

A Provably Good Multilayer Topological Planar Routing Algorithm in IC Layout Designs

Jingsheng (Jason) Cong, *Member, IEEE*, Moazzem Hossain, and Naveed A. Sherwani, *Member, IEEE*

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 applications in the layout design of multilayer IC technology which has become available recently. In this paper, we present a provably good approximation algorithm for the multilayer topological planar routing problem. Our algorithm, called the iterative-peeling algorithm, finds a solution whose weight is guaranteed to be at least $1 - (1/e) \approx 63.2\%$ of the weight of an optimal solution. The algorithm works for multiterminal nets and arbitrary number of routing layers. For fixed number of routing layers, we have even tighter performance bounds. In particular, the performance ratio of the iterative-peeling algorithm is at least 75% for two-layer routing, and is at least 70.4% for three-layer routing. Experimental results confirm that our algorithm can always route a majority of the nets without using vias, even when the number of routing layers is fairly small.

I. INTRODUCTION

ADVANCES in VLSI fabrication technology have made it possible to use more than two routing layers for interconnections. In recent years, many VLSI chips have been designed using three or more metal layers for routing, for example, the Motorola 2900ETL macrocell array is a bipolar gate array with three metal layers for routing, and the masterslice array in the IBM 4331 system uses four metal layers for routing. Several algorithms have been proposed for the multilayer routing problem (for example, [4], [8], [11]). The primary goal of these approaches was to reduce the total routing area. 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 the following applications.

Manuscript received May 13, 1991; revised March 23, 1992. This work was supported in part by the National Science Foundation under Grant MIP-9110511 and Grant USE-9052346, and by an ACM SIGDA Design Automation Scholarship. This paper was recommended by Associate Editor R. H. J. M. Otten.

J. Cong is with the Department of Computer Science, University of California at Los Angeles, Los Angeles, CA 90024.

M. Hossain and N. A. Sherwani are with the Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008.

IEEE Log Number 9201151.

1) In the layout design of multilayer IC circuits, we usually want to compute a *planar* routing solution for each of the critical nets (such as the power and ground nets and the clock nets) so that these nets can be routed in their reserved layers.

2) A good topological planar routing solution reduces the number of vias used in the final layout. For high-performance circuits, it is known that vias not only increase the fabrication 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 [1].

3) If we can topologically route the majority of the nets each in a single layer, the detailed routing problem in multilayer IC designs is greatly simplified. We can carry out planar routing for each layer independently. Several effective methods (such as rubberband routing [9], [15]) have been developed for the planar routing problem.

All these applications require efficient solutions to the multilayer topological planar routing problem. The classical two-layer topological planar routing problem has been studied extensively by CAD researchers because of its practical and theoretical importance. Unfortunately, solving the multilayer topological planar routing problem is computationally difficult. Cong and Liu showed that the multilayer topological planar routing problem is NP-complete [6]. The problem remains NP-complete, even when the routing region is restricted to a two-layer switchbox [19]. Polynomial time optimal solutions to the multilayer topological planar routing problem were developed for a special type of channel, called crossing channels. The unweighted case for crossing channels was solved by Rim, Kashiwabara, and Nakajima [18]. The general weighted case for crossing channels was solved by Cong and Liu [6] and by Sarrafzadeh and Lou [20] 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, finds a solution whose weight is guaranteed to be at least $1 - (1/e) \approx 63.2\%$ of the weight of an optimal solution. The

result holds for multiterminal 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% for two-layer routing, and is at least 70.4% for three-layer routing. According to [17, lemma 1], these results also lead to provably good solutions to the multilayer topological via minimization problem. When we tested our algorithm on a number of switchbox and channel routing benchmark examples, we found it can always route most nets without using vias, even when the number of routing layers is small.

Another important application of multilayer planar routing arises in the design of multichip modules (MCM's) and high-density printed circuit boards (PCB's), where many more routing layers are used for interconnections. For example, the MCM 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 [2], [3]). Fujitsu's latest supercomputer, the VP-2000, uses ceramic PCB's with over 50 interconnection layers [12]. However, the routing model for MCM's and PCB's is different from the routing model for IC's, because in MCM and PCB designs the routing space is no longer decomposed into channels and switchboxes. Instead, routing in MCM's and PCB's is carried out in the entire substrate or board, which leads to an immense multilayer general area routing problem. Due to such a difference, the solution technique presented in this paper is not directly applicable to MCM or PCB designs. The reader may refer to [13] and [14] for multilayer MCM or PCB routing algorithms.

The remainder of the paper is organized as follows. In Section II, we present the problem formulation. In Section III, we present an overview of our algorithm and analyze its performance. In Section IV, we describe the details of the algorithm for switchboxes, channels, and general routing regions, and analyze the time complexity of the algorithm in each case. Experimental results are presented in Section V. Section VI discusses the future extensions of our work.

II. FORMULATION OF THE PROBLEM

We shall formulate the topological planar routing problem in multilayer IC designs. 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) inside of the boundary (see Fig. 1). Terminals are located on either 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

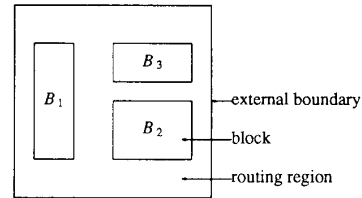


Fig. 1. A routing region.

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 (obstacle) 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 its left and right side. 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 [6], [18], [20] that the k -TPR problem for crossing channels can be solved optimally in polynomial time.

For any heuristic algorithm H_k of the k -TPR problem, we define the *performance ratio* of H_k to be $w(S_k)/w(S_k^*)$, 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 of the k -TPR problem for general routing regions with performance ratio at least 63.2%.

III. 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. At the i th step, we choose a maximum weighted planar subset N_i from $N - N_1 \cup N_2 \cup \dots \cup N_{i-1}$. We repeat this process k times. Clearly, at the end $N_1 \cup N_2 \cup \dots \cup N_k$ forms a

¹Note that we ignore the capacity constraints at this step. A set of nets is topologically routable in a single layer if their routing paths do not cross each other. However, it may not be physically routable due to capacity constraints of certain channels or switchboxes. Due to the high complexity of the multilayer routing problem, it is common that complete routing is carried out in two steps: first, a topological routing solution is computed, and then the physical routing solution is generated. See Section VI for more discussion.

k -planar subset, because each N_i is a planar subset by construction, and by definition the union of k such subsets 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 reduces the problem of computing a maximum weighted k -planar subset to a series of computations of maximum weighted planar subsets. Therefore, we can apply several existing results on computing maximum weighted planar subsets (such as the ones in [7], [16], [21]). Usually, choosing a maximum weighted planar subset is much easier than choosing a maximum weighted k -planar subset. For example, choosing a maximum weighted planar subset for a switchbox takes $O(n^2)$ time while choosing a maximum weighted k -planar subset for a switchbox is NP-hard ($k \geq 2$) [6]. 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 $\text{max_planar_subset}(N)$ to compute a maximum weighted planar subset from a set of nets N , we can describe the iterative-peeling algorithm formally, as shown in Fig. 2.

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

$$\begin{aligned} w_1 &\geq \frac{W_k^*}{k} \\ w_2 &\geq \frac{W_k^* - w_1}{k} \\ w_3 &\geq \frac{W_k^* - (w_1 + w_2)}{k} \\ &\vdots \\ w_k &\geq \frac{W_k^* - (w_1 + \dots + w_{k-1})}{k}. \end{aligned}$$

Proof: Let N be the entire set of nets in the problem. Let $M = M_1 \cup M_2 \cup \dots \cup M_k$ 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 unrouted nets is

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

Algorithm: Iterative-peeling;

1. $N' := N$;
 2. **for** $i := 1$ **to** k **do**
 $N_i := \text{max_planar_subset}(N')$,
 $N' := N' - N_i$;
 3. **output** $N_1 \cup N_2 \cup \dots \cup N_k$;
- end.**

Fig. 2. The iterative-peeling algorithm.

Note that

$$\begin{aligned} &|M_1 \cap N'| + |M_2 \cap N'| + \dots + |M_k \cap N'| \\ &= |M \cap N'| \\ &= |M \cap (N - (N_1 \cup N_2 \cup \dots \cup N_{i-1}))| \\ &= |M - M \cap (N_1 \cup N_2 \cup \dots \cup N_{i-1})| \\ &\geq |M| - |(N_1 \cup N_2 \cup \dots \cup N_{i-1})| \\ &= W_k^* - (w_1 + \dots + w_{i-1}). \end{aligned}$$

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

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

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

$$\begin{aligned} w_i = |N_i| &\geq |M_j \cap N'| \\ &\geq \frac{W_k^* - (w_1 + \dots + w_{i-1})}{k}. \quad \square \end{aligned}$$

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

$$W_k \geq \left| 1 - \left(1 - \frac{1}{k}\right)^k \right| W_k^*.$$

Proof: Let

$$\begin{aligned} x_1 &= \frac{W_k^*}{k} \\ x_2 &= \frac{W_k^* - x_1}{k} \\ x_3 &= \frac{W_k^* - (x_1 + x_2)}{k} \\ &\vdots \\ x_k &= \frac{W_k^* - (x_1 + \dots + x_{k-1})}{k}. \end{aligned}$$

Then it is easy to show that

$$\begin{aligned} x_i &= \frac{k-1}{k} x_{i-1} = \frac{(k-1)^2}{k^2} x_{i-2} = \dots \\ &= \frac{(k-1)^{i-1}}{k^{i-1}} x_1 = \frac{(k-1)^{i-1}}{k^{i-1}} \frac{W_k^*}{k}. \end{aligned}$$

Therefore, we have

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

Now we show by induction that

$$\sum_{i=1}^l w_i \geq \sum_{i=1}^l x_i \quad (3.1) \quad \text{at least}$$

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

$$\begin{aligned} \sum_{i=1}^{l+1} w_i &\geq \sum_{i=1}^l w_i + \frac{W_k^* - \sum_{i=1}^l w_i}{k} \\ &= \frac{W_k^* + (k-1) \sum_{i=1}^l w_i}{k} \\ &\geq \frac{W_k^* + (k-1) \sum_{i=1}^l x_i}{k} \\ &= \sum_{i=1}^l x_i + \frac{W_k^* - \sum_{i=1}^l x_i}{k} = \sum_{i=1}^{l+1} x_i. \end{aligned}$$

Therefore, we have

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

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,

$$\beta_k \geq 1 - \left(1 - \frac{1}{k} \right)^k. \quad \square$$

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

$$\lim_{x \rightarrow \infty} 1 - \left(1 - \frac{1}{x} \right)^x = 1 - \frac{1}{e}$$

where $e \approx 2.718$. Therefore, we have the following corollary.

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

TABLE I
PERFORMANCE RATIO OF THE ITERATIVE-PEELING ALGORITHM FOR VARIOUS NUMBERS OF ROUTING LAYERS

No. of Layers	Performance Ratio
k	$\beta_k \geq 1 - (1 - (1/k))^k$
2	$3/4 = 75\%$
3	$19/27 \approx 70.4\%$
4	$175/256 \approx 68.4\%$
5	$2101/3124 \approx 67.3\%$
∞	$1 - (1/e) \approx 63.2\%$

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

When the number of routing layers is known, we can apply 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 I shows the performance ratio of the iterative-peeling algorithm for the k -TPR problem for some small values of k .

IV. 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.

A. Computing Maximum Weighted Planar Subsets for Switchboxes

Although the problem of computing a maximum weighted k -planar subset is NP-hard [6], we can compute a maximum weighted planar subset in $O(mn)$ time based on the results in [7] and [16], 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 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). We shall find a maximum weighted planar routing solution on one side of the line using the dynamic programming technique. Let x_1, x_2, \dots, 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

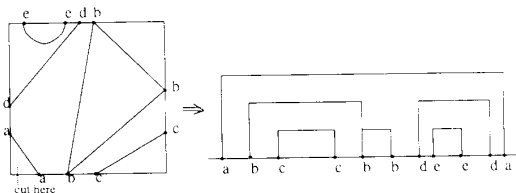


Fig. 3. Stretching the boundary of a switchbox into a straight line. A planar subset inside the switchbox is equivalent to a planar subset on one side of the straight line.

$[i, j]$, net a cannot be routed entirely within interval $[i, j]$. Therefore,

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

Case 2: Suppose that net a is in the interval $[i, j]$. Let $x_{i_1}, x_{i_2}, \dots, x_{i_t}$ be the terminals in net a with $x_{i_t} = x_i$. 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,

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

$$M(i, j) = \max \left\{ \sum_{l=1}^{t-1} M(x_{i_l} + 1, x_{i_{l+1}} - 1) + M(x_{i_t} + 1, j) + w(a), M(i + 1, j) \right\}. \quad (4.2)$$

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 [7], [16]). 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 $1 - (1 - 1/k)^k$, 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).

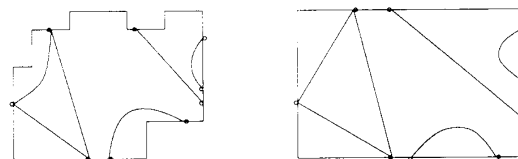


Fig. 4. Rectangular routing region which is equivalent to a switchbox.

B. 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 fixed. After we choose particular orderings for the left and right exits,² 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 addressed 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 an *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 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 minimizes 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 shown in Fig. 6.

It is not difficult to show that the OEO algorithm runs in $(n \log n)$ time where n is the number of nets, since

²Without loss of generality, we order the exits from bottom to top at the left and right sides of the channel.

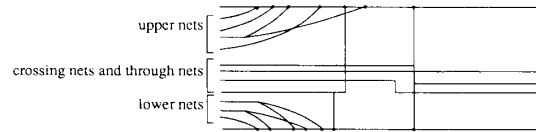


Fig. 5. Ordering of the left exits determined according to the OEO algorithm.

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.

sorting the nets is the most time-consuming operation. The following result shows that 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 an arbitrary 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 to 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 an 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 an upper net;
- 6) one net is a crossing net and the other is a through net.

It is straightforward to verify that for all six cases the two nets also intersect in B . In fact, we need to verify only 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 of 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 . \square

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 Theorem 3 results.

Theorem 3: The k -TPR problem for channels can be solved in $O(kmn)$ time by the iterative-peeling algorithm with performance ratio $1 - (1 - 1/k)^k$, 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).

C. Computing Maximum Weighted Planar Subsets for General Routing Regions

According to the results in [16], 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

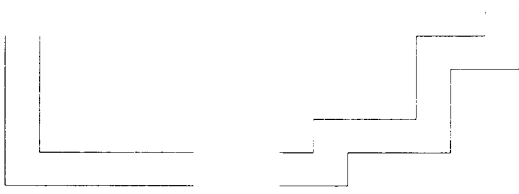


Fig. 7. L-shaped channels and staircase channels.

in polynomial time when the number 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 pseudopolynomial 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, \dots, B_k , we may merge blocks B_1, B_2, \dots, B_k 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 on 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 at 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 remaining nets using the algorithm in Section IV-A for each block. Therefore, the algorithm runs in $O(n^{b+1}m)$ time. (For details, see [16].) Based on this discussion and the result of Section III, 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 $1 - (1 - 1/k)^k$, 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. \square

When the number of blocks is large, the breadth-first search algorithm for constructing a maximum weighted planar subset could be 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 seven blocks in a design into three hyperblocks. Fig. 9(b) shows the planar mapping of the terminals on blocks B_1, B_2 , and

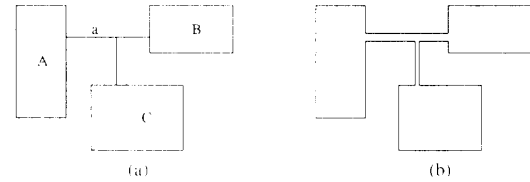
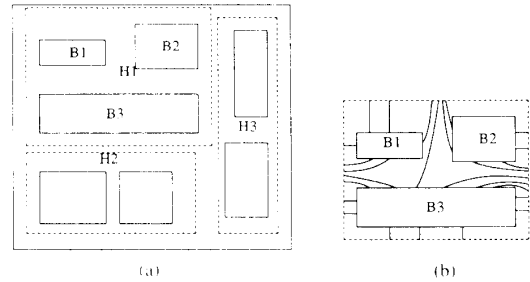
Fig. 8. Merging several blocks into one block after choosing net a .

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

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 to a topological planar routing in the original design. There is 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 terminals may lead to a suboptimal planar subset in the original design.

V. EXPERIMENTAL RESULTS

We have implemented the iterative-peeling algorithm in 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 [5] and Deutsch's difficult channel routing example [10]. Table II 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 [22]. 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 II 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 our algorithm to an optimal algorithm is proven to be at least 63.2%.) The last column shows the number of layers needed for the iterative-peeling algorithm to produce planar routings for all the nets. The computation time for each example is less than 25 s.

TABLE II
PERCENTAGE OF NETS COMPLETED BY PLANAR ROUTING USING THE
ITERATIVE-PEELING ALGORITHM WITHIN VARIOUS NUMBERS
OF ROUTING LAYERS

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

Based on the results shown in Table II, 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 II, we can route more than 60% of the nets in a planar fashion using at most five routing layers. Although all the examples in Table II 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 only a few 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 four 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.

2) Insisting on planar routing for all the nets 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 five layers, we need 4–13 layers to route the remaining 20–40% 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.

VI. 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 works for multiterminal nets and an arbitrary number of routing layers with performance ratio at least $1 - (1/e) \approx 63.2\%$. 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 in the layout design of multilayer IC technology. It can also be used for generating planar routing sketches for rubber band-based routing al-

gorithms [9], [15] to construct detailed planar routing solutions.

One limitation of this work is that it does not take routing capacity constraints into consideration. In practice, the planarity constraint is usually quite restrictive so that the planar routable nets in most layers do not exceed the physical capacity constraints. 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 pseudopolynomial time algorithm in [16] 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 pseudopolynomial time algorithm in [16]. 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.

Historical Note: The results in this paper were first obtained by Cong in [24]. The cases when $k = 2$ and $k = 4$ in Theorem 1 were derived independently for over-the-cell routing in [25] and [26], and were generalized to the case of arbitrary k by Hossain and Sherwani [27].

REFERENCES

- [1] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Menlo Park, CA, Wesley, 1990.
- [2] A. J. Blodgett, "Microelectronic packaging," *Scientific American*, pp. 86–96, July 1983.
- [3] A. J. Blodgett and D. R. Barbour, "Thermal conduction module: A high performance multilayer ceramic package," *IBM J. Res. Develop.*, vol. 26, pp. 30–36, Jan. 1982.
- [4] D. Braun *et al.*, "Chameleon: A new multi-layer channel router," in *Proc. 23rd Design Automation Conf.*, 1986, pp. 495–502.
- [5] M. Burstein, and R. Pelavin, "Hierarchical wire routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 223–234, Oct. 1983.
- [6] J. Cong and C. L. Liu, "On the k -layer planar subset and via minimization problems," in *Proc. European Design Automation Conf.*, Mar. 1990, pp. 459–463.
- [7] J. Cong, and C. L. Liu, "Over-the-cell channel routing," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 408–418, Apr. 1990.
- [8] J. Cong, D. F. Wong, and C. L. Liu, "A new approach to three- or four-layer channel routing," *IEEE Trans. Computer-Aided Design*, vol. 7, pp. 1094–1104, 1988.
- [9] W. M. Dai, R. Kong, J. Jue, and M. Sato, "Rubber band routing and dynamic data representation," in *Proc. Conf. Computer-Aided Design*, 1990, pp. 52–55.
- [10] D. Deutsch, "A dogleg channel router," in *Proc. 13th Design Automation Conf.*, 1976, pp. 425–433.
- [11] R. I. Greenberg, A. T. Ishii, and A. L. Sangiovanni-Vincentelli, "MulCh: A multilayer channel router using one, two and three layer partitions," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1988, pp. 88–91.
- [12] A. Hanafusa, Y. Yamashita, and M. Yasuda, "Three-dimensional routing for multilayer ceramic printed circuit boards," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, Nov. 1990, pp. 386–389.
- [13] J. M. Ho, M. Sarrafzadeh, G. Vijayan, and C. K. Wong, "Layer assignment for multichip modules," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 1272–1277, Dec. 1990.

- [14] K. Y. Khoo and J. Cong, "A fast multilayer general area router for MCM designs," in *Proc. EURODAC'92*, Sept. 1992, pp. 292-297.
- [15] C. E. Leiserson and F. M. Maley, "Algorithms for routing and testing routability of planar VLSI layouts," in *Proc. ACM Symp. on Theory Comp.*, 1985, pp. 69-78.
- [16] K. F. Liao, D. T. Lee, and M. Sarrafzadeh, "Planar subset of multi-terminal nets," *Integration, VLSI J.*, vol. 10, pp. 184-190, Sept. 1990.
- [17] M. Marek-Sadowska, "An unconstrained topological via minimization," *IEEE Trans. Computer-Aided Design*, vol. CAD-3, pp. 184-190, 1984.
- [18] C. S. Rim, F. Kashiwabara, and K. Nakajima, "Exact algorithms for multilayer topological via minimization," *IEEE Trans. Computer Aided Design*, vol. 8, pp. 1165-1173, Nov. 1989.
- [19] M. Sarrafzadeh and D. T. Lee, "A new approach to topological via minimization," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 890-900, Aug. 1989.
- [20] M. Sarrafzadeh and R. D. Lou, "Maximum k-covering in transitive graphs," in *IEEE Proc. Int. Symp. Circuits and Sys.*, May 1990, pp. 332-335.
- [21] K. Supowit, "Finding a maximum planar subset of a set of nets in a channel," *IEEE Trans. Computer-Aided Design*, vol. 6, pp. 93-94, 1987.
- [22] T. Yoshimura and E. S. Kuh, "Efficient algorithms for channel routing," *IEEE Trans. Computer Aided Design*, vol. CAD-1, pp. 25-35, Jan. 1982.
- [23] K. Y. Khoo and J. Cong, "Four vias are sufficient in multilayer MCM and dense PCB routing," in *IEEE Multi-Chip Module Conf.*, Mar. 1993, to appear.
- [24] J. Cong, "A Provably Good Multi-layer Topological Planar Routing Algorithm," UCLA Computer Science Department Tech. Report CSD-910014, Los Angeles, CA, Apr. 1991.
- [25] N. Holmes, N. Sherwani, and M. Sarrafzadeh, "New algorithm for over-the-cell channel routing using vacant terminals," in *Proc. 28th ACM/IEEE Design Automation Conf.*, June 1991, pp. 126-131.
- [26] —, "Algorithms for three-layer over-the-cell channel routing," in *Proc. IEEE Int. Conf. on Computer-Aided Design*, Nov. 1991, pp. 428-431.
- [27] M. Hossain and N. Sherwani, "An approximation algorithm for maximum K-colorable subgraph problem for circle graphs," Western Michigan Univ. Computer Science Dept. Tech. Rep. 1-05, Kalamazoo, MI, Mar. 1991.



Jingsheng (Jason) Cong (S'88 M'90) received the B.S. degree in computer science from Peking University in 1985, and the M.S. and Ph.D. degrees in computer science from the University of Illinois at Urbana-Champaign in 1987 and 1990, respectively.

Currently, he is an assistant professor in the Computer Science Department of the University of California, Los Angeles. From 1986 to 1990 he was a research assistant in the Computer Science Department of the University of Illinois. He

worked at the Xerox Palo Alto, CA, Research Center in 1987. He worked at the National Semiconductor Corporation in 1988. His research interests include computer-aided design of VLSI circuits, fault-tolerant design of VLSI systems, and design and analysis of efficient combinatorial and geometric algorithms. He received the Best Graduate Award from Peking University in 1985. He was awarded a DEC Computer Science Fellowship in 1988, received the Ross J. Martin Award for Excellence in Research from the University of Illinois, Urbana-Champaign, in 1989, and the NSF Engineering Research Initiation Award in 1991. He has served on several program committees. Currently he is the chairman of the 4th ACM/SIGDA Physical Design Workshop.



Moazzem Hossain received the B.S. degree in Computer Science from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 1990. Currently, he is working toward the M.S. degree in computer science at Western Michigan University, Kalamazoo, MI.

His research interests include physical design automation of VLSI circuits, design and analysis of algorithms, and graph algorithms. He is the recipient of the 1991 ACM/SIGDA Design Automation Scholarship.

Mr. Hossain is a member of ACM and ACM SIGDA.



Naveed S. Sherwani (S'85 M'87) received the M.S. and Ph.D. degrees in computer science from the University of Nebraska, Lincoln, in 1985 and 1988, respectively.

He is currently an assistant professor in the Computer Science Department at Western Michigan University, Kalamazoo, MI. His research interests include physical design automation of VLSI circuits, parallel and distributed systems, combinatorial optimization, and graph algorithms. He has published over forty refereed papers in various journals and conferences. His paper on three layer over-the-cell routing received the distinguished paper award at ICCAD-91.

Dr. Sherwani has chaired four conferences and served on the technical committees of several others.

Dr. Sherwani is a member of the ACM.