Crosstalk-Constrained Maze Routing Based on Lagrangian Relaxation

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, interconnection wires are getting packed even closer. This has increased the effect of interaction between wires on circuit performance and hence, the importance of controlling crosstalk. Maze routing is a robust and general approach which can be used in many situations including multi-layer interconnections. In this paper, we consider crosstalk avoidance in maze routing. The crosstalk-constrained maze routing problem is first formulated. Then we show it is equivalent to the multi-constrained shortest path problem and is strongly NP-complete. Based on the Lagrangian relaxation technique, an effective heuristic algorithm is designed to solve the problem.

1 Introduction

With VLSI fabrication entering the deep sub-micron 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* [2].

Previous works on crosstalk avoidance in detailed routing of ICs use spacing, track permutation, segment permutation, and layer assignment to reduce crosstalk [3, 5, 6, 9, 13, 14]. These works are mainly focused on a channel or a switchbox. As VLSI technologies evolved, multi-layers of interconnection are commonly available. In multi-layer technology, channels are no longer accurate models for routing resources. In fact, maze routing is a robust and general approach which can be used in this situation.

In this paper, we deal with how to control crosstalk in maze routing. Maze routing belongs to the category of sequential routing algorithms, that is, it routes only one single net at a time. After a net has been routed, the area occupied by its wiring will be marked as obstacles to the unrouted nets. When routing a net in the presence of obstacles and routed nets, besides crosstalk induced on the current net, extra crosstalk may be created on other routed nets by the current one. Therefore, we need to consider how to simultaneously control crosstalk on each net when routing a new net. Under a simplified crosstalk model and the assumption of two-terminal nets, we formulate it as a *crosstalk-constrained maze routing problem*. It is then modeled as a *multi-constrained shortest path problem* and shown to be strongly NP-complete. Based on the Lagrangian relaxation technique, an effective heuristic algorithm is designed to solve it. This approach is so robust that it can be easily modified to accommodate more general crosstalk models and multi-terminal nets.

The rest of the paper is organized as follows: Section 2 gives the formulation of the crosstalk-constrained maze routing problem. Section 3 discusses the complexity of the problem. In Section 4 we present a heuristic algorithm for the problem which is based on the Lagrangian relaxation technique. Section 5 considers extensions to more general crosstalk models and multi-terminal nets. Finally, Section 6 concludes the paper with experimental results.

2 Problem formulation

In maze routing, the entire routing area is represented as a k-layer grid graph. All pins, edges of bounding boxes that enclose the cells are aligned on the grids. Each edge in the graph represents a track that can accommodate exactly one wire. If obstacles obstruct the routing region, the obstructed edges are deleted from the graph and a subgraph of the k-layer graph results. To simplify our presentation, we use only one layer in this paper. The extension to k layers is straight-forward. Given a set of *nets*, each of which is defined by a set of pins need to be connected, a *routing solution* is defined as a collection of node-disjoint trees each connects all pins in one net.

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, we assume that crosstalk only exists between wires in adjacent rows or columns. Furthermore, we will set the proportion constant to 1 and treat the adjacent length between two wires as their crosstalk. The total crosstalk on one net is then defined as the summation of crosstalk induced on it. This is a simplified crosstalk model which was also used in [5, 6, 14]. It is used here to simplify the presentation. More general models can be easily adopted and will be presented in Section 5.

Suppose there are n nets. Then, for each net i, where $1 \leq i \leq n$, a constant C_i will be given as the maximum crosstalk net i can tolerate. For any routing solution, let R_i denote the crosstalk on net i for $1 \leq i \leq n$. If $R_i \leq C_i$, for all $1 \leq i \leq n$, the routing solution is called a *feasible routing solution*. For example, under the above crosstalk model, the crosstalk of the nets shown in Figure 1 is: $R_1 = 2+7+1 = 10, R_2 = 2+2 = 4, R_3 = 5+3+7+2+1 = 18, R_4 = 5+3 = 8$.

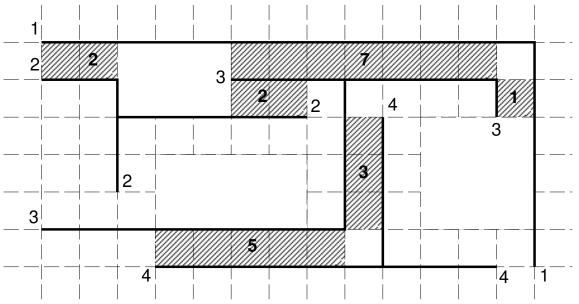

Figure 1: A routing solution and its crosstalk

Since simultaneously routing all nets is prohibitively complicated, a sequential routing approach is adopted here. First, an order is decided on all nets. Then, nets are routed one at a time according to that order. After routing one net, the area used by that net is no longer available to other nets and will be treated as obstacles. Obviously, net ordering is of great importance in this approach. But there is no generally agreed method to compute it. In practice, rip-up and re-route is heavily employed to alleviate the dependence on net ordering. Therefore, the main problem in maze routing is how to route one net in the presence of obstacles and routed nets. We first consider only two-terminal nets here. Extension to multi-terminal nets will be given in Section 5.

When routing a net among previously routed nets, besides crosstalk on the current net, there is also extra crosstalk induced on other routed nets. Therefore, it is necessary to control the crosstalk on all nets simultaneously. This can be formulated as the following problem.

Problem 1 (Crosstalk-Constrained Maze Routing (CCMR)) *Suppose we have a grid routing graph and nets $1, 2, \ldots, m-1$ are already routed on it. Given net m with two pins p_1, p_2 and crosstalk constraints C_i, for $1 \leq i \leq m$, find a shortest route for net m such that $R_i \leq C_i$ for all $1 \leq i \leq m$.*

Traditional maze routing algorithms route a net by a shortest path or a minimum bend path and can also be modified to accommodate various constraints and path selection criteria [11]. But it is hard to modify them to solve the CCMR problem, since what required here is not minimizing the total crosstalk but balancing crosstalk among different nets. This can be illustrated by the example in Figure 2.

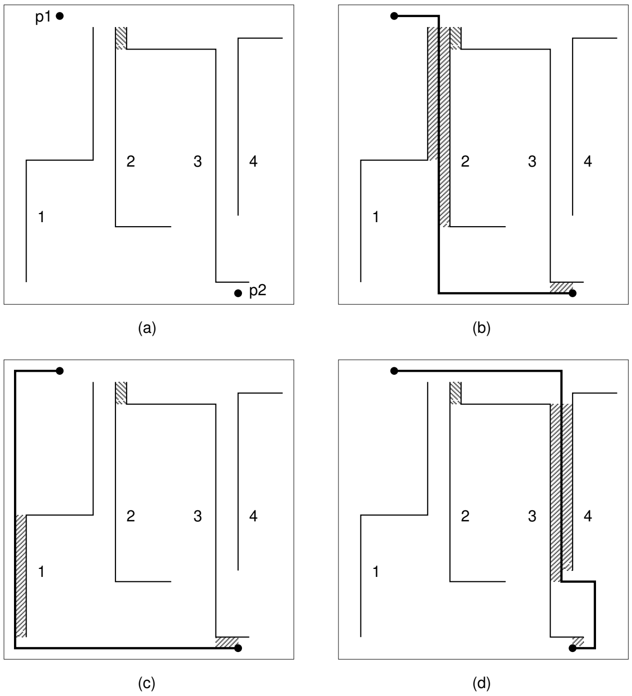

Figure 2: (a)A CCMR problem: nets 1, 2, 3, 4 are routed and net 5 is (p_1, p_2); $C_1 = 2, C_2 = 3, C_3 = 30, C_4 = 36, C_5 = 35$. (b)Shortest path route (not feasible): $R_1 = 12, R_2 = 20, R_3 = 4, R_4 = 0, R_5 = 32$. (c)Minimum total crosstalk route (not feasible): $R_1 = 11, R_2 = 2, R_3 = 4, R_4 = 0, R_5 = 13$. (d)Optimal solution: $R_1 = 0, R_2 = 2, R_3 = 19, R_4 = 15, R_5 = 32$

3 Complexity

Given an instance of the CCMR problem, we can construct a vector-weighted graph as follows. The grid nodes and edges which are occupied by the obstacles and previously routed nets are removed from the grid graph. Each remaining edge e will have a vector $(v_1^e, v_2^e, \ldots, v_m^e)$ as its weight, where v_i^e equals to the crosstalk induced on net i if net m is routed on edge e. For example, given a CCMR problem shown in Figure 3(a), the correspondent graph is constructed in Figure 3(b).

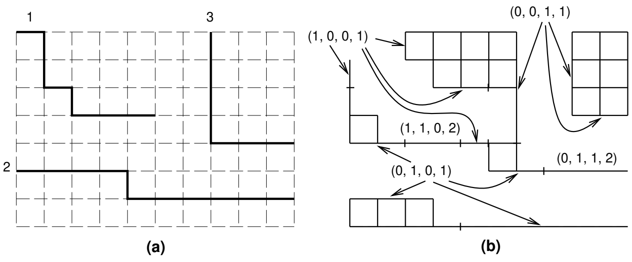

Figure 3: (a)A CCMR instance (b)Corresponding vector-weighted graph (edges not labeled have weight (0,0,0,0))

It can be shown that solving the CCMR problem is equivalent to solving the following problem.

Problem 2 (Multi-Constrained Shortest Path (MCSP)) *Given a vector-weighted graph, two nodes s and t in the graph, and a set of constants $B_1, B_2, \ldots, B_n$, find a shortest path P from s to t such that $\sum_{e \in P} v_i^e \leq B_i, \forall 1 \leq i \leq n$.*

By treating the edge length as another dimension in the weight vectors, the decision version of the above problem can be stated as follows.

Problem 3 (MCSP-Decision) *Given a vector-weighted graph, two nodes s and t in the graph, and a set of constants $B_1, B_2, \ldots, B_{n+1}$, find a path P from s to t such that $\sum_{e \in P} v_i^e \leq B_i, \forall 1 \leq i \leq n+1$.*

This problem is similar to the Shortest Weight-Constrained Path (SWCP) problem in [7]. But SWCP problem is weakly NP-complete, and can even be solved in polynomial time if all weights are equal or all lengths are equal. However, for MCSP problem, we have the following result.

Theorem 1 *MCSP-Decision problem is strongly NP-complete.*

Proof: Given an instance of MCSP-Decision problem, a path from s to t can be checked in polynomial time to see whether the constraints are obeyed. This means that MCSP-decision problem is in NP.

We will show that MCSP-Decision is strongly NP-hard by transforming 3-PARTITION [7] to it. 3-PARTITION is defined as follows.

> INSTANCE: Set A of $3m$ elements, a bound $B \in Z^+$, and a size $s(a) \in Z^+$ for each $a \in A$ such that $B/4 < s(a) < B/2$ and such that $\sum_{a \in A} s(a) = mB$.
> QUESTION: Can A be partitioned into m disjoint sets $A_1, A_2, \ldots, A_m$ such that, for $1 \leq i \leq m, \sum_{a \in A_i} s(a) = B$?

Given one of the instances described above, we can construct an instance of MCSP-Decision as follows. Denote the elements in A by $a_1, a_2, \ldots, a_{3m}$. Corresponding to each $a_i, 1 \leq i \leq 3m$, we will have $m+1$ nodes $s_i, t_{i1}, t_{i2}, \ldots, t_{im}$ and m edges $(s_i, t_{i1}), (s_i, t_{i2}), \ldots, (s_i, t_{im})$. The weight vector of edge (s_i, t_{ij}) is defined as follows: all elements are 0 except the jth element, which will be $s(a_i)$. The nodes and edges for different elements will be connected by adding edges (t_{ij}, s_{i+1}) for $1 \leq i \leq 3m-1, 1 \leq j \leq m$. We will add one more node t and m more edges $(t_{3m,j}, t), 1 \leq j \leq m$. The weights of all the newly-added edges will be 0. This will give us a weighted graph as shown in Figure 4. We will set the constraints $B_1 = B_2 = \cdots = B_m = B$.

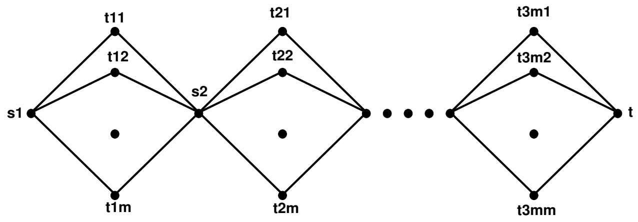

Figure 4: An instance constructed from 3-PARTITION

Now, suppose there is a path P from s_1 to t such that $\sum_{e \in P} v_j^e \leq B, \forall 1 \leq j \leq m$. According to the construction, there are only $3m$ non-zero weight edges on the path, that is,

$$\begin{aligned} \sum_{e \in P} v_j^e &= \sum_{1 \leq i \leq 3m, t_{ij} \in P} v_j^{(s_i, t_{ij})} \\ &= \sum_{1 \leq i \leq 3m, t_{ij} \in P} s(a_i) \\ &\leq B \end{aligned}$$

Hence, the following partition is a YES witness for the 3-PARTITION:

$$A_j = \{a_i | t_{ij} \in P\}, 1 \leq j \leq m$$

On the other hand, suppose there is a partition $A_1, A_2, \ldots, A_m$ such that, for $1 \leq i \leq m, \sum_{a \in A_i} s(a) = B$. Then, we can choose t_{ij} to be on the path if and only if $a_i \in A_j$. It is easy to check that such a path fulfills the constraints of MCSP-Decision problem. ∎

4 A heuristic based on Lagrangian relaxation

As we have found in the NP-completeness proof, the difficulty of MCSP problem comes from the requirement to balance costs among different dimensions of the weight vector. Since minimizing the weighted sum of all dimensions in the weight vector can be easily solved, an alternative way to balance the costs is first giving relative weight for each dimension and then minimizing the weighted sum of them. Lagrangian relaxation is a way to search better relative weights by iteratively adjusting them, and hence, will be used to solve the MCSP problem.

Given an MCSP problem, by relaxing the constraints into the objective function, that is, giving relative weights to the constraints, we have the following problem.

Problem 4 (Lagrangian Sub-Problem (LSP)) *Given an instance of MCSP and a set of non-negative constants $\mu_1, \mu_2, \ldots, \mu_n$, find a path P in the graph such that $\sum_{e \in P} 1 + \sum_{i=1}^n \mu_i(\sum_{e \in P} v_i^e - B_i)$ is minimized.*

Usually, the constants $\mu_1, \mu_2, \ldots, \mu_n$ are called Lagrangian multipliers. Note that

$$\begin{aligned}
&\sum_{e \in P} 1 + \sum_{i=1}^n \mu_i(\sum_{e \in P} v_i^e - B_i) \\
&\quad = \sum_{e \in P}(1 + \sum_{i=1}^n \mu_i v_i^e) - \sum_{i=1}^n \mu_i B_i,
\end{aligned}$$

where $\sum_{i=1}^n \mu_i B_i$ is constant for fixed multipliers. Therefore, by treating $1+\sum_{i=1}^n \mu_i v_i^e$ as the weight on edge e, we can solve the LSP problem efficiently by Dijkstra's shortest path algorithm [4]. Let $L(\mu_1, \ldots, \mu_n)$ denote the optimal objective of the above problem, that is,

$$L(\mu_1, \ldots, \mu_n) = \min_P\{\sum_{e \in P} 1 + \sum_{i=1}^n \mu_i(\sum_{e \in P} v_i^e - B_i)\}.$$

The Lagrangian multiplier problem for MCSP is as follows.

Problem 5 (Lagrangian Multiplier Problem (LMP))

$$\begin{array}{ll}
\textit{Maximize} & L(\mu_1, \ldots, \mu_n) \\
\textit{subject to} & \mu_i \geq 0, \forall 1 \leq i \leq n
\end{array}$$

It is obvious that for any $\mu_i \geq 0, 1 \leq i \leq n$,

$$\begin{aligned}
&\min_P\{\sum_{e \in P} 1 : \sum_{e \in P} v_i^e \leq B_i, 1 \leq i \leq n\} \\
&\geq \min_P\{\sum_{e \in P} 1 + \sum_{i=1}^n \mu_i(\sum_{e \in P} v_i^e - B_i) : \\
&\qquad \sum_{e \in P} v_i^e \leq B_i, 1 \leq i \leq n\} \\
&\geq \min_P\{\sum_{e \in P} 1 + \sum_{i=1}^n \mu_i(\sum_{e \in P} v_i^e - B_i)\} \\
&= L(\mu_1, \ldots, \mu_n).
\end{aligned}$$

Because the above inequality is true for any $\mu_i \geq 0, 1 \leq i \leq n$, it is also true for specific μ_is which make $L(\mu_1, \ldots, \mu_n)$ maximum, that is,

$$\begin{aligned}
&\min_P\{\sum_{e \in P} 1 : \sum_{e \in P} v_i^e \leq B_i, 1 \leq i \leq n\} \\
&\quad \geq \max_{\mu_1, \ldots, \mu_n} L(\mu_1, \ldots, \mu_n).
\end{aligned}$$

This means that the optimal solution of the LMP problem is always a lower bound of the MCSP problem. This property also gives the following optimality criteria for the MCSP problem.

Theorem 2 *If we have a path P such that $\sum_{e \in P} v_i^e \leq B_i$ for all $1 \leq i \leq n$, and $\sum_{e \in P} 1 = L(\mu_1, \ldots, \mu_n)$ for some $\mu_i \geq 0, 1 \leq i \leq n$, then P is an optimal solution to the MCSP problem, and μ_i, for $1 \leq i \leq n$, is an optimal solution to the corresponding LMP problem.*

As stated by the following theorem, the LMP problem has a very good property for optimization.

Theorem 3 *The LMP is a convex programming problem.*

Based on the theorem, non-linear programming techniques can be used to solve the LMP problem. Since $L(\mu_1, \ldots, \mu_n)$ as a function is not everywhere differentiable, sub-gradient method is the usual approach. The pseudo-code of this approach can be described as follows.

Algorithm Sub-gradient Method for LMP
1. $k = 0; \mu_i = 0, \forall 1 \leq i \leq n$;
2. Solve the LSP by Dijkstra's algorithm;
3. If stop criteria is satisfied, halt;
4. Else $\mu_i = \max\{0, \mu_i + \theta_k(\sum_{e \in P} v_i^e - B_i)\}$, $\forall 1 \leq i \leq n; k = k + 1$; goto 2.

The general theory for sub-gradient method states that if $\theta_k \to 0$ and $\sum_{j=1}^k \theta_j \to \infty$ as $k \to \infty$, the above algorithm will converge to the optimal solution [1].

The only problem with this approach is that there may exist a "gap" between MCSP and its corresponding Lagrangian multiplier problem. That is, it is possible that

$$\min\{\sum_{e \in P} 1 : \sum_{e \in P} v_i^e \leq B_i\} > \max L(\mu_1, \ldots, \mu_n).$$

Actually, solving MCSP problem is just one step in the whole scenario of a sequential maze routing. So, it is not necessary to spend too much time to get an optimal solution for MCSP. We will set up an upper bound on the number of iterations. During the iterations, we also record feasible solutions, that is, those paths P such that $\sum_{e \in P} v_i^e \leq B_i$. If the sub-gradient method did not give the optimal solution within those iterations, we will use the shortest one of the recorded feasible paths as our solution.

5 Extensions

In previous sections, to simplify the presentation, we based our work on a simplified crosstalk model and two-terminal nets. Actually, our Lagrangian relaxation based approach is such a robust method that it can be easily extended to multi-terminal nets and more general crosstalk models.

5.1 Multi-terminal nets

For multi-terminal nets, our objective is to find a minimum length Steiner tree route such that the crosstalk on each net is under constraint. But even without considering crosstalk, finding a minimum length Steiner tree is NP-hard [7]. Fortunately, there are many heuristic algorithms exist in the literature [8]. Many of them have proven worst-case upper bounds and produce good results in practice.

Using our Lagrangian relaxation based approach, when routing a multi-terminal net, the crosstalk constraints are also relaxed into the objective function. Instead of a path P, the resulting LSP needs to find a Steiner tree T, such that $\sum_{e \in T} 1 + \sum_{i=1}^{n} \mu_i (\sum_{e \in T} v_i^e - B_i)$ is minimized. Similarly, this can be solved by finding a minimum Steiner tree on a weighted graph. Although this can not be solved exactly, any minimum Steiner tree heuristic on weighted graphs can be used to find an approximate solution.

5.2 More general crosstalk model

In practice, besides coupling capacitance, crosstalk between two nets also depends on their functional and temporal sensitivities [10] and the relative drive strengths of their drivers. Therefore, two nets may not have crosstalk even they are routed closely, and for two adjacent wires, crosstalk induced on one wire may not equal to crosstalk induced on the other. In this case, we only need to change the weight vectors to represent the actual crosstalk induce on each net.

In previous sections, crosstalk is only considered between adjacent wires. That is a very reasonable model since, on one hand, the coupling capacitance decreases rapidly as the distance increases; on the other hand, if there is space between wires, shielding wires can be inserted when necessary. However, if crosstalk between non-adjacent wires must be considered, our approach can also be extended. In this case, the coupling capacitance can be computed as follows.

$$\textit{coupling-capacitance} = \alpha \frac{\textit{length}}{\textit{distance}^{\beta}},$$

where β is a small constant which was estimated to be about 2 in [12].

The only difference between non-adjacent model and the adjacent one is that, in non-adjacency model, we must consider the effect of shielding. That is, when a new net is routed between two previously routed ones, it may shield one of them from the other, and therefore decrease crosstalk on them. This can be illustrated in Figure 5. This effect can be taken into consideration by changing the weights of edges as follows. Suppose edge e lies between routed nets i and j, and c_{ij} is the crosstalk between net i and net j which might be shielded by e. Let c_i and c_j denote the crosstalk between e and nets i and j, respectively. Then we can set $v_i^e = c_i - c_{ij}$ and $v_j^e = c_j - c_{ij}$ instead of $v_i^e = c_i$ and $v_j^e = c_j$. If $c_i < c_{ij}$ or $c_j < c_{ij}$, negative weight elements are introduced. Therefore, Dijkstra's algorithm can not be used. Instead, we can use Bellman-Ford algorithm to compute the shortest path [4].

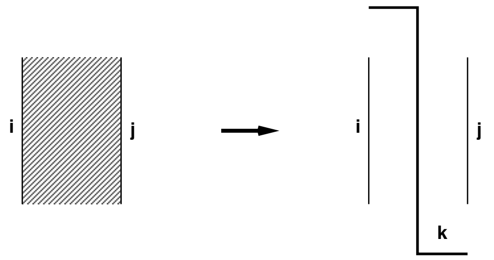

Figure 5: Crosstalk between nets i and j is shielded by a newly routed net k

Note that it can not be guaranteed to find a shortest path in terms of the weighted sum of crosstalk. This is because when a net is routed in an empty U-form as shown in Figure 6, it will shield some segments of itself from other nets. Therefore, the total weight of a path may not be equal to the sum of the weights of its component paths, and a component of an optimal path may not be an optimal one. But since we do not favor such empty U-forms in practice, the little overestimation may help us to avoid it.

6 Experimental results

We implement the Lagrangian relaxation based mazing routing algorithm in C++ on a Sun Sparc 5 workstation. Although our algorithm can be applied to multi-layer routing, the current implementation is only designed for two-layer routing in the HV model (i.e. one layer for horizontal wiring and one layer for vertical wiring). We do the experiments on the MCM benchmarks originated from 1993

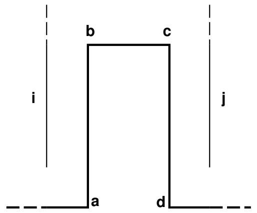

Figure 6: An empty U-form routing shields segment ab from net j, and segment cd from net i

Physical Design Workshop. These benchmarks are composed of multi-terminal nets and are intended for multi-layer routing. In the experiments, we first use the general maze routing (shortest path without considering crosstalk) to route as many nets as possible. This gives us a maximum number of nets which can be routed on a pair of layers. We then measure crosstalk on each net in this routing solution and call it the *base crosstalk* of that net. We systematically set the crosstalk constraint for each net to be a percentage of its base crosstalk and run our algorithm on it using the same net ordering. For comparison, we run once again the general maze routing on those nets. This time, after routing a net, we also compute its crosstalk. If the crosstalk violate the constraints, the current net is removed. We compare the numbers of feasibly routed nets by these two approaches and find that our algorithm can feasibly route more than 30% more nets than the general maze routing in average.

Figure 7 plots one of such experimental results on **mcm1** by setting the constraint of each net respectively to be 70%, 60%, 50%, 40%, and 0% of its base crosstalk.

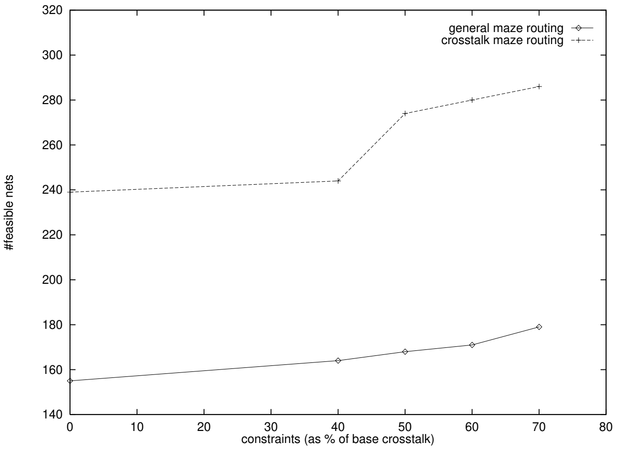

Figure 7: Comparison of the numbers of feasibly routed nets by the two algorithms

References

[1] R.K. Ahuja, T.L. Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

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

[3] K. Chaudhary, A. Onozawa and E.S. Kuh, A Spacing Algorithm for Performance Enhancement and Crosstalk Reduction. *IEEE/ACM International Conference on Computer Aided Design,* 1993.

[4] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms.* The MIT Press, 1990.

[5] T. Gao and C.L. Liu, Minimum Crosstalk Channel Routing. *IEEE/ACM International Conference on Computer Aided Design,* 1993.

[6] T. Gao and C.L. Liu, Minimum Crosstalk Switchbox Routing. *IEEE/ACM International Conference on Computer Aided Design,* 1994.

[7] M.S. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, 1979.

[8] F.K. Hwang, D.S. Richards, and P. Winter, *The Steiner Tree Problem.* North-Holland, 1992.

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

[10] D.A. Kirkpatrick and A.L. Sangiovanni-Vincentelli, Digital Sensitivity: Predicting Signal Interaction using Functional Analysis. *IEEE/ACM International Conference on Computer Aided Design,* 1996.

[11] T. Ohtsuki, Maze-Running and Line-Search Algorithms. In T. Ohtsuki ed., *Layout Design and Verification.* Elsevier Publishers B.V. (North-Holland), 1986.

[12] T. Sakurai and K. Tamaru, Simple formulas for two and three dimensional capacitance. *IEEE Trans. Electronic Devices,* 1993.

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

[14] H. Zhou and D.F. Wong, An Optimal Algorithm for River Routing with Crosstalk Constraints. *IEEE/ACM International Conference on Computer Aided Design,* 1996.