

N Topo.	<i>TL</i>	<i>TPL</i>	<i>D</i>	<i>MD</i>	<i>AMD</i>
4 1-s	1.21	7.87	1.05	1.38	1.26
mdat	1.24	8.07	.99	1.39	1.28
mcmdat	1.21	7.87	.99	1.36	1.25
8 1-s	1.99	41.85	1.54	2.47	2.01
mdat	2.11	42.60	1.28	2.40	2.07
mcmdat	2.05	40.67	1.28	2.32	1.98
16 1-s	2.96	203.71	1.99	4.13	3.05
mdat	3.21	191.20	1.48	3.78	3.01
mcmdat	3.13	185.87	1.48	3.67	2.91

Table 1: Comparison of routing results on 0.5μ CMOS IC technology.

N Topo.	<i>TL</i>	<i>TPL</i>	<i>D</i>	<i>MD</i>	<i>AMD</i>
4 1-s	12.16	78.78	10.52	11.60	8.84
mdat	12.46	80.70	9.98	11.44	9.05
mcmdat	12.17	78.79	9.98	11.19	8.74
8 1-s	19.94	418.55	15.49	28.81	18.90
mdat	21.18	426.00	12.83	25.66	18.84
mcmdat	20.51	406.74	12.83	25.07	17.72
16 1-s	29.69	2037.12	19.99	56.69	36.33
mdat	32.13	1912.05	14.87	50.08	33.23
mcmdat	31.34	1858.79	14.87	49.19	31.89

Table 2: Comparison of routing results on MCM technology.

Experimental results are shown in Tables 1 and 2. We compare tree length (*TL*), the sum total path length for all $p_i \rightarrow p_j$ pairs (*TPL*), and tree diameter *D*. Also included are the maximum delay between any source-sink pair (*MD*) and the average of maximum delays for each source (*AMD*), averaged over all runs. Delays were measured from the input transition to the output reaching 90% of its final value. Delays are shown in nanoseconds, while tree and path lengths are shown in centimeters.

Table 3 shows the solutions by 1-Steiner, MD A-tree, and MC MD A-tree when only a (randomly chosen) subset of pairs are critical. This table assumes the CMOS IC parameters, and 8 points per test set. We report the total weighted path length *TWPL*, the required weighted path length *RWPL* (a lower bound), and the maximum and average maximum delays for the critical pairs.

Compared with the best known Steiner tree heuristic, minimum cost minimum diameter A-trees offer maximum delay improvements of 1% to 11% and 4% to 13% for 0.5μ CMOS ICs and MCMs. Average maximum delays were improved by 0% to 4% and 1% to 12%. When only a subset of point pairs are critical, maximum delay improvements for 0.5μ CMOS IC routings were from 3% to 7%.

Pairs Topo.	<i>TWPL</i>	<i>RWPL</i>	<i>MD</i>	<i>AMD</i>
10 1-s	7.55	6.64	2.99	1.80
mdat	7.70		3.03	1.91
mcmdat	7.26		2.90	1.83
20 1-s	15.08	13.14	3.01	1.81
mdat	15.32		2.90	1.90
mcmdat	14.56		2.80	1.82
30 1-s	22.61	19.64	2.97	1.83
mdat	22.97		2.96	1.93
mcmdat	21.91		2.78	1.85

Table 3: Comparison of routing results for test cases where only a limited number of point pairs have $W(p_i, p_j) = 1$.

REFERENCES

- [1] C. J. Alpert, T. C. Hu, J. H. Huang, and A. B. Kahng, "A Direct Combination of the Prim and Dijkstra Constructions for Improved Performance-Driven Routing," *Proc. IEEE Int'l Symp. on Circuits and Systems*, pp. 1869-1872, 1993.
- [2] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, "Rectilinear Steiner Trees with Minimum Elmore Delay," *Proc. 31st ACM/IEEE DAC*, pp. 381-386, 1994.
- [3] K. D. Boese, A. B. Kahng, and G. Robins, "High-Performance Routing Trees With Identified Critical Sinks," *Proc. ACM/IEEE DAC*, pp. 182-187, 1993.
- [4] T.-H. Chao, Y.-C. Hsu, J.-M. Ho, K. D. Boese, and A. B. Kahng, "Zero Skew Clock Routing with Minimum Wire-length," *IEEE Trans. on Circuits and Systems*, Vol. 39, No. 11, pp. 799-814, November 1992.
- [5] J. P. Cohoon and L. J. Randall, "Critical Net Routing," *Proc. IEEE Int'l Conf. on Computer Design*, pp. 174-177, 1991.
- [6] J. Cong, A. B. Kahng, G. Robins, and M. Sarrafzadeh, "Provably Good Performance-Driven Global Routing," *IEEE Trans. Computer Aided Design*, Vol. 11, No. 6, pp. 739-752, June 1992.
- [7] J. Cong and K.-S. Leung, "Optimal Wiresizing Under the Distributed Elmore Delay Model," *Proc. Int'l Conf. on Computer-Aided Design*, pp. 110-114, 1993.
- [8] J. Cong, K.-S. Leung, and D. Zhou, "Performance-Driven Interconnect Design Based on Distributed RC Delay Model," *Proc. 30th ACM/IEEE DAC*, pp. 606-611, 1993.
- [9] J. Cong and P. H. Madden, "Performance Driven Routing with Multiple Sources" *UCLA Computer Science Department Technical Report #950002*, January 1995.
- [10] J.-M. Ho, D. T. Lee, C.-H. Chang, and C. K. Wong "Bounded-Diameter Minimum Spanning Trees and Related Problems," *Computational Geometry Conference*, pp. 276-282, 1989.
- [11] X. Hong, T. Xue, E. S. Kuh, C. K. Cheng, and J. Huang, "Performance-Driven Steiner Tree Algorithms for Global Routing," *Proc. ACM/IEEE DAC*, pp. 177-181, 1993.
- [12] A. B. Kahng and G. Robins, "A New Family of Steiner Tree Heuristics with Good Performance: The Iterated 1-Steiner Approach," *Proc. IEEE Int. Conf. Computer Aided Design*, pp. 428-431, 1990.
- [13] D. Zhou, S. Su, F. Fsu, D. S Gao, and J. Cong, "A Simplified Synthesis of Transmission Lines with a Tree Structure," *Journal of Analog Integrated Circuits and Signal Processing (Special Issue on High-Speed Interconnects)*, pp. 19-30, January 1994, Vol. 5, No. 1.

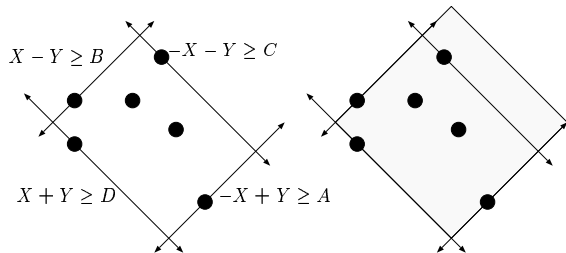


Figure 1: The smallest tilted rectangle containing the points, and a smallest tilted square which contains the rectangle.

D. Minimum Cost Minimum Diameter A-Tree Heuristic

In fact, it is not always necessary to place the root of the A-tree at the center of an *STS*. It is easy to see that as long as the root satisfies the constraint $d(p_i, r) + d(r, p_j) \leq D$ for all p_i and p_j , the tree will have minimum diameter.

In the Euclidean plane, the region which satisfies the constraint is simply the intersection of all ellipses formed by pairs p_i, p_j .

We define an *octilinear segment* to be a segment that is either horizontal, vertical, or have slope ± 1 . In the Manhattan plane, the set of points satisfying $d(p_i, r) + d(r, p_j) \leq D$ is an *octilinear ellipse* (*OE*), bounded by no more than eight octilinear segments. If $d(p_i, p_j) = D'$, the *OE* contains the bounding box of the points with a “fringe” of $\frac{D-D'}{2}$. An example is shown in Figure 2.

It can be shown that the intersection of two *OEs* is an *octilinear region* (*OR*), i.e. a region defined by eight octilinear segments. We represent each of these segments with a linear inequality.

Without loss of generality, let $aX + bY \geq C_1$ be an equation from *OR* R_i , and $aX + bY \geq C_2$ from *OR* R_j ($a, b \in \{0, 1, -1\}$, with at least one being non-zero). Then $aX + bY \geq \max(C_1, C_2)$ defines an octilinear segment (possibly of zero length) of the *OR* $R_i \cap R_j$.

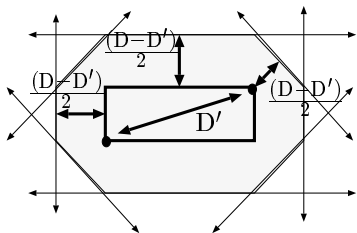


Figure 2: An *octilinear region* on the Manhattan plane, the set of points which satisfy $d(p_i, r) + d(r, p_j) \leq D$.

In order to determine the set of points which may serve as the center of a minimum diameter tree, we find the intersection of all *OEs* for the point pairs. There are $O(n^2)$ point pairs, resulting in the same number of *OEs*. It can be shown that the intersection of any number of *OEs* can be represented by a single *OR*. A linear time construction of the feasible region is also possible[9]. The shaded *OR* in Figure 3 shows the feasible region for a minimum diameter A-tree center for the given point set.

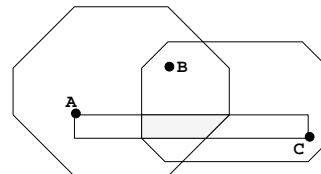


Figure 3: The feasible region for the center of a minimum diameter A-tree, the intersection of a number of octilinear ellipses.

After construction of the feasible region for a set of root points, any tree which has a shortest path from a point in the region to each point p_i will be a minimum diameter tree. As the A-tree algorithm is a heuristic, we have no exact method of determining the “optimal” root location from within the feasible region. To address this, we simply construct a set of candidate roots, and evaluate the trees produced using each.

The set of candidate roots are the corner points of the feasible region, the Hanan grid points contained within the feasible region, and the intersections of the Hanan grid lines with the feasible region. In general the number of candidates is small. The selected A-Tree is the one which minimizes tree length, with weighted total path length used to break ties.

IV. EXPERIMENTAL RESULTS

We used HSPICE to evaluate the performance of each topology, using a nominal 0.5μ transistor model supplied by MCNC. Tuned pMOS and nMOS transistors were used for drivers in the IC and MCM experiments (sized as $10\mu \times 0.5\mu$: $9.5\mu \times 0.5\mu$, and $200\mu \times 0.5\mu$: $170\mu \times 0.5\mu$ respectively). IC test sets were routed on a $1cm \times 1cm$ area, while MCM tests were on a $10cm \times 10cm$ area. Tree edges were simulated using the parameters of [2].

We compare the results of constructing trees by the Minimum Diameter A-tree heuristic (mdat), Minimum Cost Minimum Diameter A-tree heuristic (mcmdat), and the 1-Steiner (1-s) algorithm described in [12]. 100 randomly generated test sets of 4, 8, and 16 points (N) were used, with $W(p_i, p_j) = 1$ for all point pairs.

delay model, the distributed Elmore delay model [7], or calculated using SPICE.

For any two points, we define the distance $d(p_i, p_j)$ between two nodes p_i and p_j as their Manhattan distance. Given a tree T , we define the distance between nodes p_i and p_j in T as $d_T(p_i, p_j)$, the sum of edge lengths along the unique path between the points. The *diameter* D of tree T is defined as the maximum $d_T(p_i, p_j)$ over all pairs p_i, p_j .

Note that if the weights of all pairs are zero, the PD-MSR problem becomes the classical minimum Steiner tree problem, which is NP-hard. Therefore, the PD-MSR problem is also NP-hard.

When the delay bound of each pair of timing-critical nodes is given, one can also formulate the *constrained multiple source routing problem* as finding a Steiner routing tree which satisfies the delay constraint between every timing-critical pair and minimizes the total tree length.

III. MINIMUM DIAMETER A-TREE ALGORITHM FOR PD-MSR PROBLEM

Our approach to the PD-MSR problem is to construct a minimum-cost (in terms of tree length) minimum-diameter (MCMD) Steiner tree for connecting the given set of points on the Manhattan plane. That is, we want to construct a minimum-diameter Steiner tree whose cost is minimum. Such a tree minimizes the maximum path-length between any pair of nodes in the tree and uses the minimum wire length. Thus, it in general leads to a good solution to the PD-MSR problem, especially when many or all pairs of nodes are timing-critical.

The results in [10] showed that the MCMD Steiner tree (or spanning tree) problem for arbitrary graphs is NP-hard. The complexity is unknown if one wants to construct an MCMD Steiner tree on the Euclidean plane or Manhattan plane. A linear time algorithm was presented in [10] for constructing a minimum diameter Steiner tree in the Euclidean plane, but no effort was made to minimize the cost of the resulting tree.

In the remainder of this section, we present two efficient heuristic algorithms for computing the MCMD Steiner trees on the Manhattan plane, based on constructing a *minimum diameter A-tree*. Our algorithms guarantee the minimum diameter and try to minimize the tree cost as much as possible.

There are three steps to the construction of these trees: determination of the minimum possible diameter of the tree, identification of a center point to serve as the root of the A-tree, and construction of the A-tree.

A. Review of A-Tree Algorithm

In [8], the authors presented an algorithm for the construction of “A-trees,” shortest path Steiner trees rooted

at a source, and having minimal wire length.

A-trees are constructed by starting with a forest of nodes, and then iteratively merging subtrees until all components are connected. The merges progress such that new edges form shortest paths to the root. The A-tree algorithm applies three “safe” merge moves which preserve length optimality, and two heuristic moves in the event of ties. On average, it was shown that 94% of merge moves were optimal, and the trees constructed were within 4% of the optimal length.

In an A-tree T rooted at point r , $d_T(p_i, r) = d(p_i, r)$, as the routing within the tree is guaranteed to be a shortest path.

B. Minimum Tree Diameter

It was shown in [10] that the smallest enclosing circle for a set of points in the Euclidean plane also gives the minimum diameter for a tree connecting those points. A similar property holds for the Manhattan plane.

We define a *tilted rectangle (TR)* as a region defined by a rectangle with sides at 45 degree angles with respect to the axis. Such a region may be defined by a set of four equations, $-X + Y \geq A$, $X - Y \geq B$, $-X - Y \geq C$, and $X + Y \geq D$, for some set of constants A, B, C , and D .

In the Manhattan plane, the analog of the Euclidean circle is the *tilted square (TS)*, the set of points p such that $d(p, c) \leq D/2$ for some center point c and diameter D . A *TS* is a special case of a *TR*, where the distances between opposite edges is equal.

The *smallest tilted rectangle (STR)* and *smallest tilted square (STS)* are the smallest tilted rectangle and square to enclose a set of points P respectively. Clearly, the *STS* contains the *STR*.

The diameter of the *STS* for point set P gives the minimum tree diameter of any Steiner tree connecting the points in P .

For the Manhattan plane, an *STS* can easily be found in linear time. We first find the *STR* enclosing the points by examining each point, adjusting the A, B, C , and D values of the defining equations as required. We then adjust the values again (increasing the area the minimum possible) to produce an *STS* covering the points. Note that the *STS* is not necessarily unique. See Figure 1 for an example.

C. Minimum Diameter A-Tree Heuristic

The first of the two heuristics simply computes the *STS* for the point set P , and then constructs an A-tree rooted at the center. Since the A-tree has a shortest path from all points to the root, all edge lengths from the center to a point will have length less than $D/2$, and the tree has minimum diameter.

Performance Driven Routing with Multiple Sources*

Jason Cong and Patrick H. Madden
UCLA Computer Science Department
Los Angeles, CA 90024
cong@cs.ucla.edu pickle@cs.ucla.edu

ABSTRACT

Existing routing problems for delay minimization consider the connection of a *single* source node to a number of sink nodes, with the objective of minimizing the delay from the source to all sinks, or a set of critical sinks. In this paper, we study the problem of routing nets with multiple sources, such as those found in signal busses. This new model assumes that each node in a net may be a source, a sink, or both. The objective is to optimize the routing topology to minimize the total weighted delay between *all* node pairs (or a subset of critical node pairs). We present two heuristic algorithms for the multiple-source performance-driven routing problem based on efficient construction of minimum-diameter minimum-cost Steiner trees. Experimental results for sub-micron IC technology show as much as an 11% reduction in the maximum interconnect delay, while MCM results show as much as a 13% reduction, when compared to conventional minimum Steiner tree based routing algorithms.

I. INTRODUCTION

Interconnect delay minimization is an important objective in the design of high-performance systems. Recent studies show that the interconnect delay can be reduced considerably by interconnect topology optimization when the resistance ratio, i.e. the driver resistance versus unit wire resistance, is small in the design [8]. Since the scaling of MOS technology decreases the resistance ratio (as shorter transistor channel length leads to smaller driver resistance and smaller wire width leads to larger unit wire resistance), interconnect topology optimization is becoming an increasingly important problem.

A number of interconnect topologies have been proposed for interconnect performance optimization, including bounded-radius bounded-cost trees[6], AHHK trees[1], maximum performance trees[5], A-trees[8], low-delay trees[3], and IDW/CFD trees[11]. Although many of these methods effectively reduce the interconnect delay,

all of them assume that there is a single source node driving one or more sink nodes and minimize the delay from the unique source to all sinks, or a set of critical sinks.

In practice, many timing-critical nets may have multiple sources, each of them controlled by a tri-state gate and driving the net at a different time, such as those often found in signal busses. In this case, the existing performance-driven routing algorithms for single source nets may perform poorly, as a topology optimized for one source may result in long interconnect delay when some other source becomes active.

In this paper, we study the problem of routing nets with multiple sources. This new model assumes that each node in a net may be a source, a sink, or both. The objective is to optimize the routing topology to minimize the total weighted delay between *all* node pairs, where the weight between a node pair indicates the priority of delay minimization between the pair. We present two algorithms for the performance-driven multiple source routing problem based on the construction of *minimum diameter A-trees*.

II. PROBLEM FORMULATION

Given a set of points $P = \{p_1, p_2, \dots, p_n\}$ on the Manhattan plane, and a non-negative weight $W(p_i, p_j)$ as the *path weight* between each pair p_i and p_j to indicate the timing criticality between this pair of points, the *performance-driven multiple source routing (PD-MSR) problem* is defined as finding a Steiner tree T which connects all the given points and minimizes the following two objectives:

- Total weighted delay $WD(T)$ between pairs of nodes p_i and p_j . i.e. $WD(T) = \sum_{p_i, p_j} W(p_i, p_j) \times delay(p_i, p_j)$.
- Total tree length $L(T)$, defined as the sum of the lengths of each tree edge.

We assume that the first objective has higher priority than the second one. For simplicity, one may assume that $W(p_i, p_j) \in [0, 1]$, i.e. non-critical pairs of points have weight zero and critical pairs have weight one. The delay between a pair of points, $delay(p_i, p_j)$, may be estimated using an appropriate model, such as the linear

*This work is partially supported by ARPA/CSTO under Contract J-FBI-93-112, NSF Young Investigator Award MIP9357582, and a grant from Intel Corporation.