An Optimal Algorithm for River Routing with Crosstalk Constraints

Hai Zhou and D.F. Wong
Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712-1188

Abstract

With the increasing density of VLSI circuits, the interconnection wires are getting packed even closer. This has increased the effect of interaction between these wires on circuit performance and hence, the importance of controlling crosstalk. In this paper, 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 nets, we give a polynomial time algorithm to decide whether there is a feasible river routing solution and produce one with minimum crosstalk whenever the problem is feasible.

1 Introduction

With VLSI fabrication entering the deep submicron era, devices and interconnection resources 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 wiring resources. Increased coupling effect not only increases signal delays, but also decreases signal integrity due to transmission line behavior. This phenomenon is called *crosstalk* [2].

In the literature, previous works on the crosstalk problem in detailed routing fall into two main categories. In the first category, the gridless routing model is used and spacings between wires are adjusted to reduce crosstalk [3, 8]. In the second category, the gridded routing model is used; works in this category focus on channel routing [5, 7, 11] and switchbox routing [6]. The general approach used in all previous works consists of two steps. First, a routing solution is determined by a conventional routing algorithm. Then, a post-processing algorithm is designed to modify the routing solution to reduce crosstalk.

In this paper, we consider the river routing problem with crosstalk constraints. Although river routing has the best developed theory among all detailed routing problems [9], there is no previous work which considered it together with crosstalk. Contrary to works in [3, 5, 6, 7, 8, 11], 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 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 paper 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 post-processing heuristic to reduce the number of bends in the solution. Finally, Section 5 concludes the paper with experimental results and some remarks.

2 Problem Formulation

In river routing, we are given a single-layer rectangular routing channel with pins located at the top and the bottom of the channel. Let $t_1, t_2, \ldots, t_n$ be the pins at the top boundary of the channel (from left to right) and $b_1, b_2, \ldots, b_n$ be the pins at the bottom boundary of the channel (from left to right). We assume that there is a grid super-imposed 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 go under at most two directions. All previous works on river routing used this assumption, since it does not hurt the routability [9], and at the same time, also assure each route to be 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 in a routing solution is called a *legal* route.

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 increase, 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. Crosstalk between two routes is then the total number of their adjacent grid edges. Note that this is also the crosstalk model used in [5, 6]. For any routing solution, let R_i denote the crosstalk between routes i and

ICCAD '96
1063-6757/96 $5.00

$i+1$, $1 \leq i \leq n-1$. As an example, Figure 1 shows a river routing solution and the crosstalk between each pair of routes. Here, $R_1 = 2+2 = 4, R_2 = 3, R_3 = 2, R_4 = 1+1 = 2$, and $R_5 = 2$.

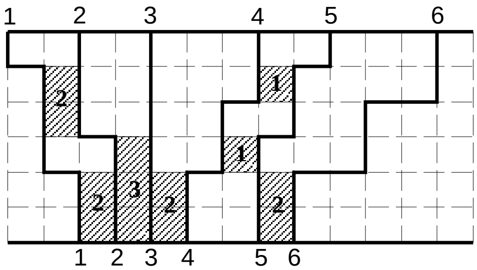

Figure 1: River routing solution and crosstalk

For each pair of neighboring nets i and $i+1$, $1 \leq i \leq n-1$, a constant C_i is given as the maximum tolerable crosstalk between them. A routing solution such that $R_i \leq C_i, 1 \leq i \leq n-1$ is called a *feasible* routing solution. A route in a feasible routing solution is called a *feasible* route. We also call $C_i - R_i, 1 \leq i \leq n-1$ 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 paper can be stated as follows: *Crosstalk-Constrained River Routing (CCRR) Problem*: Given a channel of width W, a set of n nets $(t_1, b_1), (t_2, b_2), \ldots, (t_n, b_n)$, and a set of crosstalk constraints $C_1, C_2, \ldots, C_{n-1}$, determine whether there is a feasible routing solution. If there is one, give a routing solution with maximum min-slack.

3 Optimal Algorithm

3.1 MC Routes

Given a CCRR problem, as Figure 2 shows, we can start from left to right and route each net as far to the left as possible; and then from right to left route each net as far to the right as possible. For each net, this will give us two extreme routes. We call the leftmost route the *left legal boundary* and the rightmost route the *right legal boundary.* It is easy to see, for a single net, a route is legal if and only if it is within the legal boundaries.

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

According to the definition, for the route shown in Figure 3(a), the circled segments are stairs. It is not hard to see that, except the start and end edges, no part of a stair can have crosstalk with other routes. In order to simplify our presentation, in the rest of the paper, we will use 45° line to represent the middle part of a stair, which can never have crosstalk. In addition, we will call other parts of a route *straight lines*, or specifically *horizontal/vertical lines* when we want to emphasis their orientations. Such representation for the net in Figure 3(a) is shown in Figure 3(b).

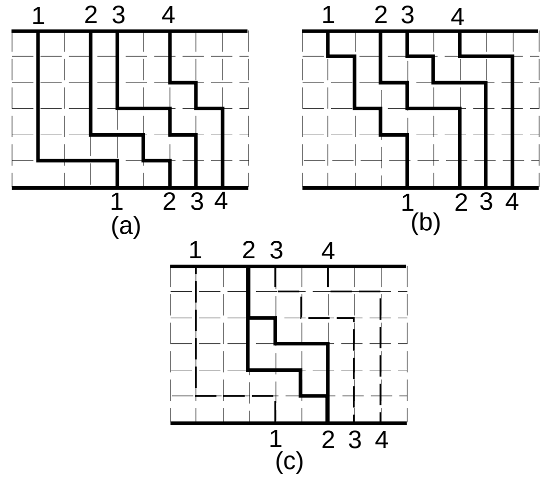

Figure 2: (a)Leftmost routes; (b)Rightmost routes; (c)Legal boundaries

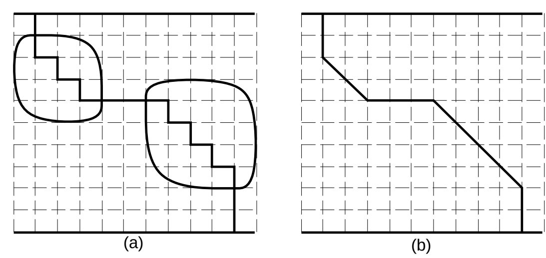

Figure 3: Stairs and their representation

Definition 2 (derivation) *We say a legal route w' is derived from another legal route w if w' is formed by changing part of w to a stair whose start and end edges are on w. If w_1 is derived from w_2, w_2 is derived from w_3, we also say that w_1 is derived from w_3.*

Figure 4(a) shows a legal route, and 4(b) shows a route derived from it. Dashed lines are the legal boundaries.

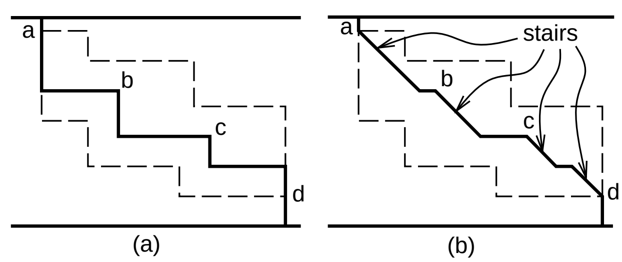

Figure 4: (a)Legal route; (b)Derived MC route

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

In fact, if w' is derived from w, the straight lines in w' are only part of those in w. So, w' can not induce more crosstalk than w. When a route has itself as the only derived route, it already reaches minimal length of straight lines, hence, is what we desire in minimum crosstalk routing.

Definition 3 (MC route) *A legal route is called an MC (minimal-crosstalk) route if it has itself as the only derived route.*

It is easy to check that the route in Figure 4(b) is an MC route. The following theorem shows that we can consider only MC routes in our routing.

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

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. A region surrounded by the left and right MC boundaries is called an MC region.*

The MC boundaries and MC regions of one net are shown in Figure 5. The dashed lines are the left and right legal boundaries of the net.

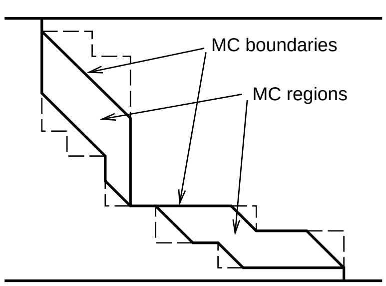

Figure 5: MC boundaries and MC regions

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;
3. The horizontal and vertical lines of a MC route, if both exist, must lie within separate MC regions.□

3.2 Space Allocation

Based on Theorem 1, to find a feasible routing solution, we need only to consider MC routes.

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

To compute potential crosstalk p_i, we can route net i along its right MC boundary and net $i+1$ as close to it as possible. Crosstalk between these two routes is actually p_i. For the two neighboring nets with their MC boundaries shown in Figure 6(a), Figure 6(b) shows the way to compute the potential crosstalk. Besides these routes, other MC routes may also induce maximum crosstalk. One pair of such routes is shown in Figure 6(c).

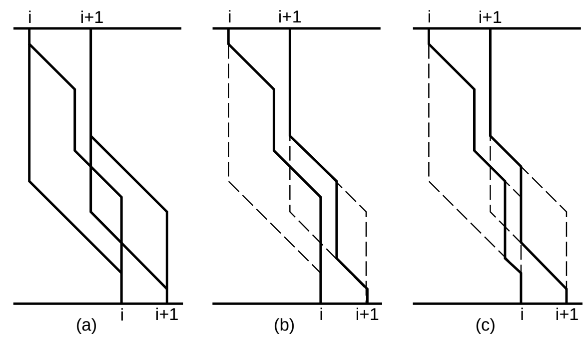

Figure 6: Potential crosstalk

If the crosstalk between two MC routes violates the constraint, we can reduce it by pushing adjacent straight lines away from each other. For example, to reduce the crosstalk in Figure 6(c), we can change straight line ab to $a'b'$ or straight line cd to $c'd'$, as shown in Figure 7(a). This can be viewed as inserting the spaces between ab and $a'b'$, or those between cd and $c'd'$. Generally speaking, because every MC route of a net has the same length of straight lines, in order to reduce one unit of crosstalk, one unit of such space must be inserted. We must also notice that the other direction is not always true. Compared with Figure 7(a), Figure 7(b) has more spaces inserted, but no crosstalk is reduced.

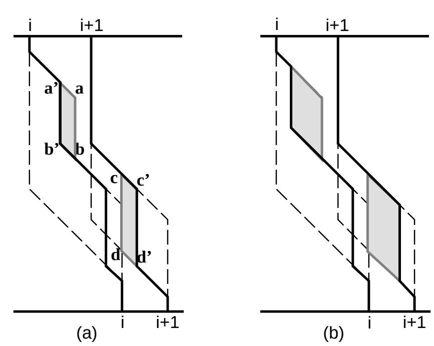

Figure 7: Crosstalk reduction vs. space insertion

Since only MC routes are considered, the only spaces that can be used to reduce crosstalk are those within MC regions. Note that we can not count the spaces in each MC region independently. For example, consider the two neighboring nets with their MC boundaries shown in Figure 8. When the horizontally shaded region is used by nets at the left of net i (that is, net i is routed along the right boundary of the region), net $i+1$ can only be routed at the right of the vertically shaded region. Therefore, the two shaded regions must be considered equivalent.

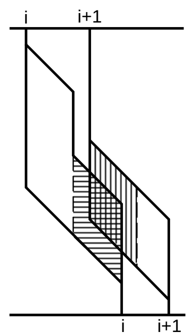

Figure 8: Equivalent spaces

We can define equivalent spaces as follows. Here a *falling net* means its top pin is at the left of its bottom pin, and a *rising net* means its top pin is at the right of its bottom pin [9].

Definition 6 (equivalent spaces) *Given net i and j with $i < j$, move MC regions of nets j $j-i$ units down and left if it is a falling net, or $j-i$ units up and left if it is a rising net. The overlapped spaces in their MC regions are called equivalent spaces.*

Treating equivalent spaces as the same resource, we can compute the total spaces which can be used to reduce crosstalk.

Definition 7 (MC spaces and MC blocks) *For $1 \le i \le n$, move MC regions of net i $i-1$ units down and left if it is a falling net, or $i-1$ units up and left if it is a rising net. The total spaces in all MC regions, counting overlapped ones only once, are called MC spaces. MC spaces are divided into small blocks by MC boundaries. Each block is called an MC block.*

Lemma 3 *Given a routing solution composed of MC routes in which the crosstalk between nets i and $i+1$ is R_i. If $R_i < p_i$, then at least $p_i - R_i$ units of MC spaces are between nets i and $i+1$.* □

From the above discussion, MC spaces are the only resource which can be used to reduce crosstalk and one space can be used only by one pair of nets. Based on Lemma 3, in order to get a feasible routing solution, we need to allocate at least $p_i - C_i$ units of MC spaces to nets i and $i+1$, for $1 \le i \le n-1$. This space allocation problem can be solved by a network flow technique and it will be presented in the next section.

3.3 Network Flow

Let s and t be the source and sink of the flow network. If MC spaces are composed of m blocks, m nodes $S_1, S_2, \ldots, S_m$ will be introduced, each represents one block. There will be an edge from s to each S_j, with capacity equal to the amount of spaces in block S_j. There will also be $n-1$ nodes $D_1, D_2, \ldots, D_{n-1}$ which represent the $n-1$ pairs of nets. Each D_i will have an edge to t with capacity of $p_i - C_i$, which is the amount of spaces needed in order for nets i and $i+1$ to satisfy the constraint.

A flow from S_j to D_i will represent the amount of spaces allocated from block S_j to nets i and $i+1$. Lemma 3 shows that crosstalk reduction comes only from space allocation. The network structure must assure the other direction, that is, if 1 unit of flow goes from S_j to D_i, 1 unit of crosstalk between nets i and $i+1$ can be actually reduced. To enforce this, first, only those blocks presented in the MC regions of nets i and $i+1$ can have flows to D_i, $1 \le i \le n-1$. Second, to avoid such cases as that in Figure 7(b) to happen, flows from MC blocks must be upper bounded. For a single block S_j, the flow from it must be restricted by the maximum crosstalk S_j can reduce, which is given by the length of straight lines an MC route may have in S_j. For adjacent blocks with common straight boundaries, the total flows from them can not be greater than the length of straight lines an MC route may have in these blocks.

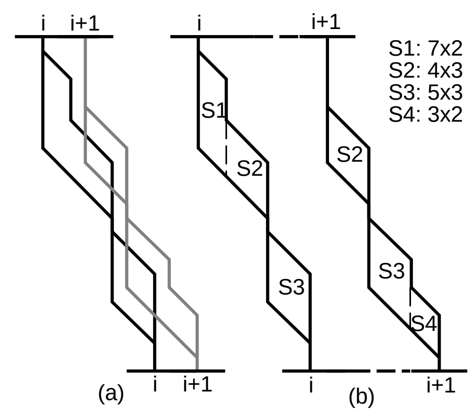

Figure 9: MC blocks of two neighboring nets

To illustrate the construction process, let us consider the pair of nets shown in Figure 9(a) as an example. Suppose MC blocks are identified as in Figure 9(b). The amount of spaces in each block is shown at the upper-right corner, where the first number is the length of the straight boundary, and the second is the width of the block. Here, S_1, S_2, S_3 and S_4 can reduce at most 7, 4, 5 and 3 units of crosstalk, respectively. These restrictions are enforced by the capacities on edges (S_1, R_1), (S_2, R_1), (S_3, R_2) and (S_4, R_2) in Figure 10. Since blocks S_1 and S_2 share a straight boundary of length 3, at most 7 units of crosstalk can be reduced by them together. This is enforced by a restriction node R_1 and the capacity on edge (R_1, D_i). Similar construction is given for S_3 and S_4. Continue the construction for other pairs of nets, We can get a flow network as shown in Figure 10.

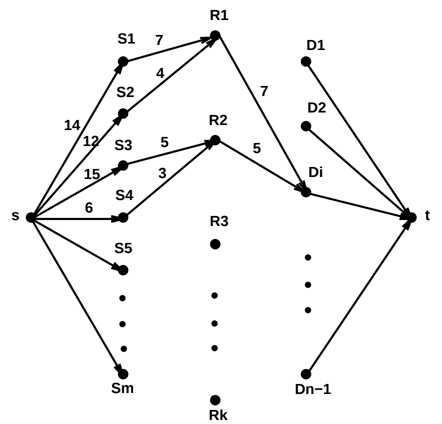

Figure 10: Flow network for space allocation

Theorem 2 *Suppose $G = (V, E, s, t, C)$ is the flow network constructed from a given CCRR problem, where s and t are the source and sink, respectively. The given problem has a feasible routing solution if and only if $(V - \{t\}, \{t\})$ is a min-cut of G.* □

In fact, if $(V - \{t\}, \{t\})$ is a min-cut of G, a feasible routing solution can be constructed by using the information from a maximum 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. One is that the crosstalk between net $i-1$ and net i, R_{i-1}, is no more than C_{i-1}. The other is that if the total flows from S_j to $D_i, D_{i+1}, \ldots, D_{n-1}$ is M then at least M units of spaces 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 an MC region 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 has 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 spaces from the block along the straight lines of net $i+1$. This will also reduce z units of crosstalk. Since the total amount of flows to D_i is $p_i - C_i$, 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 9. Suppose net i has been routed as in Figure 11(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 11(b).

Corollary 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 incident with t until they are no longer min-cut, the maximum flow then will give a routing solution with maximum min-slack.*□

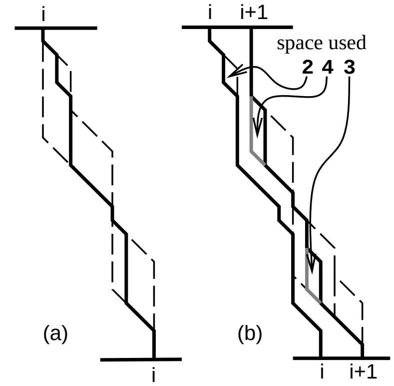

Figure 11: Construct routing from maximum flow

3.4 Algorithm Description

The pseudo-code 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 spaces and MC blocks;
5.  Build flow network G = (V, E, s, t, C);
6.  Compute max-flow of G;
7.  if (V - {t}, {t}) is not a min-cut
8.     then ``No feasible solution'';
9.  else{
10.    Increase capacities on edges to t
       simultaneously by 1;
11.    Augment current flow;
12.  } while ((V - {t}, {t}) is a min-cut);
13.  Route nets according to the flow.
```

It is easy to see that, in the above 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 Ford-Fulkerson algorithm gives us $O(n^2WL)$ running time [4].

4 Post-Processing

Routing solutions given by the above algorithm are composed of MC routes, which may have many stairs. In fact, not all stairs in the solution are necessary. We have a post-processing procedure to reduce the number of bends.

In a fixed routing solution, define the *left contour* of net i to be the leftmost route of net i which has minimum crosstalk with net $i-1$; and the *right contour* of net i to be the rightmost route of net i which has the minimum crosstalk with net $i+1$. For example, in

Figure 12, when the route of net i is given, the dashed routes are the left contour of net $i+1$ and the right contour of net $i-1$.

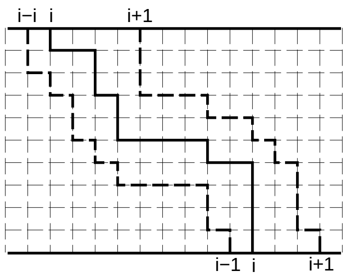

Figure 12: Left and right contours

It can be show that, in a feasible routing solution, changing one route within its contours will end up with another feasible routing solution. The post-processing procedure changes routes net by net, consisting of two passes. The first pass is from net 1 to net n. For each net, the contours are first derived. Then a line-search algorithm [10] can be used to find the leftmost route with minimum bends within the contours. The second pass is similar, but processes from net n to net 1. Instead, rightmost minimum bend routes are found. An example result of the post-processing procedure is shown in Figure 13.

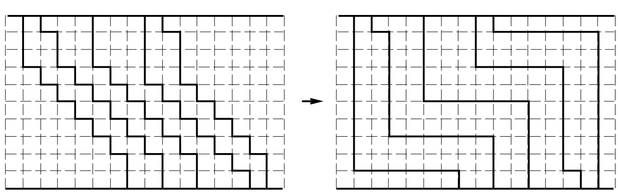

Figure 13: Post-processing

5 Experimental Results and Concluding Remarks

We implemented our algorithm in IBM RS6000. The program constructs the flow network and outputs it to a file in DIMACS format. This file is then input to a maximum flow program (here we use **gold_hlf** [1]) to get a maximum flow. We run the program on 10 problems $P_1, P_2, \ldots, P_{10}$. The results, which include the number of nets, channel width, the number of MC blocks, the number of nodes and arcs in the networks, and the running time (in seconds), are reported in Table 1.

In this paper, we presented an optimal algorithm to generate routing solutions consisting of monotonic routes for the CCRR problem. Although monotonic routes are sufficient for the traditional river routing problem, it is possible to find examples where non-monotonic routes are needed to satisfy crosstalk constraints. Whether the CCRR problem with non-monotonic routes can be solved in polynomial time is an open problem.

Table 1: Experimental results

data	nets	width	blocks	nodes	arcs	time
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

[1] R. Anderson and J. Setubal, MAXFLOW. `ftp://dimacs.rutgers.edu` in directory `/pub/netflow/submit/setubal`

[2] H.B. Bakoglu, *Circuits, Interconnections and Packaging for VLSI.* Addison-Wesley, 1990.

[3] H.H. Chen and C.K. Wong, Wiring and Crosstalk Avoidance in Multi-Chip Module Design. In *Custom Integrated Circuits Conference*, pp. 28.6.1-28.6.4, 1992.

[4] J.R. Ford and D.R. Fulkerson, *Flows in Networks.* Princeton University Press, 1962.

[5] T. Gao and C.L. Liu, Minimum Crosstalk Channel Routing. *Proc. of International Conference on Computer-Aided Design*, 1993.

[6] T. Gao and C.L. Liu, Minimum Crosstalk Switchbox Routing. *Proc. of International Conference on Computer-Aided Design*, 1994.

[7] D.A. Kirkpatrick and A.L. Sangiovanni-Vincentelli, Techniques for Crosstalk Avoidance in the Physical Design of High-Performance Digital Systems. *Proc. of International Conference on Computer-Aided Design*, 1994.

[8] K. Chaudhary, A. Onozawa and E.S. Kuh, A Spacing Algorithm for Performance Enhancement and Crosstalk Reduction. *Proc. of International Conference on Computer-Aided Design*, 1993.

[9] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout.* John Willey & Sons, 1990

[10] K. Mikami and K. Tabuchi, A Computer Program for Optimal Routing of Printed Circuit Connectors. *IFIPS Proc.*, vol. H47, pp. 1475-1478, 1968.

[11] S. Thakur, K.-Y. Chao and D.F. Wong, An Optimal Layer Assignment Algorithm for Minimizing Crosstalk for Three Layer VHV Channel Routing. *Proc. IEEE International Symposium on Circuits and Systems*, 1995.