Provably Good Algorithms for Performance-Driven Global Routing

J. Cong, A. Kahng, G. Robins, M. Sarrafzadeh† and C. K. Wong‡

Dept. of Computer Science, UCLA, Los Angeles, CA 90024
† Dept. of EE & CS, Northwestern University, Evanston, IL 60208
‡ IBM T. J. Watson Research Center, Yorktown Heights, NY 10598

Abstract

We propose a provably good performance-driven global routing algorithm for both cell-based and building block design. The approach is based on a new bounded-radius minimum routing tree formulation, which simultaneously minimizes both routing cost and the longest interconnection path, so that both are bounded by small constant *factors away from optimal. For any given value of a parameter ϵ, we construct a routing tree with longest interconnection path length at most $(1+\epsilon)\cdot R$, and with cost at most $(1+\frac{2}{\epsilon})$ times the minimum spanning tree weight, where R is the minimum possible length from the source to the farthest sink. Our method generalizes to Steiner global routing in arbitrary weighted graphs, and to the case where varying wirelength bounds are prescribed for different source-sink paths. Extensive simulations confirm the utility of the approach.*[1]

1 Introduction

With progress in VLSI fabrication technology, interconnection delay has become increasingly significant in determining circuit speed. Recently, it has been reported that interconnection delay contributes up to 70% of the clock cycle in the design of dense, high performance circuits [5] [23]. Due to this trend, performance-driven layout design has received increased attention in the past several years.

Most of the work in this area has focused on the timing-driven placement problem. The so-called zero-slack algorithm was proposed in [10]; fictitious facilities and floating anchors methods were used in [19], and a linear programming approach was used in [14] and [15]. Several other approaches, including simulated annealing, have also been studied [5] [17] [23].

While such techniques have been developed for timing-driven placement, only limited progress has been reported for the timing-driven interconnection problem. In [7], net priorities are determined based on static timing analysis. [18] outlines a hierarchical approach to timing-driven routing. In [20], a timing-driven global router based on the A* heuristic search algorithm was proposed for building-block design. However, these results do not provide a general formulation of the timing-driven global routing problem. Moreover, these solutions are not flexible enough to provide any *trade-off* between interconnection delay and routing cost.

Recently [2] proposed a new model of timing-driven global routing for cell-based design, based on the idea of finding minimum spanning trees with bounded radius. An algorithm was given which constructs a spanning tree with radius $(1+\epsilon)\cdot R$ using an analog of the classical Prim minimum spanning tree (MST) construction; R is the minimum possible tree radius and ϵ is a non-negative user-specified parameter. Such an approach offers a very natural, smooth trade-off between the tree radius (maximum signal delay) and the tree cost (total interconnection length). However, the worst-case total wire cost using this method can be an unbounded factor times optimal.

We therefore now propose a new method for timing-driven global routing which is based on a *provably good* algorithm that *simultaneously* minimizes both total wirelength and maximum delay. More specifically, given a positive real parameter ϵ and a set of terminals, our method produces a routing tree with radius at most $(1+\epsilon)\cdot R$, and with total cost at most $(1+\frac{2}{\epsilon})\cdot$ times the MST cost. In other words, both the total wirelength and the maximum delay of the routing are simultaneously bounded within small *constant* factors of their optimal values. Our method generalizes to arbitrary weighted graphs, and also to Steiner routing formulations. Exploiting underlying geometry allows further improvement of the performance bounds, and moreover our method generalizes to the case where different paths have different length constraints. This series of results is especially surprising since constructing an MST with bounded diameter in a general graph is NP-complete [11], as is the Steiner problem in graphs [9].

Our construction can minimize either total wirelength (a minimum spanning tree), or the longest source-sink path (a minimum delay, or minimum radius, tree), depending respectively on whether we set $\epsilon = \infty$ or $\epsilon = 0$. Between these two extremes, we achieve a continuous, smooth tradeoff. In practice, our algorithm exhibits very good empirical performance

[1] This research was supported in part by NSF grants MIP-9110511 and MIP-9110696, and by an IBM Graduate Fellowship.

0-7803-0593-0/92 $3.00 1992 IEEE

which confirms this smooth tradeoff between the competing requirements of minimum delay and minimum total wirelength. In the following discussion, all proofs have been omitted for brevity; see [3] for details.

2 The Problem Formulation

A signal net N is a set of terminals, with one terminal $s \in N$ a designated source and the remaining terminals sinks. We define the underlying *routing graph* to be a connected weighted graph $G = (V, E)$. A net is a subset of the nodes in this graph. A routing solution of a net N is a tree in G, which we call the *routing tree* of the net, connecting all the terminals in N.

Since the routing tree may be treated as a distributed RC tree, we may use the first-order moment of the impulse response (also called Elmore's delay) to approximate interconnection delay [8] [13]. However, although both the formula for Elmore's delay and its approximations [13] are very useful for simulation or timing verification, they involve sums of quadratic terms and are difficult to compute and optimize during the layout design process. Thus, a linear RC model is often used to derive a simpler approximation for interconnection delay (e.g., [17] [22] [2]). Here we shall also use wirelength to approximate interconnection delay in the construction of routing solutions. In practice, a subsequent iterative improvement step, based on a more accurate RC delay model, may be used to enhance the routing solutions.

We say that the *cost* of a path in G is the sum of the edge weights in the path. A *shortest path* in G between two terminals $x, y \in N$, denoted by $minpath_G(x, y)$, is a path connecting x and y with minimum cost. For a weighted graph G we use $dist_G(x, y)$ to denote the cost of $minpath_G(x, y)$.

Definition: The *radius* R of a signal net is the maximum cost of any shortest source-sink path.

Definition: The *radius* $r(T)$ of a routing tree T is the cost of a path in T from the source s to the furthest sink. Clearly, $r(T) \geq R$ for any routing tree T.

According to the linear RC delay model, we minimize the interconnection delay of a net by minimizing the radius of the routing tree, which measures the maximum interconnection delay between the source and any sink. On the other hand, we also want a routing tree with small total wirelength. Without this latter consideration, we could simply use Dijkstra's *shortest path tree* (SPT) of the net [21]. Although the SPT has the smallest possible radius of any routing tree, there exist examples where the cost of the SPT can be $\Omega(|N|)$ times greater than the MST cost [2], and such high cost may increase the overall routing area and interconnection delay. To simultaneously consider both the radius and the cost in the routing tree construction, we formulate the timing-driven global routing problem as follows:

The Bounded Radius Minimum Routing Tree (BRMRT) Problem: Given a parameter $\epsilon \geq 0$ and a signal net with radius R, find a minimum-cost routing tree T with radius $r(T) \leq (1 + \epsilon) \cdot R$.

The parameter ϵ controls the trade-off between the radius and the cost of the tree. When $\epsilon = 0$, we minimize the radius of the routing tree and thus obtain an SPT for the signal net; on the other hand, when $\epsilon = \infty$ we minimize the total cost of the tree and obtain an MST. In general, as ϵ grows, there is less restriction on the radius, allowing further reduction in tree cost.

3 Bounded-Radius MST Routing

The basic idea of our provably good bounded-radius MST algorithm is to construct a subgraph Q which spans N and has both small total cost and small radius. Thus, computing the SPT of Q will yield a routing tree with small cost and radius, and will correspond to a good routing solution. For a routing graph $G = (V, E)$ with $V = N$, our algorithm is as follows:

- Compute the shortest path tree SPT_G of G, and the minimum spanning tree MST_G of G. Initialize the graph Q to be equal to MST_G.
- Let L be the sequence of vertices corresponding to a depth-first tour of MST_G, where each edge of MST_G is traversed exactly twice (Figure 1).
- Traverse L while keeping a running total S of traversed edge costs. As this traversal reaches each node L_i, check whether S is greater than $\epsilon \cdot dist_G(s, L_i)$. If so, reset S to 0 and merge $minpath_G(s, L_i)$ into Q. Continue traversing L while repeating this process.
- Our final routing tree is SPT_Q.

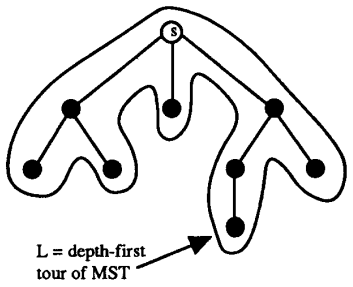

Figure 1: An MST and its depth-first tour.

The formal description of the algorithm is given in Figure 2. It is not difficult to show that for any fixed ϵ this algorithm produces a routing tree with radius and total cost simultaneously bounded by small constants times optimum.

Theorem: For any weighted graph G and parameter ϵ, the routing tree T constructed by our algorithm has radius $r(T) \leq (1 + \epsilon) \cdot R$, and total wirelength given by $cost(T) \leq (1 + \frac{2}{\epsilon}) \cdot cost(MST_G)$.

BRBC: computing a bounded-radius bounded-cost spanning tree

Input: $G = (V, E)$ (with radius R, source $s \in V$), ϵ
Output: a spanning tree of radius $\leq (1+\epsilon) \cdot R$ and cost $\leq (1 + \frac{2}{\epsilon}) \cdot \text{cost}(\text{MST}_G)$

compute MST_G and SPT_G
$Q = MST_G$
L = depth-first tour of MST_G
$S = 0$
for $i = 1$ to $|L| - 1$
  $S = S + cost(L_i, L_{i+1})$
  if $S \geq \epsilon \cdot dist_G(s, L_{i+1})$ **then**
    $Q = Q \cup minpath_G(s, L_{i+1})$
    $S = 0$
Output T = shortest path tree of Q

Figure 2: Computing a bounded-radius spanning tree T for $G = (V, E)$, with source $s \in V$ and radius R, using parameter ϵ.

Because our method yields a bounded-radius, bounded-cost routing tree, we call this the BRBC algorithm. A similar idea was recently used in the distributed computation literature in [1] for constructing spanning trees with small diameter and small weight.

4 Bounded-Radius Steiner Routing

For building-block design the underlying routing graph is based on the channel intersection graph [4], and a net N is a subset of the vertices of G. Here the BRMRT problem is actually the **Bounded Radius Optimal Steiner Tree** (BROST) problem, and the channel intersection points (i.e., nodes in $V - N$) are otential Steiner points. Since the BROST problem is NP-complete, we need a minimum-cost tree spanning N (i.e., a Steiner tree) within G.

Recall that in applying the BRBC algorithm to general graphs, the only reason we use the MST is to obtain a reasonably short tour of the vertices. Given any approximate Steiner tree $\hat{T}$, we can construct a routing tree with radius within $(1+\epsilon) \cdot r(\hat{T})$, and with cost within $(1 + \frac{2}{\epsilon}) \cdot cost(\hat{T})$. Our algorithm uses a heuristic [16] [24] to build a Steiner tree $\hat{T}$ in the underlying routing graph having cost within a factor 2 of optimal.

We construct and traverse the depth-first tour of $\hat{T}$, adding to $\hat{T}$ the shortest paths from the source to the appropriate vertices of the tour. Finally, we compute the SPT in the resulting graph and output the union of the shortest paths from the source to all terminals in N (which includes intermediate non-terminal nodes on the shortest paths as Steiner points). Thus the cost of the tour will be at most 4 times the *optimal* Steiner tree (T_{opt}) cost, and the resulting routing tree cost is at most $2 \cdot (1 + \frac{2}{\epsilon})$ times optimal.

5 Further Extensions

If we are routing in a metric space and are allowed to introduce arbitrary Steiner points to reduce the routing cost/diameter, we can slightly modify the basic algorithm to introduce Steiner points on the tour L whenever $S = 2\epsilon \cdot R$. From each of these Steiner points we construct shortest paths to the source and add them to Q as in the original algorithm. Thus, each node in the traversal of L will be within $\epsilon \cdot R$ of a Steiner point, i.e., within $(1 + \epsilon) \cdot R$ of the source. In this case, we can show that the same radius bound is maintained, while the cost of the routing tree will be bounded by $2 \cdot (1 + \frac{1}{\epsilon}) \cdot cost(T_{opt})$.

In addition, well-known results which bound the $\frac{MST}{Steiner}$ ratio in various geometries [12] [6] can be used with the above scheme to yield even better bounds whenever the edge weights correspond to a metric. For example, in the Manhattan plane, it is easy to show that the cost of the tree produced by our algorithm will be bounded by $\frac{3}{2} \cdot (1 + \frac{1}{\epsilon})$ times optimal, while in the Euclidean plane, the tree cost will be bounded by $\frac{2}{\sqrt{3}} \cdot (1 + \frac{1}{\epsilon})$ times optimal. Our method readily generalizes to accomodate different length constraints ϵ_i for different terminals.

6 Experimental Results

The algorithms were implemented in ANSI C; code is available from the authors upon request. The BRBC algorithm for spanning tree routing was tested on a large number of random nets generated from a uniform distribution in the grid. Results are summarized in Figure 3, which clearly shows the tradeoff between routing cost and maximum delay. As ϵ decreases, both the cost and radius curves shift monotonically from that of the MST to that of the SPT.

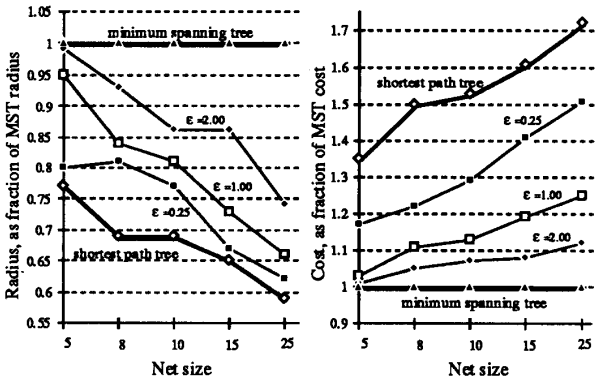

Figure 3: The smooth tradeoff between total routing cost (right) and maximum signal delay (left) produced by the BRBC algorithm. The performance envelope lies between the SPT and the MST, and the parameter ϵ determines the exact tradeoff.

The BRBC algorithm for Steiner tree routing was tested on the channel intersection graphs of random block layouts in the grid; these were generated by adding a fixed number of non-overlapping blocks, with length, width and lower-left coordinates all uniformly distributed. The simulations confirm the tradeoffs inherent in the bounded-radius routing approach. In practice, the efficiency of implementation and the provably good output provide compelling reasons to adopt the BRBC algorithm.

7 Conclusions

We have proposed a new, provably good general algorithm for timing-driven global routing. This method is based on a routing tree construction where *both* the total wirelength and the maximum delay of the routing are bounded by *constant* factors away from optimal. Our approach readily extends to Steiner tree routing in arbitrary weighted graphs, and to the case where distinct wirelength bounds are prescribed for different source-sink paths. Extensive simulations confirm that our approach gives very good performance, and indeed exhibits a smooth tradeoff between the competing requirements of minimum delay and minimum total wirelength.

Based on our methods for constructing bounded-radius routing trees, the global routing procedure will work as follows. We route all nets, one by one, according to their priorities. For each net, we construct a bounded-radius MST or bounded-radius minimum Steiner tree. The parameter ϵ is either given by the user or computed based on an estimation of the timing constraints for the net. Different values of ϵ_i can be used within a single net to reflect timing constraints in various input-output paths. The cost of each edge in the routing graph is a function of wirelengths, channel capacities, and the distribution of current channel densities. After routing each net, we update the edge costs in the routing graph. After all nets are routed, we may compute the timing-critical paths and, if necessary, further reduce the interconnection delay by rerouting some critical nets based on more accurate distributed RC delay models.

There are several remaining open problems, such as determining the complexity of computing the minimum cost bounded-radius spanning tree in the Manhattan plane, or the complexity of choosing an MST with minimum radius when the MST is not unique.

References

[1] B. Awerbuch, A. Baratz, and D. Peleg, *Cost-sensitive analysis of communication protocols*, in Proc. ACM Symp. on Principles of Distributed Computing, 1990, pp. 177–187.

[2] J. Cong, A. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, *Performance-driven global routing for cell based ic's*, in Proc. IEEE Intl. Conf. on Computer Design, Cambridge, MA, October 1991, pp. 170–173.

[3] J. Cong, A. B. Kahng, and G. Robins, *Performance-driven global routing for cell based ic's*, Tech. Rep. CSD-TR-910013, University of California, Los Angeles, Computer Science Department, April 1991.

[4] W. M. Dai, T. Asano, and E. S. Kuh, *Routing region definition and ordering scheme for building-block layout*, IEEE Trans. on CAD, CAD-4 (1985), pp. 189–197.

[5] W. E. Donath, R. J. Norman, B. K. Agrawal, and S. E. Bello, *Timing driven placement using complete path delays*, in Proc. IEEE Design Automation Conf., 1990, pp. 84–89.

[6] D. Z. Du and F. K. Hwang, *A proof of gilbert-pollak's conjecture on the steiner ratio*, in Proc. IEEE Symp. on Foundations of Computer Science, 1990.

[7] A. E. Dunlop, V. D. Agrawal, D. Deutsch, M. F. Jukl, P. Kozak, and M. Wiesel, *Chip layout optimization using critical path weighting*, in Proc. IEEE Design Automation Conf., 1984, pp. 133–136.

[8] W. C. Elmore, *The transient response of damped linear networks with particular regard to wide-band amplifiers*, J. Appl. Phys., 19 (1948), pp. 55–63.

[9] M. Garey and D. S. Johnson, *The rectilinear steiner problem is np-complete*, SIAM J. Applied Math., 32 (1977), pp. 826–834.

[10] P. S. Hauge, R. Nair, and E. J. Yoffa, *Circuit placement for predictable performance*, in Proc. IEEE Intl. Conf. on Computer-Aided Design, Santa Clara, CA, November 1987, pp. 88–91.

[11] J. Ho, D. T. Lee, C. H. Chang, and C. K. Wong, *Bounded-diameter spanning trees and related problems*, in Proc. ACM Symp. on Computational Geometry, 1989, pp. 276–282.

[12] F. K. Hwang, *On steiner minimal trees with rectilinear distance*, SIAM J. Applied Math., 30 (1976), pp. 104–114.

[13] P. P. J. Rubinstein and M. A. Horowitz, *Signal delay in rc tree networks*, IEEE Trans. on CAD, 2 (1983), pp. 202–211.

[14] M. A. B. Jackson and E. S. Kuh, *Performance-driven placement of cell-based ic's*, in Proc. IEEE Design Automation Conf., 1989, pp. 370–375.

[15] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, *Clock routing for high-performance ic's*, in Proc. IEEE Design Automation Conf., 1990, pp. 573–579.

[16] L. Kou, G. Markowsky, and L. Berman, *A fast algorithm for steiner trees*, Acta Informatica, 15 (1981), pp. 141–145.

[17] I. Lin and D. H. C. Du, *Performance-driven constructive placement*, in Proc. IEEE Design Automation Conf., 1990, pp. 103–106.

[18] E. S. K. M. A. B. Jackson and M. Marek-Sadowska, *Timing-driven routing for building block layout*, in Proc. IEEE Intl. Symp. on Circuits and Systems, 1987, pp. 518–519.

[19] M. Marek-Sadowska and S. P. Lin, *Timing driven placement*, in Proc. IEEE Intl. Conf. on Computer-Aided Design, 1989, pp. 94–97.

[20] Y. Ogawa, M. Pedram, and E. S. Kuh, *Timing-driven placement for general cell layout*, in Proc. Intl. Symp. on Circuits and Systems, 1990, pp. 872–876.

[21] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice-Hall, 1982.

[22] P. Ramanathan and K. G. Shin, *A clock distribution scheme for non-symmetric vlsi circuits*, in Proc. IEEE Intl. Conf. on Computer-Aided Design, 1989, pp. 398–401.

[23] S. Sutanthavibul and E. Shragowitz, *An adaptive timing-driven layout for high speed vlsi*, in Proc. IEEE Design Automation Conf., 1990, pp. 90–95.

[24] Y. F. Wu, P. Widmayer, and C. K. Wong, *A faster approximation algorithm for the steiner problem in graphs*, Acta Informatica, 23 (1986), pp. 223–229.