On the k-Layer Planar Subset and Topological Via Minimization Problems

Jingsheng Cong, *Member, IEEE*, and C. L. Liu, *Fellow, IEEE*

***Abstract*—An important problem in performance-driven routing is the k-layer planar subset problem which is to choose a maximum (weighted) subset of nets such that each net in the subset can be routed in one of k "preferred" layers. Related to the k-layer planar subset problem is the k-layer topological via minimization problem which is to determine the topology of each net using k routing layers, such that a minimum number of vias is used. For the case $k = 2$, the topological via minimization problem has been studied by CAD researchers for a long time because of its practical and theoretical importance. In this paper, we show that both the general k-layer planar subset problem and the k-layer topological via minimization problem are *NP*-complete. Moreover, we show that both problems can be solved in polynomial time when the routing regions are crossing channels. It can be shown that under a suitable assumption, all the channels for interblock connections in the general cell design style are crossing channels. Our algorithms are based on an efficient algorithm for computing a maximum weighted k-cofamily in a partially ordered set.**

I. Introduction

ADVANCES in VLSI fabrication technology have made it possible to use more than two routing layers for interconnections. Chips have been designed using three or four layers of metal for interconnections. Several algorithms have been proposed for the multilayer routing problem [4], [2], [12], [1], [7], [15]. The primary goal of these approaches is to reduce the total routing area. In this paper, we study two problems associated with the multilayer routing problem, namely, the k-layer planar subset problem and the k-layer topological via minimization problem. Both of these are important problems in performance-driven layout design.

The k-layer planar subset (k-PSP) is that of choosing a maximum (weighted) subset of nets such that each net in the subset can be routed entirely in one of k "preferred" layers. For example, we want to route the power bus, the ground bus, and some critical clock signal nets such that each of these nets is entirely in one of the low capacitance and low resistance layers. Liao, Lee, and Sarrafzadeh [18] studied the case when $k = 1$. They showed that 1-PSP can be solved in polynomial time when the routing region is a channel or a switchbox, but it becomes *NP*-complete when the routing region contains an arbitrary number of blocks (obstacles). In multilayer routing, we might have more than one "preferred" layers. However, the general k-PSP problem has not been studied before.

The k-layer topological via minimization problem (k-TVM) is that of determining the topological routing of a set of nets on k routing layers such that the total number of vias is minimized. It is known that an increase in the number of vias affects both yield and circuit performance. Such an increase also makes it more difficult to compact the routing solutions [5], [6]. The classical two-layer topological via minimization problem has been of both practical and theoretical interests to CAD researchers for a long time [17], [21], [30], [26], [25]. It was shown by Sarrafzadeh and Lee [26] that the 2-TVM problem is *NP*-complete. They also provided a polynomial time algorithm for the 2-TVM problem for two-shore channels. (A faster algorithm was presented later in [19] by the same authors.) However, little is known for the k-TVM problem beyond the case $k = 2$. The only known result is by Rim, Kashiwabara, and Nakajima on a polynomial time solution to the k-TVM problem for channels without local nets when the weights of all the nets are equal to one [25].

In this paper, we study the k-PSP problem and the k-TVM problem for any fixed constant k. We first show that both the general k-PSP problem and the k-TVM problem are *NP*-complete. Then, we show that these problems can be solved in polynomial time for a large class of channels, called crossing channels. Under a suitable assumption, we can show that all the channels for interblock connections in the general cell design style are crossing channels. Our algorithms remain polynomial when each net has an arbitrary positive weight. Our work improves the results in [25] for channels without local nets (equivalent to crossing channels) since their algorithm applies only to the case when all net weights are one and its generalization to the weighted cases is not obviously easy. Our work was inspired by several clever ideas in [18] and [26]. One contribution of our work is the application of some of the results from the theory of partially ordered sets. In particular, we formulate the k-PSP problem and the k-TVM problem for crossing channels as the problem of finding a maximum weighted k-cofamily in a partially

Manuscript received April 18, 1990. This work was supported in part by the National Science Foundation under Grant MIP 90-48345. This paper was recommended by Associate Editor M. Marek-Sadowska.
An extended abstract of this paper was presented at the European Design Automation Conference.
J. Cong is with the Department of Computer Science, University of California at Los Angeles, Los Angeles, CA 90024.
C. L. Liu is with the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801.
IEEE Log Number 9143304.

0278-0070/91/0800-0972$01.00

ordered set and we present a strong polynomial time algorithm for computing a maximum weighted k-cofamily in a partially ordered set with positive weights.

The remainder of the paper is organized as follows: In Section II, we introduce some definitions and basic concepts. In Section III, we present our results on the k-PSP problem. In Section IV, we present our results on the k-TVM problem. As in [26] and [25], we assume that in the k-TVM problem each net is a two-terminal net. In Section V, we present a strong polynomial time algorithm for computing a maximum weighted k-cofamily in a partially ordered set. Since the algorithms presented in this paper are probably good, we did not provide any experimental results. However, the key ideas of our algorithms are described in detail so that they may be implemented in a straightforward way following our discussions.

II. Preliminaries

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 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. Without loss of generality, we assume that all the weights are positive. The weight of a subset of nets $X \subseteq N$ is defined to be $w(X) = \Sigma_{a \in X}\, w(a)$. Given a k-layer routing region, a *k-planar subset* is a subset of nets in which each net can be routed entirely in one of the k-layers. The k-PSP is that of choosing a k-planar subset with the maximum weight. (Usually, assignment of the nets to the layers is also determined when the k-planar subset is chosen.) For a given k-layer routing region, a *topological routing solution* is a set of wires and vias such that each net is connected by some wires and vias with each wire being in a single layer and each via connecting wires in two adjacent layers. We define the *total via cost* of a topological routing solution S, denoted $c(S)$, to be $c(S) = \Sigma_{a \in N}\, w(a) \cdot v(a)$, where $v(a)$ is the number of vias used for net a in S. The k-TVM is that of finding a topological routing solution with the minimum total via cost. (When all the nets have weight one, the total number of vias is minimized.) In both the k-PSP and k-TVM problems, we assume that terminals are available on all layers, and that a via can connect only wires in two adjacent layers.

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*. Crossing channels capture a large class of channels encountered in practice. In particular, we have the following result.

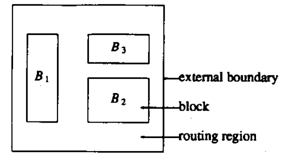


Fig. 1. A routing region.

Theorem 1: In the general cell design style, if the terminals of a net on the boundary of a block are connected on the inside of that block, then all the channels for interblock connections are equivalent to crossing channels provided that each channel is to be routed independently.

Proof: In the general cell design style [24], a circuit is partitioned into a set of blocks. Channels for interblock connections are formed by the rectangular routing regions between two blocks [22]. (See Fig. 2.) (A rectangular routing region formed by more than two blocks can be decomposed into several channels. For example, the routing region formed by B_1, B_2, and B_3 in Fig. 2 can be decomposed into two channels C_1 and C_2.) It is easy to see that the upper edge or the lower edge of each channel belongs entirely to a single block. Let us consider a particular channel C. Let a be a net in C. Without loss of generality, assume that net a has a terminal x on the upper edge of C which is part of the boundary of block B. If all other terminals in net a are also on the upper edge, then all the terminals that belong to net a in C are connected inside of block B. Thus we can simply remove net a from C. Otherwise, net a has another terminal y which is either on the lower edge of C or an exit of C. If y is on the lower edge of C, clearly, net a is a crossing net. Otherwise, y is either a left exit or a right exit of C. It is well known that a channel routing problem in which there are left and right exits can be reduced to an equivalent channel routing problem by moving the left (right) exits to the left (right) end of either the upper edge or the lower edge of the channel. In our case, we move y to the lower edge of C. Therefore, net a is a crossing net.[1] Since the same argument holds for every net, channel C is a crossing channel. □

The assumption that terminals belong to the same net on the boundary of a block B are connected inside of B is a realistic one, since in a hierarchical design all the nets in B are connected in the next level of design hierarchy. However, the assumption that each channel is to be routed independently might not be true for all general cell design systems. In some general cell design systems [23], each channel is routed independently and a switchbox router is used at every T-junction and $+$-junction to connect the exits from different channels. In this case, each channel can be treated as a crossing channel. However, in some other general cell design systems, in order to minimize the number of switchboxes, two or more channels may be

[1]Although moving y to the lower edge of C guarantees net a to be a crossing channel, we can show that there are cases, in which such a transformation may introduce an unnecessary via for net a.

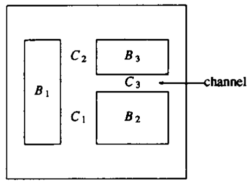


Fig. 2. Channels for interblock connections.

routed simultaneously [10], [3]. For the example shown in Fig. 2, we may route channel C_3 first. Then, we may route channels C_1 and C_2 together as one channel so that we can remove the switchbox for the T-junction. In this case, C_3 is a crossing channel, but the union of C_1 and C_2 might not always be a crossing channel. Note also that in the standard cell design style, channels are not necessarily crossing channels. Our definition of a crossing channel is equivalent to that of a two-shore channel in [18] and a channel without local net in [25].

When we introduce pseudoterminals on the lower or upper edge of a channel in the proof of Theorem 1, the ordering of these pseudoterminals can be arbitrary. We want to choose an optimal ordering such that the weight of the solution to the k-PSP problem for the resulting channel is maximum (or equivalently, the total via cost of the solution to the k-TVM problem for the resulting channel is minimum). Such an optimal ordering can be determined as follows in linear time. Suppose p_1 and p_2 are two pseudoterminals introduced at the left end of the lower edge of the channel for nets a and b, respectively. Let q_1 and q_2 be the corresponding leftmost terminals of nets a and b on the upper edge of the channel. (Since each net is a crossing net after we move the exits, the corresponding terminals q_1 and q_2 must exist.) Then, we place p_1 to the left of p_2 if and only if q_1 is to the left of q_2 so that net a does not cross net b if the terminal span of net a does not intersect the terminal span of net b. (The *terminal span* of a net is defined to be the interval from the leftmost terminal of the net to the rightmost terminal of the net, excluding the exits.) The rule for the pseudoterminals on the upper edge of the channel is similar. It is not difficult to show that these rules lead to an optimal ordering of the pseudoterminals. (This will become clearer after the discussions in Sections III-3.2 and IV-4.2.)

In the remainder of this section, we introduce several important concepts in combinatorial theory on partially ordered sets which will be used later in our algorithms. A *partially ordered set P* is a collection of elements P together with a binary relation $\leftarrow$ defined on $P \times P$ which satisfies the following conditions [20]:

1) *reflexive*, i.e., $x \leftarrow x$ for all $x \in P$;
2) *antisymmetric*, i.e., $x \leftarrow y$ and $y \leftarrow x \Rightarrow x = y$;
3) transitive, i.e., $x \leftarrow y$ and $y \leftarrow z \Rightarrow x \leftarrow y$.

We say that x and y are *related* if $x \leftarrow y$ or $y \leftarrow x$. An *antichain* in P is a subset of elements such that no two of them are related. A *chain* in P is a subset of elements such that every two of them are related. A *k-family* in P is a subset of elements that contains no chain of size $k + 1$ [16]. A *k-cofamily* in P is a subset of elements that contains no antichain of size $k + 1$ [16]. We can have an integer weight $w(p)$ associated with each element p in P. For a subset Q of P, the weight of Q, denoted $w(Q)$, is defined to be the sum of the weights of the elements in Q. A *maximum weighted k-family* (k-cofamily) in P is a k-family (k-cofamily) whose weight is maximum. An important fundamental result on partially ordered sets is a theorem due to Dilworth [11].

Theorem 2.[11]: For a partially ordered set P, if the maximum size of antichains is n, then P can be partitioned into n disjoint chains.

III. The k-Layer Planar Subset Problem

In this section, first, we show that the k-PSP problem is NP-complete for any fixed $k \geq 2$, even when the routing region is a switchbox and each net is a two-terminal net. Then, we show that the k-PSP problem can be solved for crossing channels in polynomial time.

3.1. NP-Completeness Results

When the routing region is a switchbox and each net is a two-terminal net with weight one, we call the corresponding k-PSP problem the *restricted* k-PSP problem (i.e., we simply want to maximize the number of nets in a k-planar subset). To be more precise, we state the decision problem for the restricted k-PSP problem as follows.

Restricted k-PSP problem

Instance: A switchbox, a set of two-terminal nets, and an integer M.

Problem: Can at least M nets be chosen such that each net can be routed entirely in one of k-layers?

We shall show that the restricted k-PSP problem is NP-complete for any fixed $k \geq 2$, which implies that the general k-PSP problem is NP-complete for any fixed $k \geq 2$.

First, we introduce the notion of a circle graph. Let C be a set of chords in a circle. The corresponding *circle graph G(C)* is an undirected graph in which each vertex represents a chord, and two vertices are connected if and only if the corresponding chords intersect. (See Fig. 3.) A graph is *k-colorable* if each vertex in the graph can be assigned one of k colors such that no two adjacent vertices have the same color. The *k-colorable subgraph problem for circle graphs* is stated as follows.

k-colorable subgraph problem for circle graphs

Instance: A circle graph G, and an integer M.

Question: Does G have a vertex-induced subgraph H which has at least M vertices and is k-colorable?

The following lemma shows the connection between the k-colorable subgraph problem for circle graphs and the restricted k-PSP problem. (The relation between circle

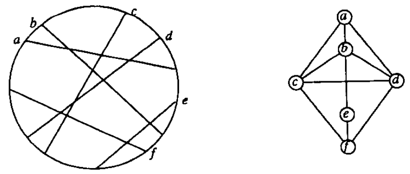


Fig. 3. (a) A set of chords. (b) The corresponding circle graph.

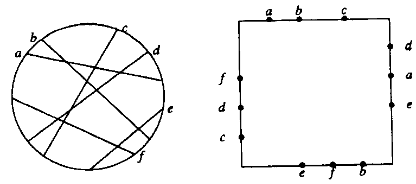


Fig. 4. Reduction from the restricted k-PSP problem to the k-colorable subgraph problem for circle graphs.

graphs and the two-layer planar routing problem for switchboxes was observed in [17].)

Lemma 1: The k-colorable subgraph problem for circle graphs can be reduced to the restricted k-PSP problem in polynomial time.

Proof: Let G be a circle graph and C the underlying set of chords. We construct an instance of the restricted k-PSP problem as follows: for each chord a in C, we introduce a two-terminal net a with weight one. Let N be the set of nets thus constructed. We assign the terminals in the nets in N to a switchbox such that the relative ordering of the terminals in the switchbox is the same as the relative ordering of the end points of all the chords in C. (See Fig. 4.) We claim that G has a k-colorable subgraph of size M if and only if N has a k-planar subset of weight M.

Suppose G has a k-colorable subgraph H of size M. We select a subset X on N as follows: for each vertex in H that is colored with color i, we include the corresponding net in X and route the net in the ith layer. Since for two vertices that have the same color their corresponding chords do not intersect, it is easy to see that the nets in X assigned to the same layer can be routed without crossing. Thus X is a k-planar subset with weight M. On the other hand, suppose X is a k-planar subset of N with weight M. For each net in X in the ith layer, we color the corresponding vertex in G with color i. Let H denote the subgraph spanned by the colored vertices in G. Clearly, the size of H is M. Moreover, it is easy to verify that the coloring thus obtained is a proper k-coloring on H. □

In the remainder of this section, we show that the k-colorable subgraph problem for circle graphs is NP-complete for k-≥ 2. First, based on the results in [31] and [26], we have the following lemma.

Lemma 2: The k-colorable subgraph problem for circle graphs is NP-complete for $k = 2$ and $k \geq 4$.

Proof: It was shown in [26] that the maximum 2-independent set problem for circle graphs is NP-complete. A *2-independent set* is the union of two independent sets in a graph. It is easy to see that a set of vertices in a graph form a 2-independent set if and only if these vertices induce a 2-colorable subgraph. Therefore, the 2-colorable subgraph problem for circle graphs is NP-complete.

It was shown in [31] that the problem of determining whether a circle graph is k-colorable is finding it NP-complete for $k \geq 4$. Clearly, the k-colorable problem for circle graphs can be reduced to the k-colorable subgraph problem for circle graphs simply by choosing M to be the number of the vertices in the given circle graph. Therefore, the k-colorable subgraph problem for circle graphs is NP-complete for $k \geq 4$. □

Here, we show that the k-colorable subgraph problem for circle graphs is also NP-complete for $k = 3$.

Lemma 3: The 3-colorable subgraph problem for circle graphs is NP-complete.

Proof: We reduce the 2-colorable subgraph problem for circle graphs to this problem. Let G be the circle graph and M be the integer in an instance of the 2-colorable subgraph problem. We construct the corresponding circle graph G' and determine the integer M' in an instance of the 3-colorable subgraph problem as follows: assume that G has n vertices. Corresponding to G, let C be the underlying circle and $p_1, p_2, \cdots, p_{2n}$ be the end points of the n chords on C in the clockwise direction. We choose a point s on arc $p_{2n}p_1$. (We always follow the clockwise direction along the circle in forming an arc). Also, we choose a point q_i on each arc p_ip_{i+1} for $1 \leq i \leq 2n - 1$. Then, we form $2n - 1$ chords $sq_1, sq_2, \cdots, sq_{2n-1}$. (See Fig. 5.) Furthermore, we make $n + 1$ copies of each chord sq_i $(1 \leq i \leq 2n - 1)$. We call these $(2n - 1)(n + 1)$ chords *forcing chords*. (For convenience, if two chords share one or two common end points, they are not considered to be intersecting, since we can always spread the common end points such that the two chords do not intersect.) Moreover, we introduce two more chords r_1r_3 and r_2r_4 such that r_1 and r_2 are on arc $p_{2n}s$ in the clockwise direction and r_3 and r_4 are on arc sp_1 in the clockwise direction. (Chords r_1r_3 and r_2r_4 are not shown in Fig. 5.) Clearly, r_1r_3 intersects r_2r_4. Furthermore, both r_1r_3 and r_2r_4 intersect every forcing chord but none of the original chords. We also make $n + 1$ copies of r_1r_3 and r_2r_4. Let G' be the circle graph corresponding to the n chords for G together with the $(2n - 1)(n + 1)$ forcing chords, $n + 1$ copies of r_1r_3 and $n + 1$ copies of r_2r_4. Let M' equal $(2n + 1)(n + 1) + M$. We claim that G has a 2-colorable subgraph of size at least M if and only if G' has a 3-colorable subgraph of size at least M'. (For convenience, in the remainder of this proof, we shall refer to a

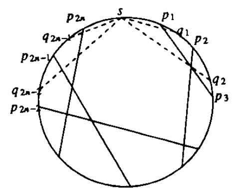


Fig. 5. Construction of circle graph G'.

chord and the corresponding vertex in the circle graph interchangeably. The color of a chord is the color of the corresponding vertex in the circle graph. Clearly, two intersecting chords cannot have the same color.)

Suppose that G has a 2-colorable subgraph H of size M. We construct a subgraph H' of G' as follows: we include in H' all the vertices in H and color them by color 1 or 2 according to their colors in H. Moreover, we include all the forcing chords in H' and color them by color 3. Furthermore, we include in H' every copy of $r_1 r_3$ and $r_2 r_4$. We color each copy of $r_1 r_3$ by color 1 and each copy of $r_2 r_4$ by color 2. Clearly H' is 3-colorable and the size of H' is $M + (2n + 1)(n + 1) = M'$.

On the other hand, suppose that G' has a 3-colorable subgraph H' of size M'. Clearly, H' contains at least one copy of each sq_i $(1 \le i \le 2n - 1)$ and at least one copy of both $r_1 r_3$ and $r_2 r_4$. Otherwise, the size of H' is smaller than $(2n + 1)(n + 1)$. Let H be the subgraph of G whose vertices appear in H' (i.e., H contains those chords in H' which are not forcing chords, $r_1 r_3$ or $r_2 r_4$). Clearly, the size of H is at least $M' - (2n + 1)(n + 1) = M$. Moreover, we can conclude that H is 2-colorable because of the following reasons: since $r_1 r_3$ intersects $r_2 r_4$, they must have different colors in H'. Without loss of generality, we assume that $r_1 r_3$ has color 1 and that $r_2 r_4$ has color 2. Then, all the forcing chords must have color 3 since each forcing chord intersects both $r_1 r_3$ and $r_2 r_4$. Thus each original chord in H can only have color 1 or 2 since each original chord intersects a forcing chord. Therefore, H is 2-colorable. □

Combining Lemmas 1, 2, and 3, we conclude the following theorem.

Theorem 3: The k-colorable subgraph problem for circle graphs is NP-complete for $k \ge 2$, which is equivalent to saying that the restricted k-PSP problem is NP-complete.

Clearly, this theorem also implies that the general k-PSP problem is NP-complete for $k \ge 2$. Note that the 1-colorable subgraph problem for circle graphs (i.e., to choose a maximum independent set in a circle graph) can be solved by a dynamic programming method in $O(n^2)$ time, where n is the number of vertices in the graph [28].

3.2. A Polynomial Time Algorithm for Crossing Channels

Let C be a crossing channel. We can assume that all the exits in C are moved to the lower or upper edge of C as shown in the proof of Theorem 1. Let $p_1, p_2, \cdots, p_s$ be the terminals on the lower edge of C from left to right. Let $q_1, q_2, \cdots, q_t$ be the terminals on the upper edge of C from left to right. For convenience, we may sometimes use the index of a terminal to refer the terminal when it is clear from the context. Since each net a is a crossing net, a has terminals on both the lower and upper edge of C. Let $x_1(a)$ and $x_2(a)$ be the minimum and the maximum indexes of the terminals of net a on the lower edge, respectively. Let $y_1(a)$ and $y_2(a)$ be the minimum and the maximum indexes of the terminals of net a on the upper edge, respectively. (Clearly, $x_1(a) \le x_2(a)$ and $y_1(a) \le y_2(a)$. We say that net a *dominates* net b (or net b is *dominated by* net a) if i) $x_1(a) \ge x_2(b)$ and $y_1(a) \ge y_2(b)$, or ii) $a = b$. (See Fig. 6.)

Lemma 4: All the nets in a crossing channel form a partially ordered set under the dominance relation.

Proof: We shall show that the dominance relation is reflexive, antisymmetric, and transitive.

1) For any net a, a dominates a by definition.

2) For any two nets a and b, suppose that a dominates b and b dominates a. If $a \ne b$, we have $x_1(a) \ne x_1(b)$ since two different nets cannot share a common terminal. Since a dominates b, $x_2(b) \le x_1(a)$. Moreover, note that $x_1(a) \le x_2(a)$ and $x_1(b) \le x_2(b)$, so we have

$$x_1(b) \le x_2(b) \le x_1(a) \le x_2(a).$$

However, since b also dominates a, we have $x_2(a) \le x_1(b)$. This implies that

$$x_1(b) = x_2(b) = x_1(a) = x_2(a).$$

This contradicts the fact that $x_1(a) \ne x_1(b)$. Therefore, we must have $a = b$.

3) For any three nets a, b, and c, suppose that a dominates b and b dominates c. Then, we have $x_2(b) \le x_1(a)$ and $x_2(c) \le x_1(b)$. Note that $x_1(b) \le x_2(b)$. Thus we have $x_2(c) \le x_1(a)$. Similarly, we have $y_2(c) \le y_1(a)$. Therefore, a dominates c. □

Given a crossing channel C, we use $P(C)$ to denote the partially ordered set formed by the nets in C under the dominance relation. Fig. 7 shows a crossing channel C and the corresponding partially ordered set $P(C)$. (We use the Hasse diagram to represent the partially ordered set in which the edges implied by transitivity are omitted). The advantage of introducing the dominance relation can be seen from the following two lemmas.

Lemma 5: In a crossing channel C, two nets can be routed entirely in the same layer without crossing if and only if one net dominates the other in $P(C)$.

Proof: For a net a, we join $x_1(a)$ and $y_1(a)$ by a line segment. Also, we join $x_1(a)$ and $y_2(a)$ by a line segment. These two line segments together with the two edges of the channel form a trapezoid, which is called the *bounding box* of net a. (See Fig. 8.) Note that the bounding box of a net may degenerate into a triangle or a line segment.

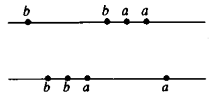
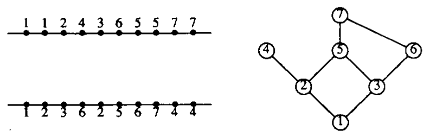


Fig. 6. Net a dominates net b.



(a) (b)

Fig. 7. (a) A crossing channel C. (b) The corresponding $P(C)$.

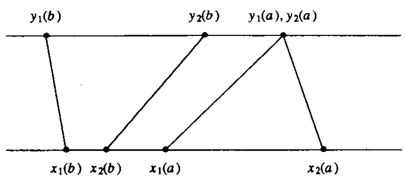


Fig. 8. Bounding boxes of nets.

It is not difficult to see that two nets can be routed entirely in the same layer without crossing if and only if their bounding boxes do not intersect. Moreover, the bounding boxes of two nets do not intersect if and only if one net dominates the other in $P(C)$. □

Lemma 6: A subset of nets M in a crossing channel C is a k-planar subset if and only if it is a k-cofamily in $P(C)$.

Proof: Suppose that M is a k-planar subset. We show that M does not contain an antichain A of size $k + 1$ in $P(C)$. Otherwise, since A is also a k-planar subset, there must be two nets in A which can be routed in the same layer without crossing (by the Pigeonhole Principle). However, according to Lemma 5, these two nets are related, which contradicts the fact that these two nets are in the antichain A. Therefore, M is a k-cofamily.

On the other hand, suppose that M is a k-cofamily in $P(C)$. Since the maximum size of antichains in M is at most k, according to Dilworth's Theorem, M can be partitioned into at most k chains. We assign the nets in the ith chain to the ith layer ($1 \leq i \leq k$). According to Lemma 5, the nets in each layer can be routed without crossing. Therefore, M is a k-planar subset. □

Based on Lemma 6 and the algorithm to be presented in Section V, we have the following theorem.

Theorem 4: The k-PSP problem for crossing channels can be solved in $O(n^2 \log n + nm)$ time, where n is the number of nets and m is bounded by n^2.

Proof: Given a crossing channel, we construct the partially ordered set $P(C)$ under the dominance relation. We assign the weight of an element in $P(C)$ to be the weight of the corresponding net in C. According to Lemma 6, finding a maximum weighted k-planar subset M in C is equivalent to finding a maximum weighted k-cofamily in $P(C)$. According to an algorithm to be presented in Section V, a maximum weighted k-cofamily of a partially ordered set P can be found in $O(n^2 \log n + nm)$, where n is the number of elements in P and m is the number of related pairs of nets in which one net dominates the other net. Clearly, m is upper bounded by $O(n^2)$. □

IV. The k-Layer Topological Via Minimization Problem

In this section, we first show that the k-TVM problem is NP-complete for any fixed $k \geq 2$, even when the routing region is a switchbox and each net is a two-terminal net. Then, we show that the k-TVM problem can be solved in polynomial time for crossing channels when each net is a two-terminal net.

4.1. *The NP-Completeness Results*

We show that the k-TVM problem is NP-complete even when the routing region is a switchbox and each net is a two-terminal net. First, it is easy to see that Lemma 1 in [21] can be generalized to k-layer routing. The generalization to k-layer routing for the unweighted cases was presented in [25]. The generalization to k-layer routing for the weighted cases can be stated as follows.

Lemma 7: Let N be a set of two-terminal nets. For any k-planar subset M of N, there is a topological routing solution in which each net in M uses no via and each net in $N - M$ uses at most one via.

Proof: We route each net in M entirely in one of the k-layers. We route the rest of the nets in $N - M$ in an arbitrary way to obtain a valid topological routing solution S. If net a in $N - M$ uses more than one via in S, we can always assign the two terminals in a two adjacent layers and use the construction procedure presented in [21] so that a is routed using only one via while the numbers of vias by other nets remains the same. We repeat such a rerouting procedure until each net in $N - M$ uses at most one via. □

Lemma 8: The k-TVM problem is equivalent to the k-PSP problem when each net is a two-terminal net.

Proof: Let N be a set of nets. For any k-planar subset M, according to Lemma 7, we can construct a topological routing solution S such that each net in M uses no via and each net in $N - M$ uses one via. The total via cost of S is

$$\sum_{a \in N} v(a) \cdot w(a) = \sum_{a \in M} 0 \cdot v(a) + \sum_{a \in N-M} 1 \cdot v(a)$$

$$= \sum_{a \in N-M} w(a) = \sum_{a \in N} w(a) - \sum_{a \in M} w(a).$$

Clearly, minimizing the total via cost of S is equivalent to maximizing the weight of the k-planar subset M. □

Since we have shown in the previous section that the restricted k-PSP problem is NP-complete, according to Lemma 8, we can easily conclude the following theorem.

Theorem 5: The k-TVM problem is NP-complete for $k \geq 2$ even when the routing region is a switchbox and each net is a two-terminal net.

4.2. A Polynomial Time Algorithm for Crossing Channels

Let C be a crossing channel in which each net is a two-terminal net. Let N denote the set of nets in C. We construct a topological routing solution S with the minimum total via cost as follows. First, find a maximum weighted k-planar subset M. We route each net in M in a single layer. Then, we route each net in $N - M$ in two adjacent layers according to the procedure in [21] such that only one via is used. Clearly, according to Lemma 7 and 8, the topological routing solution S thus constructed minimizes the total via cost. According to Theorem 4, the maximum weighted k-planar subset M can be computed in $O(n^2 \log n + nm)$ time. Moreover, the procedure in [21] can be implemented in $O(n^2)$ time to route all the nets in $N - M$, where n is the number of nets [26]. Thus we have the following theorem.

Theorem 6: If the routing region is a crossing channel and each net is a two-terminal net, the k-TVM problem can be solved in $O(n^2 \log n + nm)$ time, where n is the number of the nets, and m is bounded by n^2.

V. Computing a Maximum Weighted k-cofamily in a Partially Ordered Set

In this section, we present a strong polynomial time algorithm for computing a maximum weighted k-cofamily in a partially ordered set. Gavril [14] gave an $O(kn^2)$ time algorithm for computing a maximum k-cofamily in the unweighted case. His algorithm was applied in [25] to the topological via minimization problem for channels without local nets (equivalent to our crossing channels) for the unweighted case. However, Gavril's algorithm cannot be extended directly to obtain a polynomial time algorithm in the weighted case. In this section, we show that a maximum weighted k-cofamily can be computed in $O(n^2 \log n + mn)$ time for partially ordered sets with arbitrary positive weights. The key step in our approach is to reduce the problem of computing maximum weighted cofamilies in a partially ordered set to the problem of computing minimum cost flows in a network. (For basic concepts and terminologies in network flow, see [29]).

Let P be a partially ordered set with positive weights. Let $p_1, p_2, \cdots, p_n$ be the elements in P and $\leftarrow$ be the partial ordering relation. Let w_i denote the weight of p_i. First, we construct the *split graph* $G(P)$ associated with P as follows: for each element p_i in P, we introduce two vertices x_i and y_i in $G(P)$. We introduce a direct edge (x_i, y_j) in $G(P)$ if $p_j \leftarrow p_i$. Moreover, we introduce two more vertices s (source) and t (sink) in $G(P)$ and add edges (s, x_i) and (y_i, t) for each $1 \leq i \leq n$. Fig. 9 shows an example of a partially ordered set and its corresponding split graph. Furthermore, we choose the capacity of each edge e, denoted $c(e)$, to be 1 and the cost of each edge e, denoted $d(e)$, to be

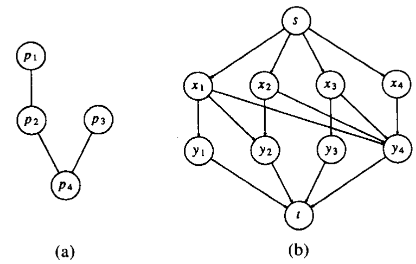


Fig. 9. (a) A partially ordered set P. (b) Its split graph $G(P)$.

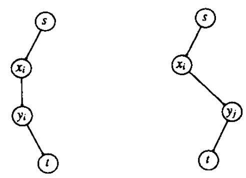
$$d(e) = \begin{cases} w_i, & \text{if } e = (x_i, y_i) \\ 0, & \text{otherwise.} \end{cases}$$

We shall show that maximum weighted cofamilies in P correspond to minimum cost flows in $G(P)$.

According to Dilworth's Theorem, any k-cofamily can be partitioned into no more than k chains. A k-cofamily is said to be *nontrivial* if it can be partitioned into exactly k chains. For a partially ordered set with positive weights, it is easy to see that any maximum weighted k-cofamily is a nontrivial k-cofamily. (Otherwise, we can increase the weight of the k-cofamily by including more elements in the k-cofamily.) The following theorem shows the connection between the nontrivial k-cofamilies in P and the $(n - k)$-flows in $G(P)$. (For convenience, we use f-flow to refer to a flow of value f from s to t in $G(P)$.)

Theorem 7: Let P be a partially ordered set of n elements with positive weights. Then, P has a nontrivial k-cofamily of weight D if and only if $G(P)$ has a $(n - k)$-flow of cost $W - D$, where W is the sum of the weights of all the elements in P. (Clearly, it is also equal to the sum of the costs of all the edges in $G(P)$.)

Proof: Suppose that P has a nontrivial k-cofamily M of weight D. We partition M into k disjoint chains $C_1, C_2, \cdots, C_k$. Let α denote the size of M and β_i denote the size of C_i $(1 \leq i \leq k)$. Clearly, $\alpha = \Sigma_{r=1}^{k} \beta_r$. We say that p_i *covers* p_j in some C_r $(1 \leq r \leq k)$ if $p_j \leftarrow p_i$ in C_r and there is no element p_l in C_r such that $p_j \leftarrow p_l \leftarrow p_i$. We construct a flow f in $G(P)$ as follows: for each element p_i in $P - M$, we assign $f(sx_i) = f(x_iy_i) = f(y_it) = 1$. (See Fig. 10(a).) We call such a unit flow a *paid flow* (since it goes through edges of nonzero cost). For each pair of element p_i and p_j in M, if p_i covers p_j in some chain C_r $(1 \leq r \leq k)$, we assign $f(sx_i) = f(x_iy_j) = f(y_jt) =$



(a) (b)

Fig. 10. Paid flow and free flow.

1. (See Fig. 10 (b).) We call such a unit flow a *free flow* (since it goes through only edges with zero cost). Since $P - M$, C_1, C_2, $\cdots$, C_k are disjoint, it is easy to see that the flow f thus constructed satisfies the capacity constraints and the flow conservation property. Clearly, the total value of the paid flows is $|P - M| = n - \alpha$. Note that each chain C_r has exactly $\beta_r - 1$ pairs of elements such that one element covers the other. Thus the total value of the free flows is

$$\sum_{r=1}^{k} (\beta_r - 1) = \sum_{r=1}^{k} \beta_r - k = \alpha - k.$$

Therefore, the total value of the flow f is $(n - \alpha) + (\alpha - k) = n - k$. Moreover, the total cost of the flow f equals the total cost of the paid flows, which is

$$\sum_{p_i \in P - M} w_i = \sum_{p_i \in P} w_i - \sum_{p_i \in M} w_i = W - D.$$

On the other hand, suppose that $G(P)$ has a $(n - k)$-flow f of cost $W - D$. Since the capacity of each edge in $G(P)$ is one, and each intermediate vertex has either one incoming edge or one outgoing edge, f can be partitioned into $n - k$ vertex-disjoint unit flows (except common end s and t). Again, for a unit flow of the form $f(sx_i) = f(x_i y_i) = f(y_i t) = 1$, we call it a paid flow. For a unit flow of the form $f(sx_i) = f(x_i y_j) = f(y_j t) = 1 (i \neq j)$, we call it a free flow. Let $\overline{M}$ be the set of elements p_i in P such that (x_i, y_i) is used by the paid flows. Let $\overline{\alpha}$ be the size of $\overline{M}$. We claim that $P - \overline{M}$ is a nontrivial k-cofamily of weight D. In fact, we can partition $P - \overline{M}$ into k disjoint chains as follows: first, we start with $n - \overline{\alpha}$ disjoint chains in $P - \overline{M}$, each chain containing a single element in $P - \overline{M}$. For each edge (x_i, y_j) $(i \neq j)$ used by the free flows, we join the two chains containing p_i and p_j and make p_i cover p_j in the resulting chain. Note that it is impossible to have both (x_i, y_{j_1}) and (x_i, y_{j_2}) (or, both (x_{i_1}, y_i) and $x_{i_2}, y_j)$) used in the free flows because all the unit flows are vertex-disjoint. Thus for any edge (x_i, y_j) $(i \neq j)$ in the free flows, before we process the edge (x_i, y_j), p_i and p_j always belong to two different chains. After we process the edge (x_i, y_j), we join these two chains, which reduces the total number of chains in $P - \overline{M}$ by one. Since there are $(n - k) - \overline{\alpha}$ edges of the form (x_i, y_j) $(i \neq j)$ used in free flows and each edge reduces the total number of chains of $P - \overline{M}$ by one, at the end, we have $(n - \overline{\alpha}) - (n - k - \overline{\alpha}) = k$ chains in $P - \overline{M}$. Moreover, the weight of $P - \overline{M}$ is

$$\sum_{p_i \in P - \overline{M}} w_i = \sum_{p_i \in P} w_i - \sum_{p_i \in \overline{M}} w_i = W - (W - D) = D.$$

Therefore, $P - \overline{M}$ is a nontrivial k-cofamily with weight D. □

Since every maximum weighted k-cofamily is a nontrivial k-cofamily, according to Theorem 7, we conclude that the problem of computing a maximum weighted k-cofamily is equivalent to the problem of computing a minimum cost $(n - k)$-flow in $G(P)$. Moreover, we can partition the maximum weighted k-cofamily thus obtained into k disjoint chains according to the proof of Theorem 7. In solving the k-PSP problem for crossing channels, such a partition gives us the layer assignment of the nets in the maximum weighted k-planar subset.

We now show how to determine a minimum cost $(n - k)$-flow in $G(P)$. We recall the result that *any flow obtained from a minimum cost flow by augmenting along an augmenting path of minimum cost is also a minimum cost flow* ([29] Theorem 8.12). A minimum cost augmenting path can be found by finding a minimum cost path from s to t in the residual graph. Our algorithm works as follows. We start with a zero flow f in $G(P)$. Initially, the residual graph R is the same as $G(P)$. We find a minimum cost path from s to t in the residual graph R and augment the flow f in $G(P)$ by one (since each edge capacity is one). Next, we modify the costs of the edges in $G(P)$ as $d'(v, w) = d(v, w) + \text{cost}(v) - \text{cost}(w)$, where $\text{cost}(v)$ is the cost of a minimum cost path from s to v in the residual graph R (these values were computed already as we were constructing the minimum cost path from s to t). Then, we update the residual graph R. We repeat the augmenting process until the value of the flow f reaches $n - k$. It is easy to show that modifying the cost of the edges in $G(P)$ does not change the relative ordering of the augmenting paths (i.e., a minimum augmenting path still has the minimum cost among all the augmenting paths after we modify the edge costs). Moreover, such a redefinition of the costs of the edges guarantees that the costs of the edges in the residual graph R are always non-negative (for details, see [29]). Therefore, we can compute a minimum cost path from s to t in R more efficiently. (Otherwise, the residual graph may have edges with negative cost so that it takes longer time to compute a path of the minimum cost in it.) If we apply Dijkstra's shortest path algorithm, a minimum cost path from s to t in R can be computed easily in $O(n^2)$ time. However, if we are willing to use a more complicated data structure (Fibonacci Heaps), a minimum cost path in R can be computed in $O(n \log n + m)$ time [13], where m is the number of edges in R. Obviously, our algorithm goes through $n - k$ augmenting steps since the capacity of each edge in $G(P)$ is one. Therefore, the complexity for computing a minimum cost $(n - k)$-flow in $G(P)$ is $O((n - k)(n \log n + m)) = O(n^2 \log n + mn)$. Based on these discussions, we have Theorem 8.

Theorem 8: For a partially ordered set of n elements with positive weights, a maximum weighted k-cofamily can be computed in $O((n - k)(n \log n + m)) = O(n^2 \log n + mn)$ time.

We note that the network used in [14] can be modified to compute a maximum weighted k-cofamily by introducing appropriate costs on the edges. In this case, a maximum weighted k-cofamily corresponds to a minimum cost k-flow in the corresponding network (instead of a minimum cost $(n - k)$-flow as in our case). Such an approach leads to a $O(k(n \log n + m))$ time algorithm for computing a maximum weighted k-cofamily in a partially ordered set. Both this algorithm and the algorithm presented in this section run in $O(n^2 \log n + mn)$ time in the worst case (when $k \approx 1/2\ n$). We choose to use our network formulation in this section because the same network can also be used to compute a maximum weighted k-family in a partially ordered set (recall that a k-family is a union of k antichains). Indeed, the same network formulation for computing a maximum weighted k-family was used in an earlier work on over-the-cell routing in standard cell design [9].

VI. Conclusions

In this paper, we studied two closely related problems—the k-layer planar subset problem and the k-layer topological via minimization problem. Both problems are important for performance-driven layout design. We showed that the k-PSP problem and the k-TVM problem are both *NP*-complete in general. When the routing region is a crossing channel, we gave polynomial time algorithms to compute optimal solutions of both problems. In particular, we formulate the k-PSP problem and the k-TVM problem for crossing channels as that of finding a maximum weighted k-cofamily in a partially ordered set and we show that a maximum weighted k-cofamily can be computed in strong polynomial time for any partially ordered set with positive weights. We believe that these results will be also useful in solving other CAD problems.

It was called to our attention that results similar to those reported in this paper were obtained independently and reported in [27].

Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments. In particular, the proof of Lemma 3 is simplified based on the suggestion of one of the reviewers.

References

[1] D. Braun *et al.*, "Chameleon: A new multi-layer channel router," in *Proc. 23rd Design Automation Conf.*, 1986, pp. 495-502.

[2] P. Bruell and P. Sun, "A 'greedy' three layer channel router," in *Proc. ICCAD-85*, 1985, pp. 298-300.

[3] H. Cai, "Connectivity biased channel construction and ordering for building-block layout," in *Proc. ACM/IEEE 25th Design Automation Conf.*, 1988, pp. 560-565.

[4] Y. K. Chen and M. L. Liu, "Three layer channel routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-3, pp. 156-163, 1984.

[5] C. K. Cheng and D. N. Deutsch, "Improved channel routing by via minimization and shifting," in *Proc. 25th ACM/IEEE Design Automation Conf.*, 1988, pp. 677-680.

[6] J. Cong and D. F. Wong, "How to generate more compactable channel routing solutions," in *Proc. IEEE/ACM 25th Design Automation Conf.*, June 1988, pp. 663-666.

[7] 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.

[8] J. Cong and C. L. Liu, "On k-layer planar subset and via minimization problems," in *Proc. European Design Automation Conf.*, Mar. 1990, pp. 459-463.

[9] J. Cong, B. Preas, and C. L. Liu, "General models and algorithms for over-the-cell routing in standard cell design," in *Proc. ACM/IEEE 27th Design Automation Conf.*, 1990, pp. 709-715.

[10] W. M. Dai, T. Assno, and E. S. Kuh, "Routing region definition and ordering scheme for building block layout," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 189-197, July 1985.

[11] R. P. Dilworth, "A decomposition theorem for partially ordered set," *Ann. Math*, vol. 51, pp. 161-166, 1950.

[12] R. J. Enbody and H. C. Du, "Near-optical n-layer channel routing," in *Proc. 23th Design Automation Conf.*, 1986, pp. 708-714.

[13] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *J. Assoc. Comput. Mach.*, vol. 34, no. 3, pp. 596-615, 1986.

[14] F. Gavril, "Algorithms for maximum k-coloring and k-covering of transitive graphs," *Networks*, vol. 17, pp. 465-470, 1987.

[15] 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.

[16] C. Greene and D. Kleitman, "The structure of Sperner k-family," *J. Combinatorial Theory, Ser. A*, vol. 20, pp. 80-88, 1976.

[17] C. P. Hsu, "Minimum via topological routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 235-246, 1983.

[18] K. F. Liao, D. T. Lee, and M. Sarrafzadeh, "Planar subset of multi-terminal nets," Tech. Rep. Dept. Elect. Eng. Comput. Sci., Northwestern Univ., Evanston, IL, 1988.

[19] R. D. Lou, M. Sarrafzadeh, and D. T. Lee, "An optimal algorithm for the maximum two-chain problem," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, Jan. 1990.

[20] C. L. Liu, *Elements of Discrete Mathematics.* New York: McGraw-Hill, 1977.

[21] M. Marek-Sadowska, "An unconstrained topological via minimization problem for two-layer routing," *IEEE Trans. Computer-Aided Design*, vol. CAD-3, pp. 184-190, 1984.

[22] B. Preas, "Placement and routing algorithms for hierarchical integrated circuit layout," Ph.D. dissertation, Stanford Univ. 1979.

[23] ——, "General cell layout design system," Xerox PARC Internal Documents, 1987.

[24] B. Preas, and M. Lorenzetti, eds., *Physical Design Automation of VLSI Systems.* Menlo Park, CA: Benjamin/Cummings, 1988.

[25] C. S. Rim, T. Kashiwabara, and K. Nakajima, "Exact algorithms for multilayer topological via minimization," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 1165-1173, 1989.

[26] M. Sarrafzadeh and D. T. Lee, "A new approach to topological via minimization," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 890-900, 1989.

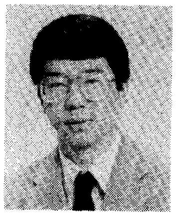
[27] M. Sarrafzadeh and R. D. Lou, "Maximum k-coverings in transitive graphs," in *IEEE Proc. Int. Symp. on Circuits and Systems*, May 1990.

[28] K. J. Supowit, "Finding a maximum planar subset of a set of nets in a channel," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 93-94, Jan. 1987.

[29] R. E. Tarjan, *Data Structures and Network Algorithms.* Philadelphia, PA: Society for Industrial and Applied Mathematics, 1983.

[30] K. S. The, D. F. Wong, and J. Cong, "Via minimization by layout modification," in *Proc. 26th ACM/IEEE Design Automation Conf.*, 1989, pp. 799-802.

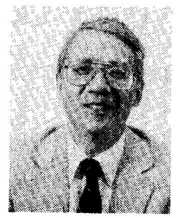
[31] W. Unger, "On the k-coloring of circle graphs," in *Proc. 5th Ann. Symp. on Theoretical Aspects of Computer Science*, 1988, pp. 61-72.



Jingsheng Cong (M'91) received the B.S. degree in computer science from Peking University in 1985, the M.S. and Ph.D. degrees in computer science from University of Illinois at Urbana-Champaign in 1987 and 1990, respectively.

Currently, he is an Assistant Professor in the Computer Science Department of University of California at Los Angeles. From 1986 to 1990, he was a research assistant in the Computer Science Department of University of Illinois. He worked at Xerox Palo Alto Research Center in the summer of 1987. He worked at National Semiconductor Corporation in the summer of 1988. His research interests include design and analysis of efficient combinational and geometric algorithms, computer-aided design of VLSI circuits, and fault-tolerant design of VLSI systems.

Dr. Cong received the Best Graduate Award from Peking University in 1985. He received the Ross J. Martin Award for excellence in research from University of Illinois at Urbana-Champaign in 1989. He received the Student Research Paper Award from Sigma Xi, the scientific research society in University of Illinois at Urbana-Champaign in 1990. He is a member of ACM.



C. L. Liu (M'78–SM'82–F'86) received the B.Sc. degree from the National Cheng Kung University in Taiwan in 1956, and the M.S. and E.E. degrees in 1960 and the Sc.D. degree in 1962, all in electrical engineering from the Massachusetts Institute of Technology.

He is currently a Professor of Computer Science at the University of Illinois at Urbana-Champaign. He serves currently on the editorial board of the journals: IEEE TRANSACTIONS ON COMPUTERS, *Graphs and Combinatorics*, and *Algorithmica*, and is also an editorial advisor of the Computer Science Series published by the Wood Scientific Publishing Company. His areas of research interest are design and analysis of algorithms, computer-aided design of integrated circuits, and combinatorial mathematics.