DaKJ90]) have been developed for the planar routing problem.

All these applications require efficient solutions to the multilayer topological planar routing problem. Unfortunately, solving the multilayer topological planar routing problem is computational difficult. Cong and Liu showed that the multilayer topological planar routing problem is NP-complete [CoLi90]. The problem remains NP-complete even when the routing region is restricted to a two-layer switchbox [SaLe89]. Polynomial time optimal solutions to the multilayer topological planar routing problem were developed for a special type of channels, called crossing channels. The unweighted case for crossing channels were solved by Rim, Kashiwabara and Nakajima [RiKN89]. The general weighted case for crossing channels were solved by Cong and Liu [CoLi90] and by Sarrafzadeh and Lou [SaLo90] independently. However, there is no effective solution to the multilayer topological planar routing problem for general routing regions.

In this paper, we present a provably-good approximation algorithm for the multilayer topological planar routing problem which is applicable to switchboxes (or arbitrary rectilinear polygons), channels (including L-shaped and staircase channels), and general routing regions. Our algorithm, called the iterative-peeling algorithm, can always find a solution whose weight is at least $1-\frac{1}{e} \approx 63.2\%$ of the weight of an optimal solution. The result holds for multi-terminal nets and arbitrary number of routing layers. When the number of routing layers is fixed, we have even tighter performance bounds. In particular, the performance-ratio of the iterative-peeling algorithm is at least 75% if there are two routing layers and is at least 70.4% if there are three routing layers. According to Lemma 1 in [Ma84], these results also lead to provably good solutions to the multilayer topological via minimization problem. We tested our algorithm on a number of switchbox and channel routing benchmark examples, our algorithm can always route most nets without using vias even when the number of routing layers is small.

The remainder of the paper is organized as follows: In Section 2, we present the problem formulation; In Section 3, we present an overview of our algorithm and analysis its performance; In Section 4, we describe the details of the algorithm for switchboxes, channels, and general routing regions and analysis the time complexity of the algorithm in each case. Experimental results are presented in Section 5. Section 6 discusses the future extensions of our work.

2. Formulation of the Problem

A *routing problem* consists of a set of nets N and a routing region. A *routing region* is a layered routing area enclosed by an external boundary with (possibly) a number of blocks (obstacles) insides of the boundary. (See Fig. 1.) Terminals are located either on the external boundary or on the boundaries of the blocks. Routing over the blocks is prohibited. A *net* is a set of terminals to be connected. Each net $a \in N$ is assigned a positive weight $w(a)$ which is a measure of the priority of the net. The weight of a subset of nets $X \subseteq N$ is defined to be $w(X) = \sum_{a \in X} w(a)$. A *planar subset* is a set of nets which are topologically routable in a single layer without crossing each other[1]. A *k-planar subset* is a set of nets which can be partitioned into the union of at most k planar subset. Clearly, given a k-layer routing region, we can always route a k-planar subset without using vias (except the stacked vias bringing the terminals to their proper layers, which are indispensable). The *k-layer topological planar routing problem* (k-TPR) is that of choosing a k-planar set with the maximum weight. (Usually, assignment of the nets to the layers is also determined when the k-planar set is chosen.)

A *switchbox* is a rectangular routing region without any block inside. A *channel* is a switchbox with terminals only on the upper and lower edge of the routing region. A channel may have exits at both the left and right side of the channel. A net in a channel is called a *crossing net* if it has terminals on both the upper edge and the lower edge of the channel. If every net in a channel is a crossing net, we call the channel a *crossing channel*. It was shown in [RiKN89, CoLi90, SaLo90] that the k-TPR problem for crossing channels can be solved optimally in polynomial time.

For any heuristic algorithm H_k for the k-TPR problem, we define the *performance ratio* of H_k to be $\frac{w(S_k)}{w(S_k^*)}$, where $w(S_k)$ is the weight of the k-planar set selected by the algorithm H_k and $w(S_k^*)$ is the weight of the k-planar set computed by the optimal k-TPR algorithm. In the next two sections, we shall present an approximation algorithm for the k-TPR problem for general routing regions with performance ratio at least 63.2%.

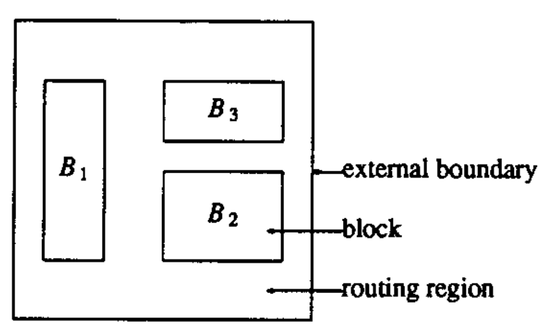


Fig. 1 A routing region.

[1] Note that we ignore the capacity constraints at this step. A set of nets are topologically routable in a single layer if their routing paths do not cross each other. However, they may not be physically routable due to capacity constraints. See Section 6 for a more detailed discussion.

3. Overview of Our Algorithm

Our algorithm for the k-TPR routing is conceptually simple. Let N be the set of nets to be routed. First, we choose a maximum weighted planar subset N_1 from N and assign N_1 to layer 1. Then, we choose a maximum weighted planar subset N_2 from the remaining nets $N - N_1$ and assign N_2 to layer 2, and so on. We repeat this process k times. Clearly, at the end $N_1 \cup N_2 \cup \cdots \cup N_k$ forms a k-planar subset. Since at each iteration we 'peel' off a maximum weighted planar subset from the remaining nets, we call this algorithm the *iterative-peeling* algorithm. Clearly, the iterative-peeling approach reduce the problem of computing a maximum weighted k-planar subset to a series computations of maximum weighted (1-)planar subsets. Therefore, we can apply several existing results on computing maximum weighted planar subsets (such as the ones in [Su87, CoLi90b, LiLS90]). Usually, choosing a maximum weighted planar subset is much easier than choosing a maximum weighted k-planar subset. For example, choosing a maximum weighed planar subset for a switchbox takes $O(n^2)$ time while choosing a maximum weighted k-planar subset for a switchbox is NP-hard ($k \geq 2$) [CoLi90]. We shall discuss the details and the complexity of the algorithms for computing a maximum weighted planar subset for various types of routing regions in the next section. Assuming that we have a procedure *max_planar_subset(N)* to compute a maximum weighted planar subset from a set of nets N, we can describe the iterative-peeling algorithm formally as follows:

Algorithm: Iterative-peeling;
1. $N' := N$;
2. **for** $i := 1$ to k **do**
   $N_i :=$ *max_planar_subset*(N'),
   $N' := N' - N_i$;
3. output $N_1 \cup N_2 \cup \cdots \cup N_k$;

end.

Fig. 2 The iterative-peeling algorithm.

Although the iterative-peeling algorithm is greedy in nature, we are able to show that it has a good performance ratio. In fact, for any number of routing layers, the iterative-peeling algorithm has a performance ratio at least 63.2%. This lower bound is established based on the following results:

Lemma 1 Let W_k^* be the weight of the optimal solution to the k-TPR problem. Let w_i be the weight of the subset N_i ($1 \leq i \leq k$) chosen by the iterative-peeling algorithm at the $i-th$

iteration. Then we have

$$w_1 \geq \frac{W_k^*}{k}$$

$$w_2 \geq \frac{W_k^* - w_1}{k}$$

$$w_3 \geq \frac{W_k^* - (w_1 + w_2)}{k}$$

......

$$w_k \geq \frac{W_k^* - (w_1 + \cdots + w_{k-1})}{k}$$

Proof Let N be the entire set of nets in the problem. Let $M = M_1 \cup M_2 \cup \cdots \cup M_k$ be an optimal solution to the k-TPR problem, where each M_i is a planar subset and M_i's are pairwise disjoint. At the end of $(i-1)$-th iteration of the iterative-peeling algorithm, the set of un-routed nets is

$$N' = N - (N_1 \cup N_2 \cup \cdots \cup N_{i-1})$$

Note that

$$|M_1 \cap N'| + |M_2 \cap N'| + \cdots + |M_k \cap N'| = |M \cap N'|$$

$$= |M \cap (N - (N_1 \cup N_2 \cup \cdots \cup N_{i-1})|$$

$$\geq |M - M \cap (N_1 \cup N_2 \cup \cdots \cup N_{i-1})|$$

$$\geq |M| - |(N_1 \cup N_2 \cup \cdots \cup N_{i-1}| = W_k^* - (w_1 + \cdots + w_{i-1})$$

By Pigeonhole's Principle, there exist at least a j $(1 \leq j \leq k)$ such that

$$|M_j \cap N'| \geq \frac{W_k^* - (w_1 + \cdots + w_{i-1})}{k}$$

Since $M_j \cap N'$ is a planar subset of N' and N_i is a maximum weighted planar subset of N' according to the iterative-peeling algorithm, we have

$$w_i = |N_i| \geq |M_j \cap N'| \geq \frac{W_k^* - (w_1 + \cdots + w_{i-1})}{k} \qquad \square$$

Lemma 2 Let $W_k = w_1 + w_2 + \cdots + w_k$ where w_i is the weight of the subset N_i produced by the iterative-peeling algorithm. Then, we have

$$W_k \geq [1 - (1 - \frac{1}{k})^k] W_k^*$$

Proof Let

$$x_1 = \frac{W_k^*}{k}$$

$$x_2 = \frac{W_k^* - x_1}{k}$$

$$x_3 = \frac{W_k^* - (x_1 + x_2)}{k}$$

......

$$x_k = \frac{W_k^* - (x_1 + \cdots + x_{k-1})}{k}$$

Then, it is easy to show that

$$x_i = \frac{k-1}{k} x_{i-1} = \frac{(k-1)^2}{k^2} x_{i-2} = \cdots = \frac{(k-1)^{i-1}}{k^{i-1}} x_1 = \frac{(k-1)^{i-1}}{k^{i-1}} \frac{W_k^*}{k}$$

Therefore, we have

$$\sum_{i=1}^{k} x_i = \sum_{i=1}^{k} \frac{(k-1)^{i-1}}{k^{i-1}} \frac{W_k^*}{k} = [1 - (1 - \frac{1}{k})^k] W_k^*$$

Now we show by induction that

$$\sum_{i=1}^{l} w_i \geq \sum_{i=1}^{l} x_i \qquad (3.1)$$

holds for every $1 \leq l \leq k$. When $l = 1$, inequality (3.1) holds since $w_1 = x_1$. Assume that inequality (3.1) holds for l. Then, for $l + 1$, we have

$$\sum_{i=1}^{l+1} w_i \geq \sum_{i=1}^{l} w_l + \frac{W_k^* - \sum_{i=1}^{l} w_l}{k} = \frac{W_k^* + (k-1)\sum_{i=1}^{l} w_l}{k}$$

$$\geq \frac{W_k^* + (k-1)\sum_{i=1}^{l} x_l}{k} = \sum_{i=1}^{l} x_l + \frac{W_k^* - \sum_{i=1}^{l} x_l}{k} = \sum_{i=1}^{l+1} x_i$$

Therefore, we have

$$W_k = \sum_{i=1}^{k} w_i \geq \sum_{i=1}^{k} x_i = [1-(1-\frac{1}{k})^k]W_k^* \qquad \square$$

From these two lemmas, we can conclude the following:

Theorem 1 Let β_k be the performance ratio of the iterative-peeling algorithm for the k-TPR problem. Then,

$$\beta_k \geq 1-(1-\frac{1}{k})^k. \qquad \square$$

It is easy to show that the function $f(x) = 1-(1-\frac{1}{x})^x$ is a decreasing function. Moreover,

$$\lim_{x\to\infty} 1-(1-\frac{1}{x})^x = 1-\frac{1}{e}$$

where $e \approx 2.718$. Therefore, we have

Corollary For any integer k, the performance ratio of the iterative-peeling algorithm for the k-TPR problem is as least

$$\beta_k \geq 1-\frac{1}{e} \approx 63.2\%$$

When the number of routing layers is known, we can use the formula in Lemma 2 to obtain a more precise performance ratio for the iterative-peeling algorithm. In particular, the performance ratio of the iterative-peeling algorithm is at least 75% for the 2-TPR problem and 70.4% for the 3-TPR problem. Table 1 shows the performance ratio of the iterative-peeling algorithm for the k-TPR problem for some small values of k.

# of layers	performance ratio
k	$\beta_k \geq 1-(1-\frac{1}{k})^k$
2	$\frac{3}{4} = 75\%$
3	$\frac{19}{27} \approx 70.4\%$
4	$\frac{175}{256} \approx 68.4\%$
5	$\frac{2101}{3124} \approx 67.3\%$
∞	$1-\frac{1}{e} \approx 63.2\%$

Table 1 Performance ratio of the iterative-peeling algorithm for different numbers of routing layers.

4. Computing Maximum Weighted Planar Subsets

In this section, we shall present efficient algorithms for computing a maximum weighted planar subset for various routing regions including switchboxes, regular channels and L-shaped channels, and general routing regions. These algorithms are used for implementing the procedure *max_planar_subset*(N) in the iterative-peeling algorithm presented in the preceding section.

4.1. Computing Maximum Weighted Planar Subsets for Switchboxes

Although the problem of computing a maximum weighted k-planar subset is NP-hard [CoLi90], we can compute a maximum weighted planar subset in $O(mn)$ time based on the results in [CoLi90b, LiLS90], where m is the number of terminals and n is the number of nets. Given a switchbox, we can 'cut' the switchbox at one point and 'stretch' the boundary of the

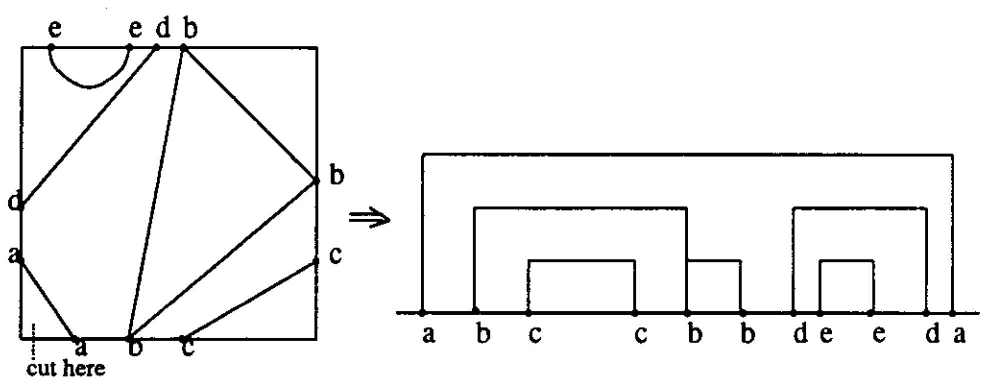


Fig. 3 Stretching the boundary of a switchbox into a straight line.
A planar subset inside the switchbox is equivalent to a planar subset on one side of the straight line.

switchbox into a straight line. It is not difficult to show that a planar routing inside the switchbox corresponds to a planar routing at one side of the straight line (see Fig. 3). line using the dynamic programming technique. Let $x_1, x_2, ..., x_m$ denote the terminals on the straight line. Let $S(i,j)$ denote a maximum weighted planar subset between terminals x_i and x_j and $M[i,j]$ be its weight. (Clearly, we want to compute $S(1,n)$.) $S(i,j)$ can be computed as follows: Suppose that x_i belongs to net a.

Case 1: If net a has some terminal outside of interval $[i,j]$, net a cannot be routed entirely within interval $[i,j]$. Therefore,

$$S(i,j)=S(i+1,j) \text{ and } M(i,j)=M(i+1,j) \tag{4.1}$$

Case 2: Suppose that net a is in the interval $[i,j]$. Let $x_{i_1}, x_{i_2}, ..., x_{i_t}$ be the terminals in net a with $x_{i_1}=x_i$. If $S(i,j)$ contains net a, $M(i,j)=\sum_{l=1}^{t-1} M(x_{i_l}+1, x_{i_{l+1}}-1) + M(x_{i_t}+1, j) + w(a)$; otherwise, $M(i,j)=M(i+1,j)$. $M(i,j)$ is given by the greater of the two values. Therefore

$$S(i,j)=\begin{cases} S(i+1,j), \text{ if } \sum_{l=1}^{t-1} M(x_{i_l}+1, x_{i_{l+1}}-1) + M(x_{i_t}+1, j) + w(a) > M(i+1,j) \\ \bigcup_{l=1}^{t-1} S(x_{i_l}+1, x_{i_{l+1}}-1) \cup S(x_{i_t}+1, j) \cup \{a\}, \text{ otherwise} \end{cases} \tag{4.2}$$

$$M(i,j)=\max\{\sum_{l=1}^{t-1} M(x_{i_l}+1, x_{i_{l+1}}-1) + M(x_{i_t}+1, j) + w(a), M(i+1,j)\}$$

Based on recursive relations (4.1) and (4.2), we can apply the dynamic programming method to compute $S(1,n)$ in $O(mn)$ time (for details, see [CoLi90b, LiLS90]). Since the iterative-peeling algorithm computes a maximum weighted planar subset k times, we have

Theorem 2 The k-TPR problem for switchboxes can be solved in $O(kmn)$ time by the iterative-peeling algorithm with performance ratio $1-(1-\frac{1}{k})^k$, where k is the number of routing layers, m is the number of terminals and n is the number of nets.

Clearly, the same result holds for any routing regions which are topologically equivalent to switchboxes, including arbitrary rectilinear polygons without holes (see Fig. 4).

4.2. Computing Maximum Weighted Planar Subsets for Channels

The main difference between a switchbox and a channel (as far as topological routing is concerned) is that the ordering of the exits at the left side and the right side of the channel is not

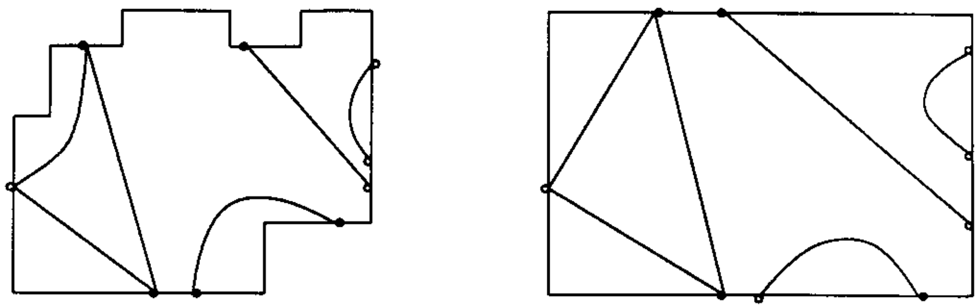


Fig. 4 Rectangular routing region which is equivalent to a switchbox.

fixed. After we choose particular orderings for the left and right exits, the channel becomes a switchbox and we can use the algorithm presented in the preceding section to compute a maximum weighted planar subset. An ordering of the (left and right) exits is *optimal* if the maximum weight of k-planar subsets of the resulting switchbox is larger than or equal to the maximum weight of k-planar subsets of the switchbox induced by any other ordering of the (left and right) exits. The main problem in this section is to find the optimal ordering of the exits of a given channel.

Given a channel, we can classify the nets in the channel as follows. A net is a *lower net* if all its terminals are on the lower edge of the channel; a net is a *upper net* if all its terminals are on the upper edge of the channel; a net is a *crossing net* if it has terminals on both the upper and the lower edge of the channel; and a net is a *through net* if it has no terminals (in this case, it must have both left and right exits). Given a net a, we use $min(a)$ and $max(a)$ to respectively denote the leftmost and rightmost positions of the terminals in net a. Our algorithm computes an optimal ordering of the left and right exits as follows: We assign the ordering of the left exits and the right exits separately from bottom to top at each side of the channel. For the left exits, we first sort all the left exits of the lower nets in increasing order of their $min(a)$'s since such an ordering minimizes the intersections of the lower nets. Then, we order all the left exits of the crossing nets

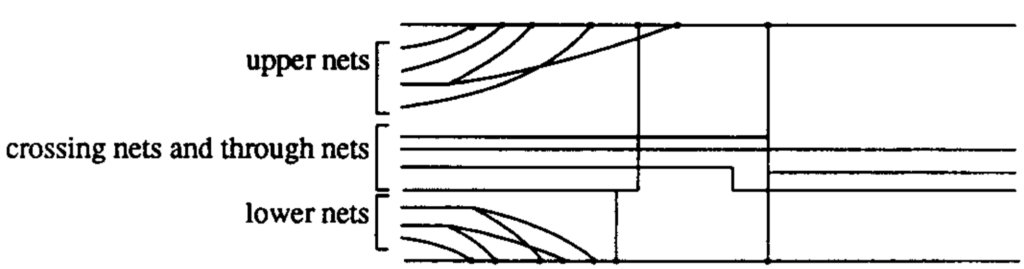


Fig. 5 Ordering of the left exits determined according to the OEO algorithm.

and through nets arbitrarily since these nets always intersect. Finally, we sort all the left exits of the upper nets in decreasing order of their $min(a)$'s since such an ordering minimize the intersections of the upper nets. The right exits can be sorted in a similar way, with the restriction that the right exits of the through nets are ordered the same as their left exits. Fig. 5 illustrates the arrangement of the exit ordering according to our algorithm. Our optimal exit ordering algorithm (OEO algorithm) can be summarized as follows:

Algorithm: Optimal_Exit_Ordering (OEO algorithm);
** Order left exits **
1. L_1 = sorted list of the left exits of the lower nets in increasing order of $min(a)$'s;
2. L_2 = list of the left exits of the crossing nets and through nets;
3. L_3 = sorted list of the left exits of the upper nets in decreasing order of $min(a)$'s;
4. $L = L_1$ *concatenate* L_2 *concatenate* L_3 is the ordered list of left exits (upward);
** Order right exits **
5. R_1 = sorted list of the right exits of the lower nets in decreasing order of $max(a)$'s;
6. R_2 = list of the right exits of the crossing nets and through nets,
   where the right exits of the through nets have the same order as their left exits;
7. R_3 = sorted list of the right exits of the upper nets in increasing order of $max(a)$'s;
8. $R = R_1$ *concatenate* R_2 *concatenate* R_3 is the ordered list of right exits (upward);

end.

Fig. 6 Algorithm for determining an optimal exit ordering in a channel.

It is not difficult to show that the OEO algorithm runs in $(n \log n)$ time where n is the number of nets since sorting the nets is the most time-consuming operation. The following result shows that the OEO algorithm indeed produces the optimal ordering of the left and right exits of a given channel.

Lemma 3 Given a channel C, let $\hat{B}$ be the switchbox induced by the exit ordering produced by the OEO algorithm and B be the switchbox induced by any exit ordering. Then, any k-planar subset in B is also a k-planar subset in $\hat{B}$.

Proof Given a switchbox SB, we can define the intersection graph $G(SB)$ of SB as follows: Each node in $G(SB)$ represents a net. An undirected edge connects nets a and b in $G(SB)$ if and only if a and b cannot be topologically routed in the same layer (in this case, we say that a *intersects* b). Clearly, the k-TPR problem for switchbox SB is equivalent to the problem of finding a maximum weighted k-colorable subgraph of $G(SB)$. We shall show that

$G(\hat{B})$ is a spanning subgraph of $G(B)$, which gives us a proof of the lemma.

According the construction of the OEO algorithm, no two through nets intersect in $\hat{B}$. Moreover, a through net does not intersect a lower net or upper net in $\hat{B}$. Furthermore, a lower net does not intersect a upper net in $\hat{B}$. Therefore, if two nets intersect in $\hat{B}$, there are six possibilities:

(1) Both nets are lower nets;

(2) Both nets are upper nets;

(3) Both nets are crossing nets;

(4) One net is a crossing net and the other is a lower net;

(5) One net is a crossing net and the other is a upper net;

(6) One net is a crossing net and the other is a through net;

It is straight forward to verify that for all six cases the two nets also intersect in B. In fact, we need only to verify cases (1), (3), and (4) since cases (1) and (2) are similar and cases (4) and (5) are similar. Case (6) is obvious since a crossing net and a through net always intersect regardless the ordering of the exits. We leave it to the reader to verify that for cases (1), (3) and (4) the two nets indeed also intersect in B. □

After we have computed the optimal exit ordering, we may use the algorithm in the preceding section to compute a maximum weighted planar subset for the corresponding switchbox. Thus, the complexity of computing a maximum weighted planar subset of a channel is $O(n \log n) + O(mn) = O(mn)$, and we have

Theorem 3 The k-TPR problem for channels can be solved in $O(kmn)$ time by the iterative-peeling algorithm with performance ratio $1-(1-\frac{1}{k})^k$, where k is the number of layers, m is the number of terminals and n is the number nets.

Clearly, the algorithm presented in this section applies not only to rectangular channels but also to other routing regions which are equivalent to rectangular channels, including L-shaped channels and staircase channels (see Fig. 7).

4.3. Computing Maximum Weighted Planar Subsets for General Routing Regions

According to the results in [LiLS90], the problem of computing a maximum weighted planar subset for general routing regions is NP-hard (based on a reduction from the problem of finding a maximum planar subset of line segments in the plane). However, a maximum weighted planar subset in a general routing region can be computed in polynomial time when the number

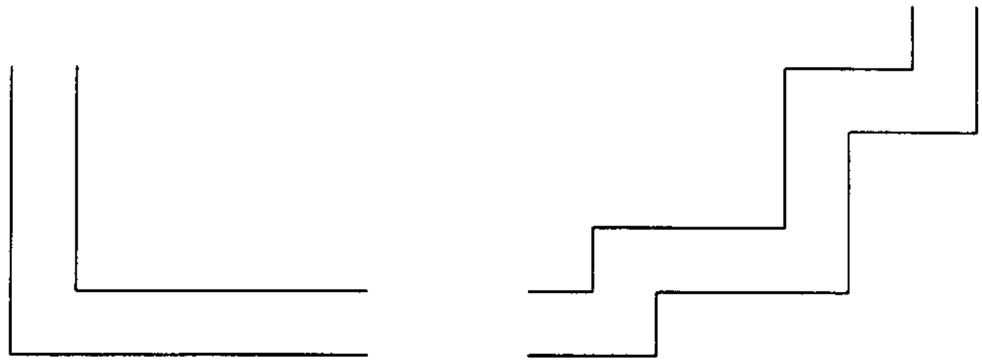


Fig. 7 L-shaped channels and staircase channels.

of blocks in the routing region is fixed. In particular, the algorithm runs in $O(n^{b+1}m)$ time, where b is the number of blocks, n is the number of nets and m is the number of terminals. Such a pseudo-polynomial time algorithm is based on the following key observation. Suppose we decide to choose net a in the planar subset in the final solution. If net a has pins on blocks $B_1, B_2, ..., B_s$, we may merge blocks $B_1, B_2, ..., B_s$ into one block as connected by net a since no other net can cross net a in the final solution. For example, net a connects blocks A, B, C in Fig. 8(a). If we decide to choose net a in the final solution, we may merge blocks A, B, C into one block as shown in Fig. 8(b). Based this observation, we may carry out a breadth-first search to construct a maximum weighted planar subset. At each level, we try to add each net in the current un-routed set to the planar subset being constructed. Thus, the branching of breadth-first search at each node is $O(n)$, where n is the total number of nets. Each time we include a net connecting several blocks, we reduce the total number of blocks by at least one. Thus, the search tree has height as most $O(b)$ and at each leaf node of the search tree no net has pins on more than one blocks. Clearly, at each leaf node we could compute a maximum weighted planar set of the

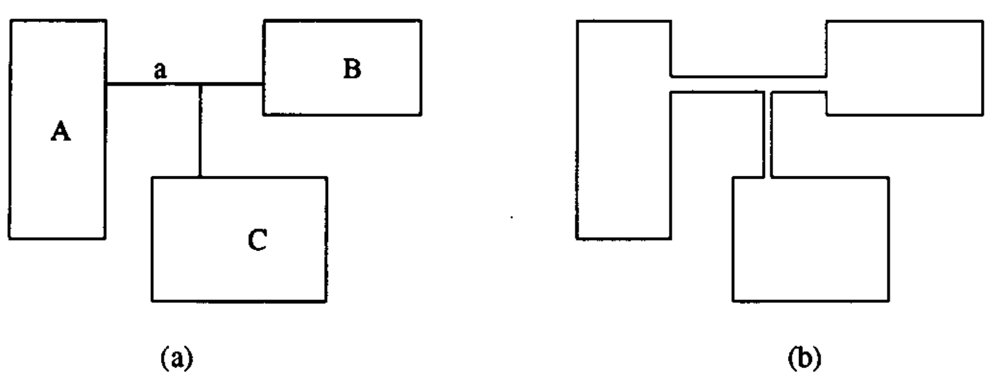


Fig. 8 Merging several blocks into one block after choosing net a.

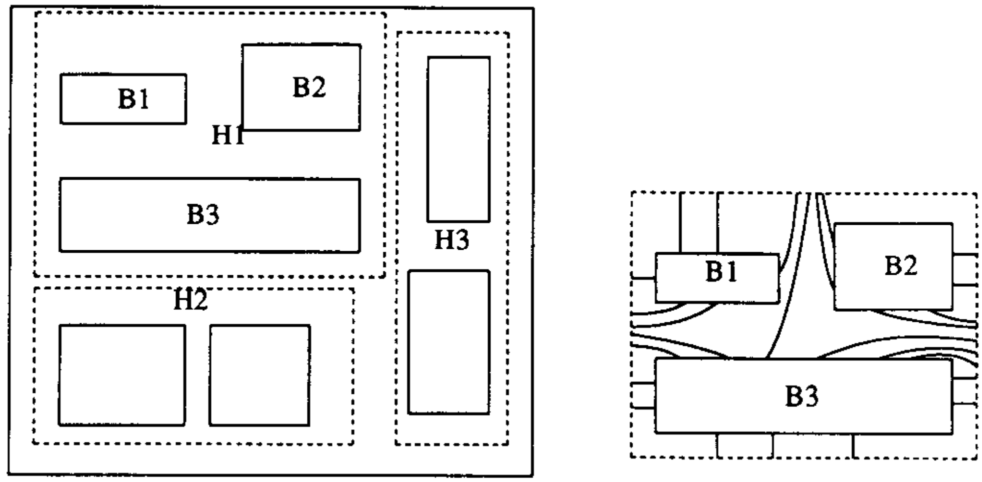
remaining nets using the algorithm in Section 4.1 for each block. Therefore, the algorithm runs in $O(n^{b+1}m)$ time. For details, see [LiLS90]. Based on this discussion and the result of Section 3, we have the following result.

Theorem 4 The k-TPR problem for general routing regions can be solved in $O(kmn^{b+1})$ time by the iterative-peeling algorithm with performance ratio $1-(1-\frac{1}{k})^k$, where b is the number of blocks, k is the number of routing layers, m is the number of terminals and n is the number of nets. □

When the number of blocks is large, the breadth-first search algorithm for constructing a maximum weighted planar subset would be quite inefficient. In this case, we may group several blocks into a *hyperblock* to reduce the total number of blocks in the design. Moreover, we shall map the terminals on the original blocks to the boundaries of the hyperblocks in a planar fashion so that a topological planar routing for the hyperblocks will also yield a topological planar routing for the original design as well. As an example, Fig. 9(a) shows one way of grouping 7 blocks in a design into 3 hyperblocks. Fig. 9(b) shows the planar mapping of the terminals on blocks B_1, B_2 and B_3 onto the boundary of the hyperblock H_1. Clearly, a topological planar routing for the hyperblocks H_1, H_2, and H_3 will also lead a topological planar routing in the original design. There are usually more than one way of carrying out the planar mapping of the terminals from the original blocks to the hyperblocks. Although we can always find a maximum weighted planar subset for the hyperblocks, some planar mappings of the terminal may lead to a sub-optimal planar subset in the original design.

5. Experimental Results

We implemented the iterative-peeling algorithm in the C language on Sun SPARC workstations and tested it on a number of switchbox and channel routing benchmark examples, including Burstein's difficult switchbox routing example [BuPe83] and Deutsch's difficult channel routing example [De76]. Table 2 reports the results of the iterative-peeling algorithm on these examples. The first column shows the names of the test examples. The Burstein example is labeled as 'burs' and the Deutsch example is labeled as 'deut'. The remaining test cases are channel routing examples from [YoKu82]. For all examples, we simply assign the weight of each net to be one, i.e. we maximize the cardinality of the k-planar subset to be computed. The next five columns of Table 2 show the percentages of the nets completed using planar routing by the iterative-peeling algorithm for one to five routing layers. (Note that these values are *not* the performance ratio of the iterative-peeling algorithm; recall that the performance ratio of the algorithm is proven to be at least 63.2%.) The last column shows the number of layers needed for



(a) (b)

Fig. 9 (a) Grouping blocks into hyperblocks;
(b) Planar mapping of the terminals from the blocks to a hyperblock.

the iterative-peeling algorithm to produce planar routings for all the nets. The computation time for each example is less than 25 seconds.

From the results shown in Table 2, we have a few interesting observations:

(1) We can have a planar routing for the majority of nets even when the number of routing layers is fairly small. For example, for all the test cases in Table 2, we can route more than 60% of the nets in a planar fashion using at most of 5 routing layers. Although all the examples in Table 2 are switchbox or channel routing examples, we expect that a similar result would also hold for general routing regions since the majority of nets are usually local nets (which span over

Ex	1L	2L	3L	4L	5L	Total layer
burs	41%	58%	70%	79%	83%	9
ex1	38%	52%	66%	76%	80%	9
ex3a	45%	59%	68%	75%	79%	11
ex3b	31%	53%	68%	78%	82%	10
ex3c	37%	55%	62%	70%	75%	13
ex4b	40%	57%	68%	75%	81%	13
ex5	31%	48%	59%	70%	78%	9
ex5b	31%	48%	60%	71%	79%	11
deut	23%	38%	50%	58%	63%	18

Table 2 Percentage of nets completed by planar routing using the iterative-peeling algorithm for different number of routing layers.

only a couple of channels or switchboxes) and only a few nets are global nets (such as clock nets), assuming that we have a reasonably good placement solution. Therefore, given a relatively large number of routing layers (say, more than 4 layers) we can route most of the nets without vias if we first use the iterative-peeling algorithm for layer assignment of the nets and then carry out planar routing in each layer. Existing approaches to MCM or dense PCB routing problems either carry out three dimensional maze routing [HaYY90] or divide the routing layers into a number of x-y layer pairs then assign the nets to x-y layer pairs and carry out two-layer routing in each x-y layer pair (x-layers run horizontal wires and y-layers run vertical wires) [HoSV90]. Very likely, these approaches will use a large number of vias for interconnections. However, if we use the unreserved layer model (i.e. each layer can run both horizontal wires and vertical wires) and assign nets to layers using the iterative-peeling algorithm, we could reduce the number of vias significantly in MCM and PCB designs.

(2) Insisting on planar routing for all the net is very costly, i.e. it requires a large number of routing layers. Although we can have planar routing for over 60% of the nets in the first 5 layers, we need 4 to 13 layers to route the remaining 20% to 40% of the nets. Therefore, it is unrealistic to insist on planar routing for all the nets. It would be more practical to construct planar routing for most of the nets (especially critical nets) based on the iterative-peeling algorithm. Then, we may route the remaining nets either by a three dimensional maze router or by assigning these nets to several x-y layer pairs and carrying out two-layer routing for each layer pair.

6. Conclusions and Future Extensions

In this paper, we have presented a provably good multilayer topological planar routing algorithm based on the idea of iterative peeling. Our algorithm is easy to implement and it works for multi-terminal nets and arbitrary number of routing layers with performance ratio at least $1 - \frac{1}{e} \approx 63.2\%$. Experimental results show that our algorithm can generate planar topological routing for most of the nets using a small number of routing layers. Such an algorithm is important to the multilayer interconnection problems in the design of MCMs and dense PCBs. It can also be used for generating planar routing sketches for rubber-band based routing algorithms [LeMa85, DaKJ90] to construct detailed planar routing solutions.

One limitation of this work is that it did not take routing capacity constraints into consideration. The planarity constraint usually restricts the number of nets we could route in each layer so that the routing capacity constraints in most layers will not be violated. However, in the first one or two layers we may have a large number of nets which are topologically routable and they may exceed the physical routing capacity. We are in the process of developing a multilayer router which takes both topological and physical constraints into consideration.

The pseudo-polynomial time algorithm in [LiLS90] for computing a maximum weighted planar subset in general routing regions could be inefficient when the number of blocks is large. It would be interesting to design an efficient approximation algorithm for the maximum weighted planar subset problem in general routing regions. Grouping blocks into hyperblocks suggests one way of reducing the complexity of the pseudo-polynomial time algorithm in [LiLS90]. We would like to study this approach more carefully to determine the optimal way of grouping blocks and the best planar mapping of the terminals from the original blocks to hyperblocks.

7. Acknowledgement

The author thanks Andrew Kahng for his careful reading of the manuscript and helpful comments.

References

[Ba90] Bakoglu, H. B., *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley Publishing Company, Menlo Park, California, 1990.

[Bl83] Blodgett, A. J., "Microelectronic packaging". *Scientific American* (July 1983) pp. 86-96.

[BlBa82] Blodgett, A. J. and D. R. Barbour, "Thermal conduction module: a high performance multilayer ceramic package". *IBM Journal of Research and Development* (Jan. 1982) Vol. 26, pp. 30-36.

[BuPe83] Burstein, M. and R. Pelavin, "Hierarchical Wire Routing". *IEEE Trans. on Computer-Aided Design* (Oct. 1983) Vol. CAD-2, pp. 223-234.

[Co91] Cong, J., "A Provably Good Multilayer Topological Planar Routing Algorithm For MCM and Dense PCB Designs". *Technical Report* (April, 1991), Department of Computer Science, UCLA, Los Angeles, CA 90024.

[CoLi90] Cong, J. and C. L. Liu, "On the k-Layer Planar Subset and Via Minimization Problems". *Proc. of European Design Automation Conf.* (March, 1990) pp. 459-463.

[CoLi90b] Cong, J. and C. L. Liu, "On the Over-the-Cell Channel RoutingProblems". *IEEE Trans. Computer-Aided Design* (April, 1990) pp. 408-418.

[DaKJ90] Dai, W. M., R. Kong, J. Jue and M. Sato, "Rubber Band Routing and Dynamic Data Representation". *IEEE Int'l Conf. on Computer-Aided Design* (1990) pp. 52-55.

[De76] Deutsch, D., "A Dogleg Channel Router". *Proc. 13th Desisgn Automation Conf.* (1976) pp. 425-433.

[HaYY90] Hanafusa, A., Y. Yamashita and M. Yasuda, "Three-Dimensional Routing for

Multilayer Ceramic Printed Circuit Boards". *Proc. IEEE Int'l Conf. on Computer-Aided Design* (Nov. 1990) pp. 386-389.

[HoSV90] Ho, J. M., M. Sarrafzadeh, G. Vijayan and C. K. Wong, "Layer Assignment for Multichip Modules". *IEEE Trans. on Computer-Aided Design* (Dec. 1990) Vol. 9, pp. 1272-1277.

[LeMa85] Leiserson, C. E. and F. M. Maley, "Algorithms for Routing and Testing Routability of Planar VLSI Layouts". *Proc. ACM Symposium on Theory of Computing* (1985) pp. 69-78.

[LiLS90] Liao, K. F., D. T. Lee and M. Sarrafzadeh, "Planar Subset of Multi-terminal Nets". *Integration, the VLSI Journal* (Sept. 1990) Vol. 10, pp. 184-190.

[Ma84] Marek-Sadowska, M., "An unconstrained topological via minimization". *IEEE Trans. on Computer-Aided Design* (1984) Vol. CAD-3, pp. 184-190.

[RiKN89] Rim, C. S., T. Kashiwabara and K. Nakajima, "Exact Algorithms for Multilayer Topological Via Minimization". *IEEE Trans. on CAD* (1989) Vol. 8, pp. 1165-1173.

[SaLe89] Sarrafzadeh, M. and D. T. Lee, "A New Approach to Topological Via Minimization". *IEEE Trans. Computer-Aided Design* (1989) Vol. 8, pp. 890-900.

[SaLo90] Sarrafzadeh, M. and R. D. Lou, "Maximum k-Covering in Transitive Graphs". *IEEE Proc. Int'l Sym. on Circuits and Systems* (May 1990) pp. 332-335.

[Su87] Supowit, K., "Finding a Maximum Plannar Subset of a Set of Nets in a Channel". *IEEE Trans. Computer-Aided Design* (1987) Vol. 6, pp. 93-94.

[YoKu82] Yoshimura, T. and E. S. Kuh, "Efficient Algorithms for Channel Routing". *IEEE Trans. on Computer Aided Design of ICAS* (Jan. 1982) Vol. CAD-1, pp. 25-35.