

Optimal River Routing with Crosstalk Constraints

HAI ZHOU and D. F. WONG

University of Texas at Austin

With the increasing density of VLSI circuits, the interconnection wires are being packed even closer. This has increased the effect of interaction among these wires on circuit performance and hence, the importance of controlling crosstalk. In this article, we consider river routing with crosstalk constraints. Given the positions of the pins in a single-layer routing channel and the maximum tolerable crosstalk between each pair of neighboring nets, we give a polynomial time algorithm to decide whether there is a feasible river routing solution and produce one with minimum crosstalk when it is feasible.

Categories and Subject Descriptors: B.7.2 [**Integrated Circuits**]: Design Aids—*placement and routing*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*network problems*; J.6 [**Computer-Aided Engineering**]: *computer-aided design*

General Terms: Algorithms, Design, Experimentation, Performance

Additional Key Words and Phrases: Crosstalk, river routing

1. INTRODUCTION

With VLSI fabrication entering the deep submicron era, devices and interconnection wires are being placed at an ever-increasing proximity. Reduction in the interconnection and transistor switching delays results in faster signal transition times. All these factors increase the coupling effect (inductive and capacitive) between interconnection wires. Increased coupling effect not only increases signal delays, but also decreases signal integrity due to transmission line behavior. This phenomenon is called *crosstalk* [Bakoglu 1990].

In the literature, previous works on crosstalk avoidance in detailed routing fall into two main categories: the gridless routing model is used and spacings between wires are adjusted to reduce crosstalk [Chen and Wong 1992; Chaudhary et al. 1993]; or the gridded routing model is used with works in this category focusing on channel routing [Gao and Liu 1993;

Authors' address: Department of Computer Sciences, University of Texas at Austin, Austin, TX 78712-1188; email: (haizhou@cs.utexas.edu).

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

© 1999 ACM 1084-4309/99/0700-0496 \$5.00

Kirkpatrick and Sangiovanni-Vincentelli 1994; Thakur et al. 1995] and switchbox routing [Gao and Liu 1994]. The general approach taken in most previous work belongs to postrouting optimization. That is, a routing solution is first determined by a conventional routing algorithm, then algorithms are designed to modify the routing solution to reduce crosstalk.

In this article, we consider the river routing problem with crosstalk constraints. Although river routing has the best developed theory among all detailed routing problems [Lengauer 1990], there is no previous work with consideration of crosstalk avoidance. Contrary to Chen and Wong [1992], Gao and Liu [1993, 1994], Chaudhury et al. [1993], and Thakur et al. [1995], we consider crosstalk among wires in a global way during the routing process and develop a novel routing algorithm. Given the pin positions and the maximum tolerable crosstalk between each pair of neighboring nets, for any fixed channel width, the algorithm can decide whether there is a feasible routing solution. Furthermore, if there is one, the algorithm can give a routing solution with minimum crosstalk at the same time complexity.

The rest of the article is organized as follows. Section 2 formally defines the crosstalk-constrained river routing problem. The optimal algorithm is presented in Section 3. In Section 4 we present a postprocessing heuristic to reduce the number of bends in the solution. Finally, Section 5 concludes the article with experimental results and some remarks.

2. PROBLEM FORMULATION

In a river routing problem, we are given a single-layer rectangular routing channel with pins located at the top and the bottom of the channel. Let t_1, t_2, \dots, t_n be the pins at the top boundary of the channel (from left to right) and b_1, b_2, \dots, b_n be the pins at the bottom boundary of the channel (from left to right). We assume that there is a grid superimposed over the channel and that all pins are on grid points. For $1 \leq i \leq n$, net i is defined by (t_i, b_i) . A route of net i is a simple path in the grid graph connecting t_i and b_i . We also assume all routes to be *monotonic*; that is, starting from one pin, a route can only proceed in at most two directions. All previous works on river routing used this assumption, since it does not hurt routability [Lengauer 1990], and at the same time, makes each route the shortest one. A *routing solution* is a set of routes, one for each net, such that no route shares a grid point. A route is *legal* if it exists in at least one routing solution.

Generally speaking, crosstalk between two parallel straight wires is proportional to the coupling capacitance between them, which in turn is proportional to their coupling length and inversely proportional to their separating distance. Since the coupling capacitance between two wires decreases rapidly as the distance between them increases, it is reasonable to assume that crosstalk only exists between wires in adjacent rows or columns. Without loss of generality, we can set the proportion constant to 1 and treat the adjacent length between two wires as their crosstalk.

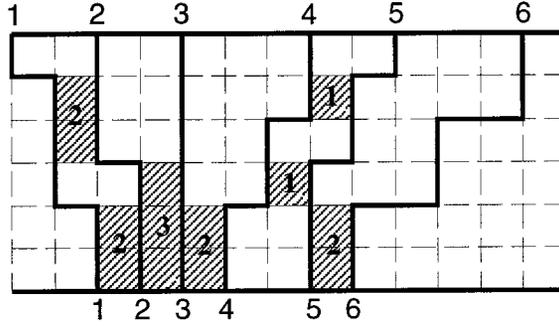


Fig. 1. River routing solution and crosstalk.

Crosstalk between two routes is then the total number of their adjacent grid edges. Note that this is also the crosstalk model used in Gao and Liu [1993, 1994]. In any routing solution, let R_i denote the crosstalk between net i and net $i + 1$, for $1 \leq i \leq n - 1$. For example, Figure 1 shows a river routing solution and crosstalk between each pair of routes. Here, $R_1 = 2 + 2 = 4$, $R_2 = 3$, $R_3 = 2$, $R_4 = 1 + 1 = 2$, and $R_5 = 2$.

For each pair of neighboring nets i and $i + 1$, where $1 \leq i \leq n - 1$, a constant C_i is given as the maximum tolerable crosstalk between them. A routing solution such that $R_i \leq C_i$, for $1 \leq i \leq n - 1$, is called a *feasible* routing solution. A route is feasible if it exists in at least one feasible routing solution. We also call $C_i - R_i$, for all $1 \leq i \leq n - 1$, crosstalk *slacks* and define *min-slack* to be $\min_{1 \leq i \leq n-1} (C_i - R_i)$. The problem we address in this article can be stated as follows.

Crosstalk-Constrained River Routing (CCRR) Problem. Given a channel of width W , a set of n nets $(t_1, b_1), (t_2, b_2), \dots, (t_n, b_n)$, and a set of crosstalk constraints C_1, C_2, \dots, C_{n-1} , determine whether there is a feasible routing solution. If there is one, give a routing solution with maximum min-slack.

3. OPTIMAL ALGORITHM

In a river routing problem, since only one layer is used, the relative positions of wires are fixed. Therefore, the only way to reduce the crosstalk between two neighboring nets is to position them a little farther away. This can be regarded as inserting space between them. Because of the fixed pin positions and monotonic route requirement, the space that can be used is limited and must be allocated properly to fulfill the crosstalk constraints. In this sense, the CCRR problem might be transformed to and solved by a space allocation problem. But the transformation is not trivial because there is no one-to-one correspondence between space insertion and crosstalk reduction. There are two problems: crosstalk can be reduced without space insertion and space insertion sometimes cannot reduce crosstalk. We explain them by two simple examples. In Figure 2(a), two nets are routed as close to each other as possible, and hence have the same

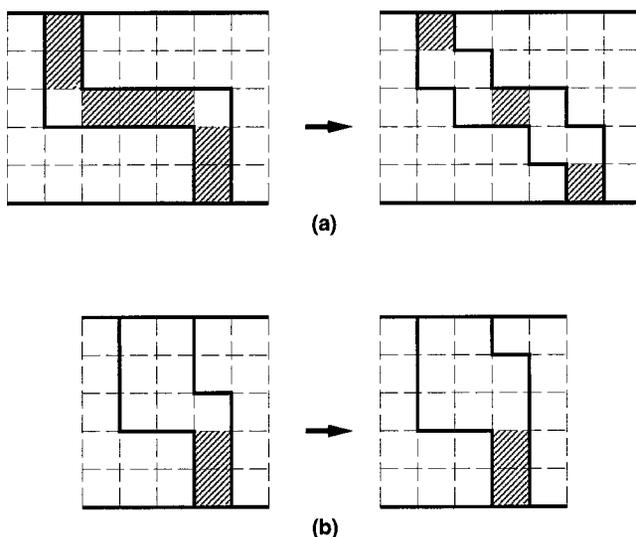


Fig. 2. (a) Crosstalk reduction without space insertion; (b) space insertion without crosstalk reduction.

space between them. Crosstalk is decreased when they change their shapes. In Figure 2(b), one more unit of space is inserted between two nets, but crosstalk between them does not decrease.

Analyzing the situation in Figure 2(a), we can find that crosstalk is decreased because more bends are introduced. The basic reason for this is that at the concave side of a bend the nearest wire is at least two units apart. Since bends are beneficial for crosstalk avoidance, we are interested in those routes with as many bends as possible. For each net, such routes are defined as *MC* (stands for minimal-crosstalk) routes, inasmuch as they will have minimal crosstalk if neighboring nets are routed closely along them. It is shown that, if there is a feasible routing solution, then there will also be one composed of MC routes. That is, considering only MC routes is sufficient to find an optimal solution for the CRR problem. In addition, MC routes have many good properties. One of them is that space must be inserted in order to reduce crosstalk between two MC routes. This avoids the problem in Figure 2(a) and helps in modeling the CRR problem by space allocation. Details about MC routes are presented in Section 3.1.

In order to measure the crosstalk reduction requirement for each pair of nets, we need to compute the maximum crosstalk between their MC routes. We define this crosstalk as *potential crosstalk* of the two nets, which is the crosstalk when no space is inserted. Obviously, the difference between potential crosstalk and crosstalk constraint defines the requirement. Based on the properties of MC routes, the same amount of space is needed to get a feasible solution. The space that can be used to reduce potential crosstalk is defined as *MC space*. It may be decomposed into subregions called *MC blocks*, each of which is shared by a set of net pairs. Furthermore, if we restrict the maximum amount of MC space allocated to each pair of nets,

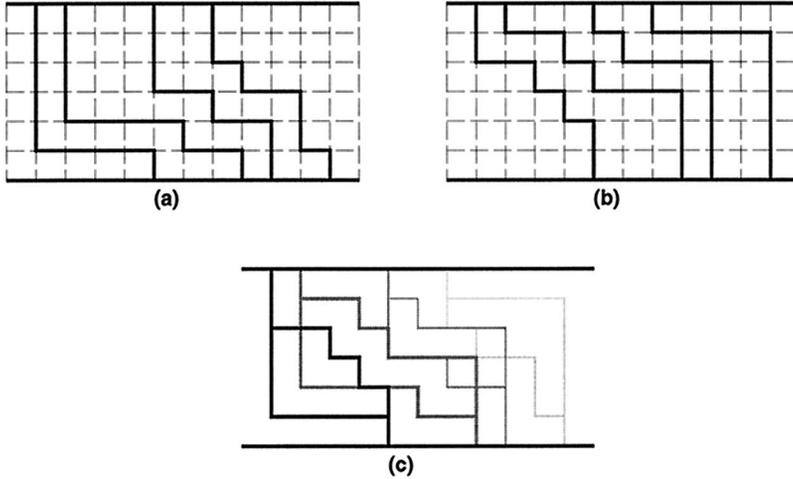


Fig. 3. (a) Leftmost routes; (b) rightmost routes; (c) legal boundaries.

we can also avoid the problem shown in Figure 2(b). Under this restriction, we establish a one-to-one correspondence between MC space insertion and crosstalk reduction. Therefore, the CRR problem is transformed into a space allocation problem. It can then be modeled by a flow network and solved by a maximum flow computation. Details about the space allocation and network flow are presented in Sections 3.2 and 3.3, respectively.

3.1 MC (Minimal-Crosstalk) Routes

Without considering crosstalk, river routing is a well-solved problem. Given a river routing problem, there exists an $O(n^2)$ time algorithm that decides whether there is a routing solution and gives one if possible [Dolev et al. 1981; Lengauer 1990; Lloyd and Ravi 1984]. In the CRR problem, among all those possible solutions, one with crosstalk under constraints must be selected. In order to specify the flexibility of the selection, we are interested in two extreme routing solutions: all nets are routed as far to the left as possible, as shown in Figure 3(a), and all nets are routed as far to the right as possible, as shown in Figure 3(b). For each net, we call these routes left and right *legal boundaries*, respectively (Figure 3(c)). It is easy to see that a route is legal if and only if it is within the legal boundaries.

As observed from Figure 2(a), bends are useful in reducing crosstalk. Therefore, we prefer to perform routing using as many bends as possible.

Definition 1 (Stair). A *stair* is a route segment composed of alternating horizontal and vertical grid edges.

To illustrate the definition, a route is shown in Figure 4(a), where the encircled segments are stairs. It is easy to see that except the first and last edges, which may induce crosstalk on one side, the middle part of a stair cannot have crosstalk with any wire. In order to simplify our presentation, in the rest of the article, we use a 45° line to represent the middle part of a

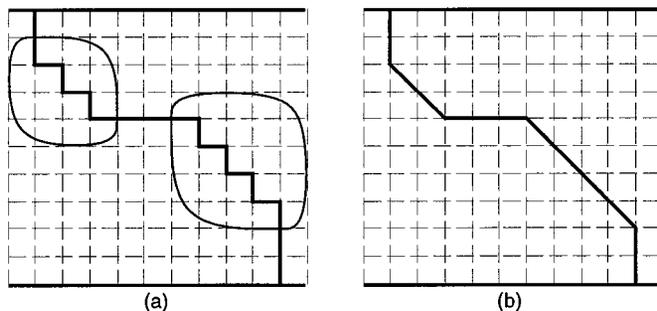


Fig. 4. Stairs and their representations.

stair, which can never have crosstalk. In addition, we call other segments straight lines, or specifically horizontal/vertical lines when the orientations are emphasized. For example, Figure 4(b) is used to represent Figure 4(a).

Based on stairs and straight lines, for each net, we can define a binary relation on the set of its legal routes.

Definition 2 (Derivation). Let w and w' be two legal routes of a given net; we say w' is derived from w if its straight lines are all on w .

It is easy to check that the derivation relation is reflexive, transitive, and antisymmetric, and hence a partial order. Based on the definition, in Figure 5, the route in (b) is derived from that in (a). Dashed lines in the figure are legal boundaries, drawn to show that both routes are legal.

Since only straight lines can induce crosstalk, if w' is derived from w , it usually will not induce more crosstalk than w . Formally, this is stated as the following lemma.

LEMMA 1. *Suppose w' is a legal route derived from route w . If w is feasible, then w' is also feasible.*

PROOF. Consider a feasible routing solution that includes route w . Substitute w with w' . According to the definition of derivation, only stairs of w' may get crossed with the neighboring route. If this happens, we also change the crossed parts of the neighboring route and route them as close as possible to w' . It is not difficult to see that the changed parts are all stairs with the first and last edges on the original route. Continue the process until there is no crossing between routes. This is always possible, because w' is legal and we always route the changed parts as close to each other as possible. Because all new parts are middle parts of stairs, crosstalk cannot increase (usually, it decreases). \square

The preceding lemma shows that if w' is derived from w , considering only w' is sufficient to find a feasible solution. Therefore, we are interested in a set of special routes defined as follows.

Definition 3 (MC Route). A legal route is called an *MC (minimal-crosstalk) route* if its only derived route is itself.

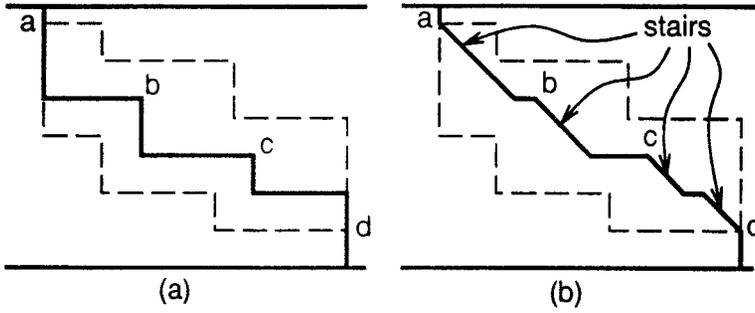


Fig. 5. (a) Legal route; (b) derived MC route.

We can check that the route in Figure 5(b) is actually an MC route. The following theorem shows that considering only MC routes is sufficient in the CCRR problem.

THEOREM 1. *If there is a routing solution in which crosstalk between net i and net $i + 1$ is R_i , for $1 \leq i \leq n - 1$, then there will be a routing solution composed of MC routes in which crosstalk between net i and net $i + 1$ is R'_i such that $R'_i \leq R_i$, for $1 \leq i \leq n - 1$.*

PROOF. Suppose r is the given routing solution; if all of its routes are MC routes, we already have the result. Otherwise, there must be a route w that is not an MC route. By the process in the proof of Lemma 1, we can change w to an MC route without increasing crosstalk. That process will not change other MC routes to non-MC ones because every new part is a stair. Repeating the process for each non-MC route, we will finally get a routing solution composed of all MC routes. \square

Similar to legal routes, MC routes of a given net are also restricted within some boundaries.

Definition 4 (MC Boundary and MC Region). For each net, the MC route derived from the left legal boundary is called the *left MC boundary*; the MC route derived from the right legal boundary is called the *right MC boundary*. The regions surrounded by the left and right MC boundaries are called *MC regions*.

The MC boundaries and MC regions of one net are shown in Figure 6. The dashed lines are the left and right legal boundaries of the net.

We summarize the properties of MC routes in the following lemma.

LEMMA 2. *For each net, we have*

- (1) *all MC routes are within the MC boundaries;*
- (2) *every MC route has the same length of horizontal lines and the same length of vertical lines; and*
- (3) *the horizontal and vertical lines of a MC route, if both exist, must lie within separate MC regions.*

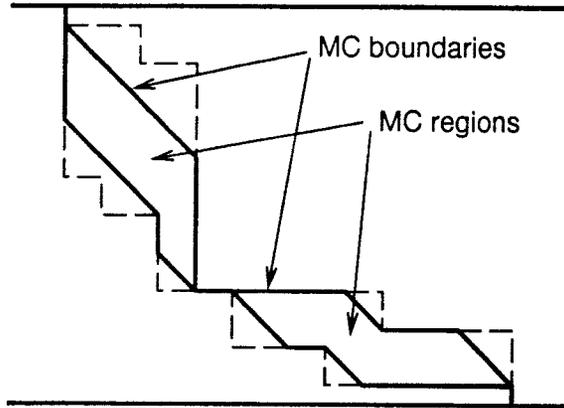


Fig. 6. MC boundaries and MC regions.

PROOF. Suppose Property 1 is not true; that is, there is an MC route w that goes outside, say, the left MC boundary. According to the definition of an MC route, w must be within the left legal boundary. Let s denote a segment lying between the legal boundary and the MC boundary. It is easy to see that s can be changed to a segment on the MC boundary, which is always a stair. This is a contradiction to the fact that w is an MC route.

For Properties 2 and 3, first we show that on the left MC boundary, which is also an MC route, horizontal and vertical lines cannot be boundaries of the same MC region.

Suppose there are one horizontal and one vertical line on the boundary of a region. Without loss of generality, we may assume that the top pin is at the left of the bottom pin. Because of the monotonicity requirement, the two straight lines have only two relative positions, shown in Figure 7. In Figure 7(a), the vertical line is to the upper left of the horizontal line. The left MC boundary from a to b must meet the right boundary. Otherwise, we can “push” the stair up and right and get a longer stair. In Figure 7(b), the horizontal line is to the upper left of the vertical line. The stairs starting from a and b will both meet the right boundary. Otherwise, we can substitute the straight line by the corresponding stair. This, then, means the right MC boundary meets some point between a and b . In both positions, horizontal and vertical lines cannot lie on the boundary of the same region.

For any other MC route, we only need to prove that it has the same length of horizontal and vertical lines as the left MC boundary in any separate MC region. This can be easily shown by comparing the coordinates of the two points where the left MC boundary meets the right MC boundary. Every route must go through these two points, and if their horizontal distance is larger (smaller) than the vertical distance, then every MC route must have horizontal (vertical) lines whose length will be the difference of the two distances. \square

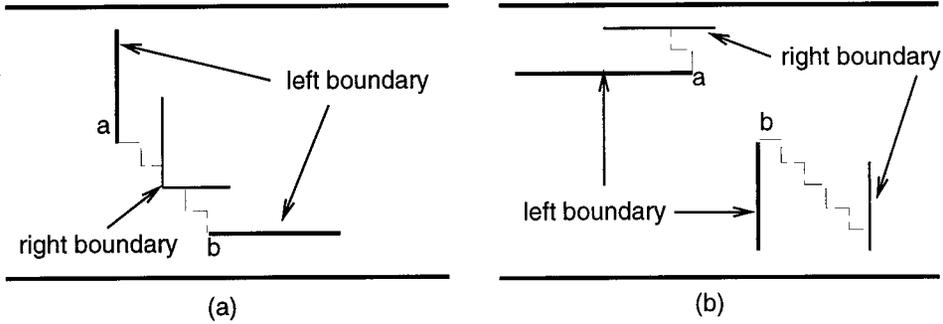


Fig. 7. Relative positions of straight lines.

3.2 Space Allocation

To make the presentation clear, in the previous section we discuss each net independently as though other nets do not exist. In this section, we consider the interaction among different nets. Theorem 1 allows us to focus only on MC routes.

We first consider the worst-case crosstalk between two neighboring nets.

Definition 5 (Potential Crosstalk). The maximum crosstalk between any MC routes of net i and net $i + 1$ is called their *potential crosstalk* and is denoted by p_i .

Since only straight lines can induce crosstalk, the potential crosstalk happens when two MC routes have their straight lines as close to each other as possible. In fact, a general way to compute potential crosstalk p_i is to route net i along its right MC boundary and route net $i + 1$ as close to it as possible. For example, given two neighboring nets with their MC boundaries as shown in Figure 8(a), Figure 8(b) shows the general way to compute the potential crosstalk; Figure 8(c) gives another situation when potential crosstalk happens.

If $p_i \leq C_i$, net i and net $i + 1$ can always be routed close to each other without crosstalk violation. But if $p_i > C_i$, we cannot route net i and net $i + 1$ closely. The reason is that we are now using MC routes and, according to Lemma 2, changing shape cannot reduce the length of straight lines, and hence closely routed MC routes of net i and net $i + 1$ will always have crosstalk p_i . Comparing Figures 8(b) and 8(c) illustrates this conclusion. Therefore, to reduce the crosstalk between two MC routes, space must be inserted. For example, to reduce crosstalk in Figure 8(c), as shown in Figure 9(a), we can change straight line ab to $a'b'$ or straight line cd to $c'd'$, which is equivalent to inserting space between ab and $a'b'$, or between cd and $c'd'$. It must also be noticed that, for each space insertion, there is a limit on the maximum achievable crosstalk reduction. Beyond that limit, inserting space cannot reduce crosstalk. Such an example is shown in Figure 9(b).

Since each net must be routed in an MC route, the space that can be used to reduce crosstalk is restricted within its left and right MC boundaries

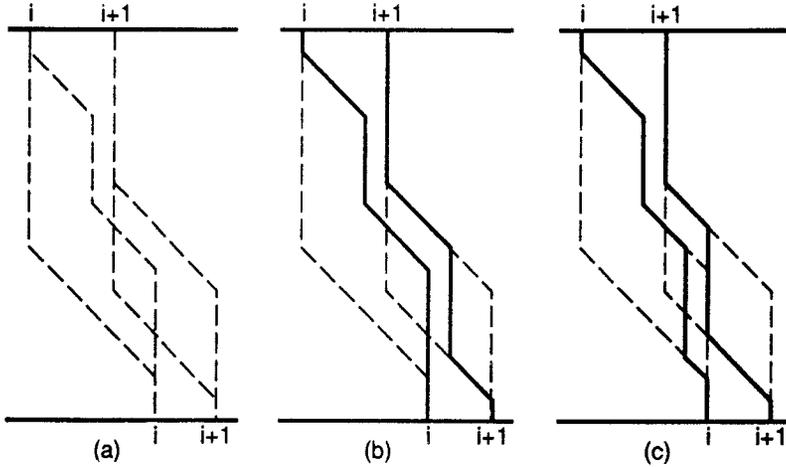


Fig. 8. Potential crosstalk.

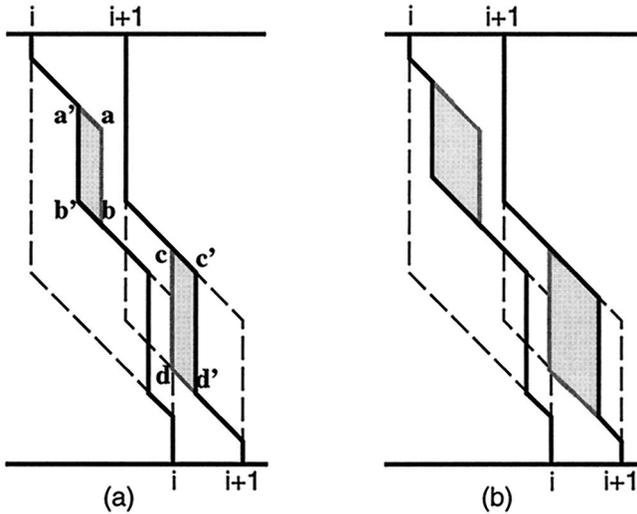


Fig. 9. Crosstalk reduction vs. space insertion.

(which are the left- and rightmost MC routes according to Lemma 2), that is, the space in the MC regions. But the space in each MC region cannot be counted independently. Different nets may share the space in their MC regions. For example, consider once again the two nets shown in Figure 8(a). When net i is routed along its right MC boundary, as shown in Figure 10(a), the shaded region in net $i + 1$ cannot be used. Similarly, as shown in Figure 10(b), if net $i + 1$ is routed along the left MC boundary, the shaded region of net i cannot be used either. In fact, the two shaded regions must be treated as the same block shared by the two nets.

Space sharing not only happens between neighboring nets, but may span a set of consecutive nets. Generally, *shared space* can be identified as

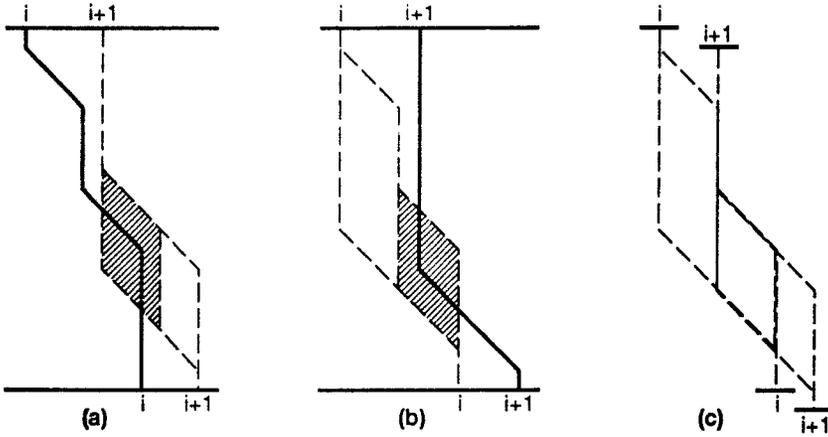


Fig. 10. Space sharing among nets.

follows. Here we call a net a *falling net* if its top pin is at the left of its bottom pin, or a *rising net* if its top pin is at the right of its bottom pin.

Definition 6 (Shared Space). Given net i and net j with $i < j$, we can move net j left and down both by $j - i$ units if it is a falling net; or move it left and up both by $j - i$ units if it is a rising net. The overlapped space in their MC regions then is their *shared space*.

For the two nets shown in Figure 8(a), we move net $i + 1$ left and down both by $(i + 1) - i = 1$ unit, as shown in Figure 10(c). The overlapped blocks in their MC regions are actually the shaded blocks in Figures 10(a) and (b).

After identifying the shared space among different nets, we can now count the actual space that can be used to reduce crosstalk.

Definition 7 (MC Space and MC Blocks). For each net i , where $1 \leq i \leq n$, move it left and down both by $i - 1$ units if it is a falling net; or move it left and up both by $i - 1$ units if it is a rising net. The space within the MC regions then (shared space is counted only once) is called *MC space*. The whole MC space may be divided into blocks by MC boundaries. Each block is called an *MC block*.

To make it clear, we consider the following example composed of four falling nets. The nets together with their MC boundaries are given in Figure 11(a). As shown in Figure 11(b), net 2, net 3, and net 4 are moved left and down by 1, 2, and 3 units, respectively. The MC space is the space within the MC regions. It is composed of three MC blocks, S_1 , S_2 , S_3 , which are separated by some MC boundaries. Showing each net separately in Figure 11(c), we find that S_1 is shared by net 1 and net 2, S_2 is shared by all four nets, and S_3 is shared by net 3 and net 4.

The following lemma summarizes the relationship between MC space insertion and crosstalk reduction.

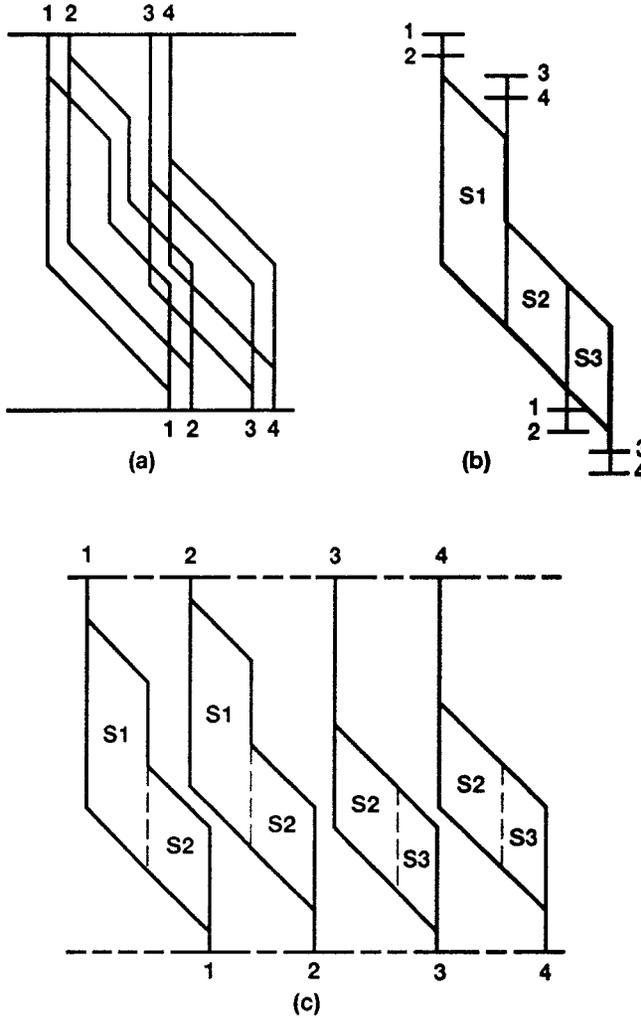


Fig. 11. Identifying MC blocks and MC space.

LEMMA 3. Let p_i be the potential crosstalk between net i and net $i + 1$. If, in a routing solution composed of MC routes, the crosstalk between the two nets is R_i , then at least $p_i - R_i$ units of MC space is inserted between them.

PROOF. According to the definition of potential crosstalk, the two MC routes will have crosstalk p_i when there is no MC space inserted. Because the crosstalk comes only from straight lines, and all MC routes of a net will have the same length of straight lines, there must be $p_i - R_i$ unit length of wires on the straight lines that have a distance of at least two from the other net. This means at least $p_i - R_i$ units of MC space are between these two nets. □

In summary, if we put two neighboring MC routes close to each other, the potential crosstalk p_i will be induced. Based on the properties of MC

routes, in order to reduce one unit of crosstalk, one unit of MC space must be inserted. The MC space is composed of MC blocks, each of which is shared by a set of nets. Therefore, in order to get a feasible routing solution, MC space must be allocated properly to each pair of nets. The space providers are the MC blocks, each with a finite amount of space. The space consumers are all pairs of neighboring nets, each with a requirement of $\max\{p_i - C_i, 0\}$. Of course, each MC block has a limit on maximum achievable crosstalk reduction (recall the case in Figure 9(b)). Furthermore, a set of blocks may also have such a limit in total crosstalk reduction (explained in Section 3.3). These restrictions, together with the space allocation problem, can be modeled by a flow network. This is also presented in Section 3.3.

3.3 Network Flow

In this section, we solve the space allocation problem derived in the previous section. We have MC blocks, each with a finite amount of MC space that will be allocated to a set of pairs of neighboring nets. Each pair of neighboring nets, based on their potential crosstalk p_i and crosstalk bound C_i , requires $\max\{p_i - C_i, 0\}$ units of MC space to be feasible. Since the objective of MC space allocation is to reduce crosstalk, the amount allocated from a set of blocks to each pair of nets must be restricted to make sure the same amount of crosstalk can actually be reduced. We model the space allocation as a maximum flow problem. The flow network can be constructed as follows.

Suppose we identified m MC blocks in the CRR problem. For each MC block S_i , where $1 \leq i \leq m$, a node S_i is introduced. Also, there will be $n - 1$ nodes D_1, D_2, \dots, D_{n-1} which represent the $n - 1$ pairs of neighboring nets. We plan to use the flow from S_i to D_j to represent the amount of MC space allocated from block S_i to the pair of nets j and $j + 1$. Therefore, we add a source node s and a sink node t . For $1 \leq i \leq m$, there is an edge from s to each S_i , whose capacity is the amount of space in block S_i , used to model the available space. For $1 \leq j \leq n - 1$, there is also an edge from each D_j to t with capacity of $\max\{p_j - C_j, 0\}$, representing the amount of space requirement.

The network structure between nodes S_i , for $1 \leq i \leq m$, and nodes D_j , for $1 \leq j \leq n - 1$, is used to control the space allocation, such that, if an amount of space is allocated from a block, the same amount of crosstalk can be reduced. It can be constructed independently for each pair of nets. Consider net j and net $j + 1$. Node S_i can have flows into D_j if and only if the two nets may have crosstalk within or along the boundary of block S_i . This amount of crosstalk also gives the limit on the maximum achievable crosstalk reduction by S_i , and is used as the capacity on the edge going from S_i . For adjacent blocks with common boundaries, the maximum crosstalk they can reduce together is smaller than the summation of their limits. In this case, a *restriction node* R_k is introduced. The edges from these blocks are made to enter R_k and the capacity on the edge from R_k is set to control the flow.

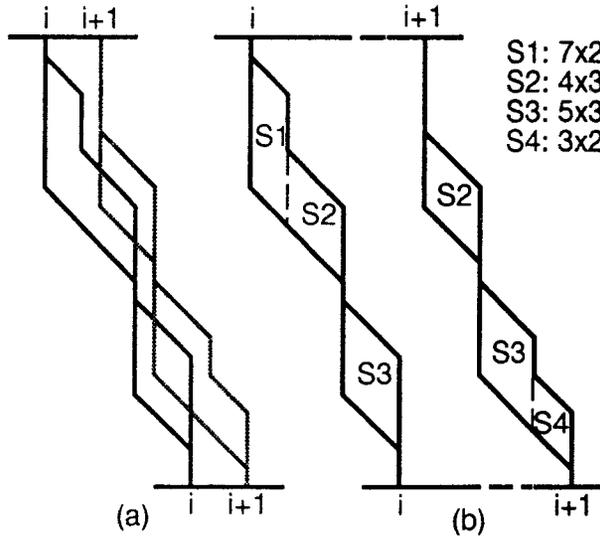


Fig. 12. MC blocks of two neighboring nets.

To make it clear, we consider a pair of nets with their MC boundaries shown in Figure 12(a). Suppose their MC blocks are identified as in Figure 12(b). The dimension of each block is given at the upper-right corner, where the first number is the length of the straight boundary, and the second number is the width of the block. Here, net i and net $i + 1$ could be routed along the right boundary of block S_1 and induce 7 units of crosstalk on the straight lines. Therefore, S_1 can reduce at most 7 units of crosstalk for them. They could also be routed closely in block S_2 and induce 4 units of crosstalk, which is the length of straight lines they have in block S_2 . Therefore, S_2 can reduce at most 4 units of crosstalk. But S_1 and S_2 have a common boundary and together can reduce at most 7 units of crosstalk. Hence, a restriction node R_1 is introduced. As shown in Figure 13, three edges are used to control the flows: (S_1, R_1) with capacity of 7, (S_2, R_1) with capacity of 4, and (R_1, D_i) with capacity of 7. Similarly, S_3, S_4 can reduce at most 5 and 3 units of crosstalk separately, but only 5 units together. These are controlled by the edges (S_3, R_2) , (S_4, R_2) , and (R_2, D_i) . Doing the construction for every pair of neighboring nets, we get a network as shown in Figure 13.

The correctness of our approach is given by the following theorem.

THEOREM 2. *Suppose $G = (V, E, s, t, C)$ is the flow network constructed from a given CRR problem, where s and t are the source and sink, respectively. The given problem has a feasible routing solution if and only if there is a flow that saturates all edges entering t .*

PROOF.

\Leftarrow) Suppose the given CRR problem has a feasible routing. According to Theorem 1, we can assume it is composed of MC routes. By Lemma 3, at

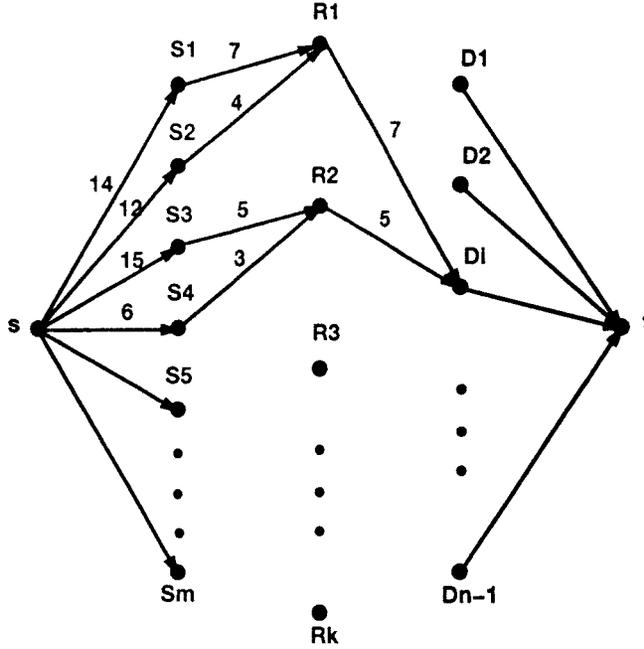


Fig. 13. Flow network for space allocation.

least $p_i - C_i$ units of MC space are between net i and net $i + 1$, for $1 \leq i \leq n - 1$. This gives a flow in G saturating all edges entering t .

\Rightarrow) If G has a flow that saturates all edges entering t , a feasible routing solution can be constructed by using the information from the flow. The nets will be routed one by one from left to right. When routing each net, say net i , two conditions must be satisfied: the crosstalk between net $i - 1$ and net i , R_{i-1} , is no more than C_{i-1} , and if the total flow from S_j to $D_i, D_{i+1}, \dots, D_{n-1}$ is M then at least M units of space in S_j are at the right of net i . For the first net, net 1, routing trivially along its left MC boundary fulfills both conditions. Now suppose net i has been routed. We initially route net $i + 1$ as close to net i as possible. Consider each flow to D_i with value z . If it comes from a block presented only in MC regions of net i , at least z units of spaces have been left between the route of net i and its right MC boundary. Hence at least z units of crosstalk have been reduced from potential crosstalk p_i . Otherwise, it comes from a block in MC regions of net $i + 1$. In this case, we can insert z units of space from the block along the straight lines of net $i + 1$. This will also reduce z units of crosstalk. Since the total amount of flow to D_i is $p_i - C_i$ if $p_i > C_i$, the final crosstalk R_i is no more than $p_i - (p_i - C_i) = C_i$. Since we use only spaces flowing to D_i , after routing of net $i + 1$, the second condition is also kept.

To make it clear, consider again the example in Figure 12. Suppose net i has been routed as in Figure 14(a), and the flows from S_1, S_2, S_3 , and S_4 to D_i are 2, 4, 3, and 0, respectively. Then net $i + 1$ will be routed as in Figure 14(b). \square

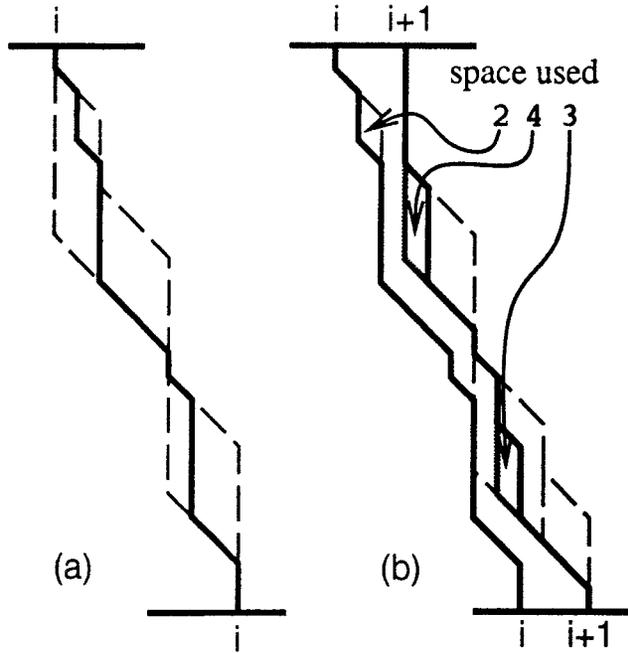


Fig. 14. Construct routing from maximum flow.

Based on the preceding theorem, a routing solution that maximizes the min-slack can be easily constructed.

COROLLARY 2.1. *Suppose a CCRR problem has a feasible solution and $G = (V, E, s, t, C)$ is the flow network derived from it. If we simultaneously increase the capacities on all edges entering t while keeping them saturated, the last maximum flow will give a routing solution with maximum min-slack. \square*

3.4 Algorithm Description

Summarizing the contents of the previous subsections, the pseudocode of the algorithm can be presented as follows.

Optimal Algorithm for CCRR Problem

1. Construct legal boundaries;
2. Derive MC boundaries;
3. Compute potential crosstalk;
4. Identify MC space and MC blocks;
5. Build flow network $G = (V, E, s, t, C)$;
6. Compute max-flow of G ;
7. if the flow does not saturate edges entering t
8. then "No feasible solution";
9. else {
10. Increase capacities on edges entering t simultaneously by 1;
11. Augment current flow;

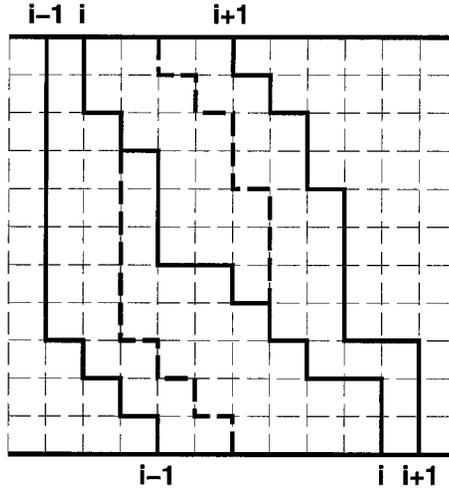


Fig. 15. Contour.

12. } while flow saturates edges entering t ;
13. Route nets according to the flow.

It is easy to see that, in the preceding algorithm, the maximum flow computation will be the dominant part of time complexity. Suppose there are n nets, and the width and length of the channel are W and L , respectively. It can be proved that the number of MC blocks is bounded by $O(n)$. In the worst case, each net can have $O(n)$ blocks in its MC regions. So the numbers of nodes and edges in the flow network are $O(n^2)$. Since the total amount of spaces is at most WL , simply using the Ford–Fulkerson [1962] algorithm gives us $O(n^2WL)$ worst-case running time.

4. POSTPROCESSING

As we can see, routing solutions given by the preceding algorithm are composed of MC routes, which may have many stairs. Although stairs are useful to reduce crosstalk, not all of them are necessary in the final solution. To reduce the number of bends, we design a heuristic as a postprocessing procedure for the preceding algorithm.

Intuitively, the procedure is an iterative “rip-up and reroute” process. Each time, only one net is chosen to be rerouted, and other nets are all fixed. We want to find a new route with fewer bends than the old one, but more important, it cannot increase the crosstalk (on itself or others). To fulfill this requirement, the *contour* of a chosen net can be defined as follows. Given a routing solution, the *left contour* of net i is its leftmost route such that the segments inducing crosstalk are on its given route; the *right contour* of net i is its rightmost route such that the segments inducing crosstalk are on its given route. For example, in the given routing solution shown in Figure 15, the dashed lines show the contour of net i .

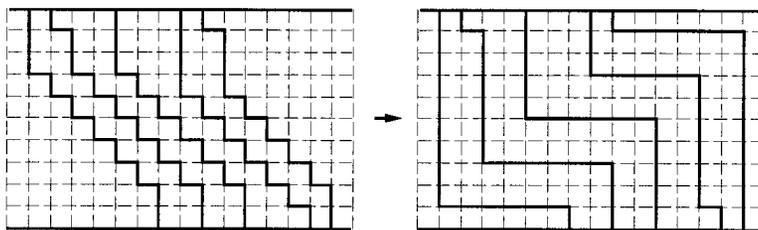


Fig. 16. Postprocessing.

It is easy to see that, in a feasible routing solution, changing one route within its contours will end up with another feasible routing solution. The postprocessing procedure consists of the following passes. In the first pass, net 1 to net n are changed, one by one in that sequence. For each net, the contour is first derived; then a line-search algorithm (such as Mikami and Tabuchi [1968]) is used to find a leftmost route with minimum bends within the contour. The second pass processes in the reverse order, from net n to net 1. Instead, for each net, a rightmost minimum bend route is found within the contour. An example result of the postprocessing procedure is shown in Figure 16.

5. EXPERIMENTAL RESULTS AND CONCLUDING REMARKS

To test the efficiency of our algorithm, we implemented it on an IBM RS6000 machine. After computing potential crosstalk and identifying MC blocks, the program constructs a flow network and writes it out in the DIMACS format. A maximum flow program (here we use **gold_hlf** [Anderson and Setubal 1992]) is then run on the network to compute the maximum flow. Since there are no benchmarks for river routing, let alone those with crosstalk constraints, routing instances are generated by a program. For each instance, we give two parameters: channel width and number of nets. The pin positions and crosstalk constraints are randomly generated. Here, we report the results on 10 such instances P_1, P_2, \dots, P_{10} . In addition to the running time, for each instance, the number of identified MC blocks and the number of nodes and arcs in the networks are also reported in Table I.

In this article, we presented an optimal algorithm to generate routing solutions for the CCRP problem. In the formulation of the problem, crosstalk constraints are specified on each pair of neighboring nets and routes are required to be monotonic. But in practice, crosstalk constraints may be given on each net. In this case, our algorithm can follow a preprocessing step that translates “net constraints” into “net-pair constraints.” Furthermore, although monotonic routes are sufficient for the traditional river routing problem, it is possible to find examples where nonmonotonic routes are needed to satisfy crosstalk constraints. Whether the CCRP problem with nonmonotonic routes can be solved in polynomial time is still an open problem.

Table I. Experimental Results

data	#nets	chnl width	#MC blocks	#nodes	#arcs	time(sec.)
P_1	20	7	4	26	30	0.01
P_2	30	10	29	123	219	0.01
P_3	35	35	33	396	948	0.03
P_4	50	30	126	705	1488	0.04
P_5	100	50	239	1497	3221	0.10
P_6	200	100	1142	8882	19844	0.55
P_7	200	100	2112	8710	18704	0.33
P_8	400	100	881	4156	8739	0.24
P_9	500	100	1323	8720	19351	0.57
P_{10}	500	200	2169	17914	45619	1.26

REFERENCES

- ANDERSON, R. AND SETUBAL, J. 1992. MAXFLOW. Available at ftp://dimacs.rutgers.edu in directory /pub/netflow/submit/setubal.
- BAKOGLU, H. B. 1990. *Circuits, Interconnections and Packaging for VLSI*. Addison-Wesley, Reading, MA.
- CHAUDHARY, K., ONOZAWA, A., AND KUH, E. S. 1993. A spacing algorithm for performance enhancement and cross-talk reduction. In *Proceedings of the International Conference on Computer-Aided Design* (Santa Clara, CA, Nov.), 697–702.
- CHEN, H. H. AND WONG, C. K. 1992. Wiring and crosstalk avoidance in multi-chip module design. In *Proceedings of the Custom Integrated Circuits Conference* (Boston, MA, May), 28.6.1–28.6.4.
- DOLEV, D., KARPLUS, K., STRONG, A., AND ULLMAN, J. D. 1981. Optimal wiring between rectangles. In *Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing* (Milwaukee, WI, May), 312–317.
- FORD, J. R. AND FULKERSON, D. R. 1962. *Flows in Networks*. Princeton University Press, Princeton, NJ.
- GAO, T. AND LIU, C. L. 1993. Minimum crosstalk channel routing. In *Proceedings of the International Conference on Computer-Aided Design* (Santa Clara, CA, Nov.), 692–696.
- GAO, T. AND LIU, C. L. 1994. Minimum crosstalk switchbox routing. In *Proceedings of the International Conference on Computer-Aided Design* (San Jose, CA, Nov.), 610–615.
- KIRKPATRICK, D. A. AND SANGIOVANNI-VINCENTELLI, A. L. 1994. Techniques for crosstalk avoidance in the physical design of high-performance digital systems. In *Proceedings of the International Conference on Computer-Aided Design* (San Jose, CA, Nov.), 616–619.
- LENGAUER, T. 1990. *Combinatorial Algorithms for Integrated Circuit Layout*. Wiley, New York.
- LLOYD, E. L. AND RAVI, S. S. 1984. One-layer routing without component constraints. *J. Comput. Syst. Sci.* 28 (June 1984), 420–438.
- MIKAMI, K. AND TABUCHI, K. 1968. A computer program for optimal routing of printed circuit connectors. *IFIPS Proc. H47*, 1475–1478.
- THAKUR, S., CHAO, K.-Y., AND WONG, D. F. 1995. An optimal layer assignment algorithm for minimizing crosstalk for three layer VHV channel routing. In *Proceedings of the IEEE International Symposium on Circuits and Systems* (Seattle, WA, Apr. 30–May 3), 207–210.

Received December 1996; revised August 1997; accepted October 1997