Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596

PERFORMANCE-DRIVEN GLOBAL ROUTING FOR CELL BASED IC'S

Jason Cong
Andrew Kahng
Gabriel Robins

December 1990
CSD-900052

Performance-Driven Global Routing for Cell Based IC's

Jason Cong, Andrew Kahng and Gabriel Robins

UCLA Department of Computer Science
Los Angeles, California 90024

Abstract

Advances in VLSI technology and the increased complexity of circuit designs cause performance to become an increasingly important constraint for layout. In this paper, we address the issue of delay optimization during the global routing phase. We formulate this problem as the construction of a bounded-radius spanning tree for a given pointset in the plane, and present a family of effective heuristics. Our approach has very good empirical performance with respect to total wirelength, and can be smoothly tuned between the competing requirements of minimum delay and minimum total netlength. We present extensive computational results which confirm this. Extensions can be made to the graph and Steiner versions of the problem, and a number of open problems are described.

1 Introduction

As VLSI fabrication technology advances, interconnection delay becomes increasingly significant in determining overall circuit speed. Recently, it has been reported that interconnection delay contributes up to 50% to 70% of the clock cycle in the design of very dense and high performance circuits [1] [15]. Thus, with submicron device dimensions and nearly a million transistors integrated on a single microprocessor, on-chip and chip-to-chip interconnections are playing a major role in determining the performance of digital systems.

Due to this trend, performance-driven layout design has received considerable attention in the past few years. However, most of the work in this area has been on the timing-driven placement problem. A number of methods have been developed to generate good placements wherein the blocks or cells in timing-critical paths are placed close together. The so-called zero-slack algorithm was proposed by Hauge, Nair and Yoffa [4]; fictitious facilities and floating anchors methods were used by Marek-Sadowska and Lin [11], and a linear programming approach was used by Jackson, Srinivasan and Kuh [6] [7]. Several other approaches, including simulated annealing, have also been studied [1] [10] [15]. Since no global routing solution is generally available at the placement step, most of these placement algorithms use the net bounding box semiperimeter to estimate the interconnection delay of a net.

While such techniques have been developed for timing-driven placement, only limited progress has been reported for the timing-driven interconnection problem. In [2], net priorities are determined based on static timing analysis; nets with high priorities are processed earlier using fewer feedthroughs. In [8], a hierarchical approach to timing-driven routing was outlined. In [12], 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, their solutions are not flexible enough to provide a trade-off between interconnection delay and routing cost.

In this paper, we give a solution for timing-driven global routing in cell-based design regimes. The method is motivated by considering the highly similar problem of finding minimum spanning trees of bounded radius. In particular, when given a lower bound R for the spanning tree radius, we search for spanning trees with radius $(1+\epsilon)\cdot R$. Such a formulation offers a very natural, smooth trade-off between the tree radius (maximum signal delay) and the tree weight (total interconnection length). This in turn affords the circuit designer a great deal of algorithmic flexibility, as the parameter ϵ can be varied depending on performance constraints. The timing-driven global router that we propose is based on several simple yet very effective heuristic algorithms for computing bounded radius minimum spanning trees. Extensive experimental results show that our global router reduces interconnection delay by, e.g., an average of 28% for 10-pin nets, when compared with conventional minimum spanning tree based global routers. Moreover, our method indeed produces an entire class of routing solutions which embody the trade-off between minimum delay and minimum wire cost.

The remainder of this paper is organized as follows. In Section 2, we present a general formulation of the performance-driven global routing problem. In Section 3, we present a simple yet very effective heuristic algorithm for computing bounded radius minimum spanning trees, and analyze performance bounds for the algorithm. Section 4 describes several variations and improvements of this basic algorithm, and experimental results are reported in Section 5.

2 Formulation of the Problem

A *net* is a set of terminals to be connected where one of the terminals is a *source* and the rest are *sinks*. A routing solution of a net is a spanning tree T (called the *routing tree* of the net) which connects all of the terminals in the net. 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 [3] [14]. A more accurate approximation can be obtained using the upper and lower bounds on delay in an RC tree derived in [14]. However, although both the formula for Elmore's delay and those in [14] 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 (where interconnection delay between a source and a sink is proportional to the wire length between the two terminals) is often used to derive a simpler approximation for interconnection delay (e.g., [10] [13]). In this paper, we shall also use wire length 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.

The *radius* R of a *net* is the maximum Manhattan distance from the source to any sink in the net. The length or cost of an edge between points x and y is the Manhattan distance $dist(x, y)$. The *pathlength* in the routing tree T from the source s to any sink x, denoted by *pathlength(T,s,x)*, is the sum of the lengths of all edges in the unique path from s to x. We define the *radius* of a *routing tree*, $r(T)$, to be the maximum pathlength from the source to any sink. Clearly, $r(T) \geq R$ for any routing tree T. According to the linear RC delay model, in order to minimize the interconnection delay of a net, we want to minimize the radius of the routing tree since it measures the maximum interconnection delay between the source and any sink. If minimizing the radius is our only consideration, the global routing problem is trivial: simply connecting the source to every sink using the shortest path yields $r(T) = R$, which is the best possible result. However, the cost of this routing tree, i.e., the total edge length, might be very high. In fact, we can show that the cost of the tree can be $\Omega(n)$ times larger than the cost of the minimum spanning tree, where n in the number of terminals in the net, as illustrated in Figure 1.

A routing tree with high cost may increase the overall routing area. Moreover, high cost also contributes to the interconnection delay which is not captured in the linear RC model. In order to balance the radius and the cost in the routing tree construction, we formulate the timing-driven global routing problem as follows:

The Bounded Radius Minimum Spanning Tree (BRMST) Problem: Given a net, find a routing tree T with minimum total cost such that the radius of the routing tree is no more than $(1 + \epsilon) \cdot R$, where R is the radius of the net and ϵ is a prescribed constant.

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 when $\epsilon = \infty$ we minimize the total cost of the tree. In general, as ϵ grows, there is less restriction on the radius, so we can further minimize the cost of the

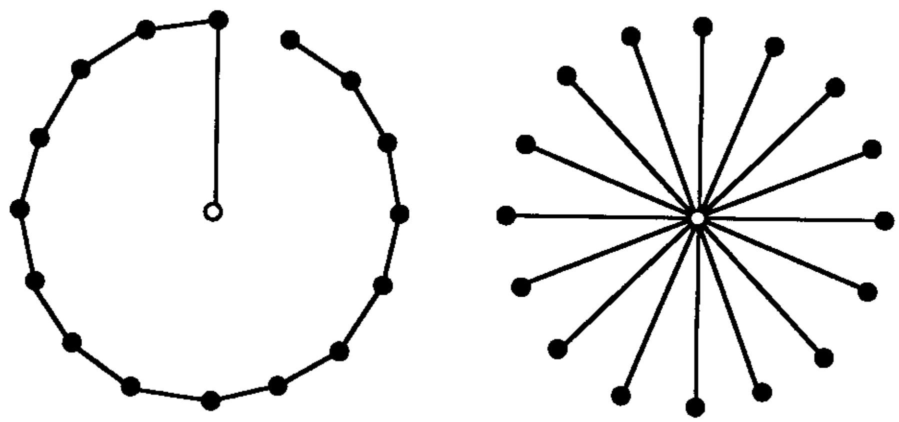

Figure 1: An example where the cost of a routing tree (right) is $\Omega(n)$ times larger than the cost of a minimum spanning tree (left).

tree. For the example of Figure 2, we show three spanning trees obtained using different values of ϵ.

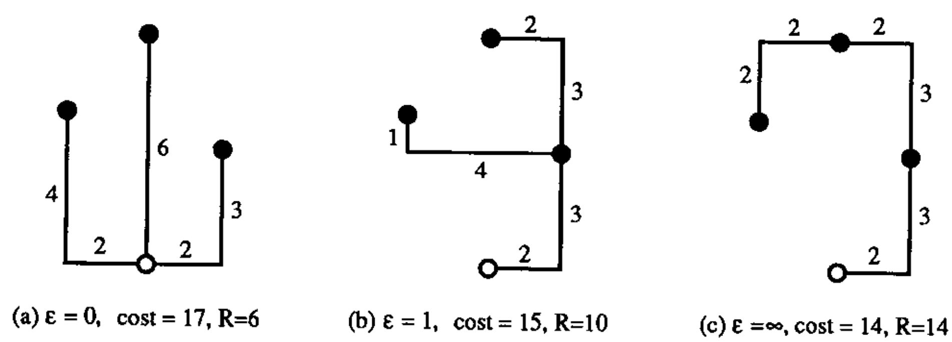

Figure 2: An example of how increasing the value of ϵ may result in a decreased tree cost, but an increased radius R.

Figure 2(a) shows the minimum radius spanning tree corresponding to the case $\epsilon = 0$, with maximum pathlength $r(T) = 6$; Figure 2(b) shows a solution with $r(T) = 10$ corresponding to the case $\epsilon = 1$; and Figure 2(c) shows the minimum cost spanning tree corresponding to the case $\epsilon = \infty$, with $r(T) = 14$. We can see that the tree cost decreases as the radius increases. The bounded-radius minimum spanning tree formulation provides a great deal of flexibility for our timing-driven global router. In practice, for nets in the timing-critical paths, the router uses small ϵ so that the interconnection delay is minimized. For nets not in any timing-critical path, the router uses large ϵ so that the total wire length is minimized.

According to the result in [5], constructing a minimum spanning tree with bounded radius in a general graph is NP-complete. However, this result does not imply that the BRMST problem is NP-complete since in our formulation, terminals are in the Manhattan plane (whose underlying graph is a grid graph) rather than a general graph. At this point, the exact complexity of the BRMST problem is still unknown.

The objective of this paper is to present a heuristic algorithm for the BRMST problem. Our goal is to construct a bounded-radius spanning tree with small cost. In the following discussion, the terminals of a net will correspond to a pointset P.

3 An Algorithm for Computing Bounded Radius Minimum Spanning Trees

Our basic algorithm for a pointset P finds a routing solution by growing a single component, following the general scheme of Prim's classical minimum spanning tree construction. We grow a tree $T = (V, E)$ which initially contains only the source s. At each step, we choose $x \in V$ and $y \in P - V$ such that $dist(x, y)$ is minimum. If adding (x, y) to T does not violate the radius constraint, i.e., $pathlength(T, s, x) + dist(x, y) \leq (1 + \epsilon) \cdot R$, we include the edge (x, y) in T. Otherwise, we "backtrace" along the path from x to s to find the first point x' such that (x', y) is *appropriate* (i.e., $pathlength(T, s, x') + dist(x', y) \leq R$), and add (x', y) to the tree. In the worst case, the backtracing will terminate with $x' = s$, since the edge(s, y) is always appropriate. Note that in backtracing we could choose x' such that $pathlength(T, s, x') + dist(x', y) \leq (1 + \epsilon) \cdot R$. However, our choice of appropriate edges leads to fewer backtracing operations, while guaranteeing that backtracing is still always possible. In other words, we intentionally introduce some "slack" at y so that points within an ϵR neighborhood of y will not cause additional backtracing. Limiting the amount of backtracing in this way will keep the cost of the resulting tree close to that of the minimum spanning tree.

We call this algorithm the **Bounded Prim** (BPRIM) construction. The high-level description is given in Figure 3. This algorithm has several advantages. First, we can show that the radius of the resulting tree is never greater than the radius of the MST whenever the MST is unique.

Property 1: If the MST is unique[1], then $r(T_{BPRIM}) \leq r(T_{MST})$.

Proof: If $r(T_{MST}) \leq (1 + \epsilon) \cdot R$, then $r(T_{BPRIM}) = r(T_{MST})$ since the two trees will be identical. Otherwise, $r(T_{BPRIM}) \leq (1 + \epsilon) \cdot R < r(T_{MST})$ by construction. □

With regard to total tree cost, we note that the difference between BPRIM and MST tree cost will

[1] When the MST is not unique, see the discussion at the end of Section 4.

```
s = source
R = maximum distance between s and any x ∈ P
T = (V, E) = ({s}, ∅)
while |V| < |P|
  Select two points x ∈ V and y ∈ P − V minimizing dist(x, y)
  if pathlength(T, s, x) + dist(x, y) ≤ (1 + ε) · R  then
      V = V ∪ {x}
      E = E ∪ {(x, y)}
  else find the first point x' along the path from x to s
      such that pathlength(T, s, x') + dist(x', y) ≤ R
      V = V ∪ {x'}
      E = E ∪ {(x', y)}
```

Figure 3: Computing a bounded-radius spanning tree T for a given pointset P and tolerance ϵ.

depend on the parameter ϵ. In practice, most nets will have between two and four pins. Furthermore, it is unlikely that a single gate will be used to drive more than six gates in CMOS design. In this case, we can show that the cost of the resulting tree is within a small constant factor of the cost of the MST for nets of practical size. Table 1 gives the worst-case ratio of BPRIM cost over MST length for small values of $|P|$, as a function of ϵ.

Property 2: Let $B(\epsilon)$ be the worst-case ratio of the cost of BPRIM output to the MST cost. Then the following bounds hold:

Net size	Bound B(ϵ)	$\epsilon = 0$	$\epsilon = \frac{1}{2}$	$\epsilon = 1$
$n = 2$	1	1	1	1
$n = 3$	$\frac{2}{1+\epsilon}$	2	$\frac{4}{3}$	1
$n = 4$	$\max(\frac{2+\epsilon}{1+\epsilon}, \frac{3}{1+2\epsilon})$	3	$\frac{5}{3}$	$\frac{3}{2}$
$n = 5$	$\max(\frac{3+\epsilon}{1+\epsilon}, \frac{4}{1+3\epsilon})$	4	$\frac{7}{3}$	2
$n = 6$	$\max(\frac{4+\epsilon}{1+\epsilon}, \frac{5}{1+4\epsilon})$	5	3	$\frac{5}{2}$

Table 1: Analysis for small nets. Results are based on Manhattan distance.

Proof: These results are obtained by studying the number of backtracings that can occur. We show the proof for $n = 5$. Other cases are similar.

Assume that the point set has unit radius. Let $c(T)$ be the cost of a minimum spanning tree. Suppose that there is only one backtracing. Clearly $c(T) \leq 1 + \epsilon$. Let $c(e)$ be the edge which cause the backtracing. Then

$$B(\epsilon) \leq \frac{c(T) - c(e) + 1}{c(T)} \leq 1 + \frac{1}{c(T)} \leq 1 + \frac{1}{1+\epsilon} = \frac{2+\epsilon}{1+\epsilon}$$

If backtracing occurs twice, let $c(x)$ and $c(y)$ be the costs of the edges which cause the backtracings. Then,

$$B(\epsilon) \leq \frac{c(T) - c(x) - c(y) + 2}{c(T)} \leq 1 + \frac{2}{c(T)} \leq 1 + \frac{2}{1+\epsilon} = \frac{3+\epsilon}{1+\epsilon}$$

If backtracing occurs three times, the tree produced by BPRIM is a star graph. Moreover, in this case, it is easy to see that $T \leq 1 + 3\epsilon$. Thus,

$$B(\epsilon) \leq \frac{4}{c(T)} \leq \frac{4}{1+3\epsilon}$$

Therefore,

$$B(\epsilon) \leq \max(\frac{2+\epsilon}{1+\epsilon}, \frac{3+\epsilon}{1+\epsilon}, \frac{4}{1+3\epsilon}) = \max(\frac{3+\epsilon}{1+\epsilon}, \frac{4}{1+3\epsilon})$$

□

In fact, the experimental results of Section 5 show that $B(\epsilon)$ is still bounded by a a small constant even for very large pointsets (i.e., see the tables of Appendix I). However, examples exist which show that the worst-case performance ratio of BPRIM is not bounded by any constant for any value of ϵ.

Theorem: For any ϵ and any constant K, there exists a pointset for which BPRIM will have a performance ratio greater than K.

Proof: On the pointset illustrated in Figure 4, BPRIM will have an unbounded performance ratio; moreover, this pointset can be modified (by adding closely-spaced points which form an arbitrarily long snaking path between the clock source s and y) so as to yield arbitrarily large performance ratio for *any* value of ϵ.

□

The time complexity of BPRIM is $O(n^2)$ and is easily shown to be $\Theta(n^2)$, since there are instances where each new point will force examination of most of the points that have already been added to the tree.

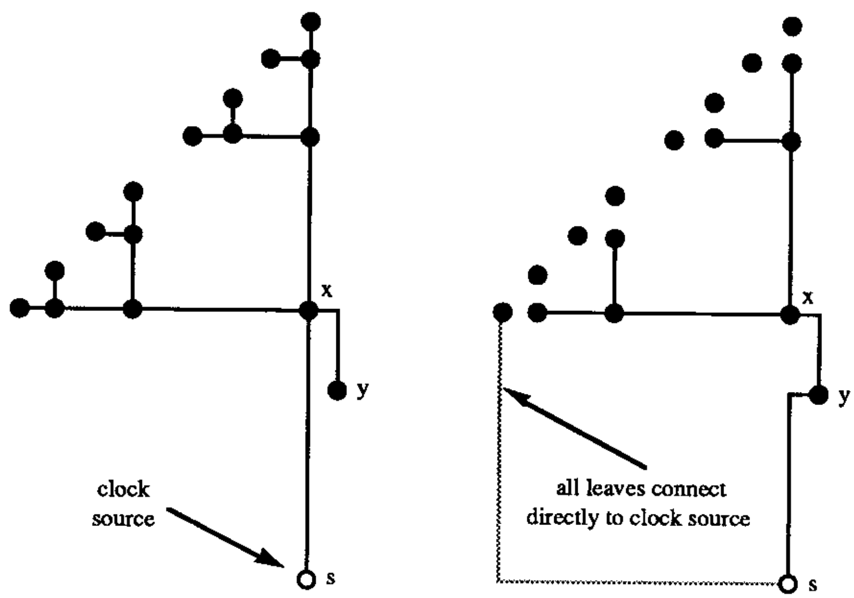

Figure 4: Example where the performance ratio of the algorithm is not bounded by any constant for any ϵ. The optimal solution is shown on the left, while the BPRIM output is shown on the right.

4 Extensions of the Basic Algorithm

As it turns out, the bounded-radius construction can also be applied to minimum spanning tree methods other than Prim's algorithm. A more general algorithm template could be as follows:

$s = source$
$R =$ maximum distance between s and any $x \in P$
$T = (V, E) = (\{s\}, \emptyset)$
while $|V| < |P|$
  Select two points $x \in V$ and $y \in P - V$,
    with $pathlength(T, s, x) + dist(x, y) \leq (1 + \epsilon) \cdot R$
  $V = V \cup \{x\}$
  $E = E \cup \{(x, y)\}$

Figure 5: Computing a bounded-radius spanning tree T for a given pointset P and tolerance ϵ.

This general template gives rise to a number of distinct variants, depending upon how the pair of points x and y are selected inside the inner loop. Several variants give significant performance improvements over the BPRIM algorithm:

- **H1** - Find x and y as in BPRIM, and among all points x' along the path in T from x to s, select the point x' which yields a *minimum-length* appropriate edge (x', y); add (x', y) to T.
- **H2** - Find a point $y \in P - V$ that minimizes $dist(x, y)$ for any $x \in V$, and among all points $x' \in V$, select the point x' which yields a minimum-length appropriate edge (x', y); add (x', y) to T.
- **H3** - Find a pair of points $x \in V$ and $y \in P - V$ that yield a minimum-length appropriate edge (x, y); add (x, y) to T.

Property 1 also holds for H1, H2, and H3. However, when the MST is *not* unique, it is possible for the radius of the BPRIM construction to be arbitrarily larger than the radius of the MST. For example, in Figure 4, if we move the point y slightly to the right so that $dist(s, x) = dist(s, y)$, BPRIM may initially connect s to either x or y, and will eventually construct either the tree shown on the left or the one shown on the right of Figure 4, respectively. In this way, an unfortunate sequence of choices by BPRIM may yield $r(T) >> r(MST)$ even though the two trees have identical costs.

The time complexity of variants H1 and H2 is $\Theta(n^2)$, while variant H3 can be easily implemented within time $O(n^3)$. Figure 4 shows that each variant will also have unbounded worst-case performance ratio.

5 Experimental Results

The BPRIM algorithm and variants H1, H2, and H3 were implemented in ANSI C for the Sun-4, Macintosh and IBM environments; code is available from the authors upon request. The algorithms were tested on a large number of random pointsets of up to 500 points, generated from a uniform distribution in the 1000 x 1000 grid. As noted in, e.g., [9], any set of approximation heuristics induces a *meta-heuristic* which returns the best solution found by any heuristic in the set and has asymptotic complexity equal to that of the slowest original heuristic; we implemented the meta-heuristic over H1, H2, and H3 and labeled this as H4, which takes $O(n^3)$ time.

Although there exist examples where the BPRIM algorithm outperforms the more complicated variants (e.g., see Figure 6), the data indicates that *on average*, variant H1 significantly dominates BPRIM, H2 significantly dominates H1, and H3 significantly dominates H2, as can be seen in the tables of Appendix II.

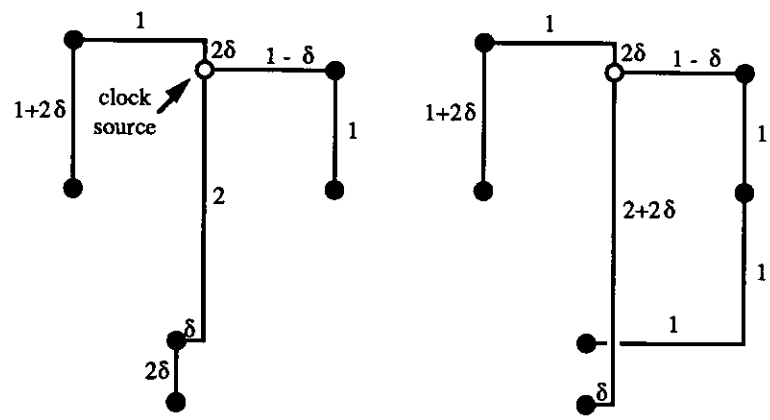

Figure 6: Example where BPRIM outperforms variants H2 and H3; here δ is a very small real number and $\epsilon = (2 - 3\delta)/(2 + 3\delta)$.

6 Extensions and Open Problems

Our basic algorithm and all of its variants readily extend to other norms, to higher dimensions, and to alternate geometries (e.g., 45- or 30-60-90-degree routing regimes). In addition, the algorithm can be applied to arbitrary weighted graphs with non-metric edge weights, as in building-block and mixed-mode design. Extensions to performance-driven Steiner routing are also straightforward.

There are several interesting open problems. First, the complexity of the BRMST problem is still unknown when points are in a plane. Also, it is open if there is a polynomial time approximation algorithm for the BRMST problem with constant performance ratio (the constant would be independent of the problem size but depend on the value of ϵ). Moreover, for a given set of points, if the minimum spanning tree is not unique, it is unknown how to choose one with the minimum radius. Further studies of these problems are in progress.

7 Conclusion

We present a family of global routing heuristics which construct a bounded-radius spanning tree for a given pointset. Our approach allows a smooth tradeoff between delay minimization and total wirelength, and has good empirical performance along with efficient time complexity.

References

[1] W. E. Donath, R. J. Norman, B. K. Agrawal, and S. E. Bello, "Timing Driven Placement Using Complete Path Delays", *Proc. IEEE Design Automation Conf.*, 1990, pp. 84-89.

[2] A. E. Dunlop, V. D. Agrawal, D.N. Deutsch, M. F. Jukl, P. Kozak, and M. Wiesel, "Chip Layout Optimization Using Critical Path Weighting", *Proc. IEEE Design Automation Conf.*, 1984, pp. 133-136.

[3] W. C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wide-Band Amplifiers", *J. Appl. Phys.* 19(1) (1948), pp. 55-63.

[4] P. S. Hauge, R. Nair, and E. J., Yoffa, "Circuit Placement for Predictable Performance", *Proc. IEEE Intl. Conf. on Computer Aided Design*, 1987, pp. 88-91.

[5] J. Ho, D. T. Lee, C. H. Chang, and C. K. Wong, "Bounded-Diameter Spanning Trees and Related Problems", *Proc. ACM Symp. on Computational Geometry*, 1989, pp. 276-282.

[6] M. A. B. Jackson, and E. S. Kuh, "Performance-Driven Placement of Cell-Based IC's", *Proc. IEEE Design Automation Conference*, 1989, pp. 370-375.

[7] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, "Clock Routing for High-Performance IC's", *Proc. IEEE Design Automation Conf.*, 1990, pp. 573-579.

[8] M. A. B. Jackson, E. S. Kuh, and M. Marek-Sadowska, "Timing-Driven Routing for Building Block Layout", *Proc. IEEE Intl. Symp. on Circuits and Systems*, 1987, pp. 518-519.

[9] A. Kahng and G. Robins, "A New Family of Steiner Tree Heuristics With Good Performance: The Iterated 1-Steiner Approach", *Proc. IEEE Intl. Conf. on Computer Aided Design*, 1990, pp. 428-431.

[10] I. Lin, and D. H. C. Du, "Performance-Driven Constructive Placement", *Proc. IEEE Design Automation Conference*, 1990, pp. 103-106.

[11] M. Marek-Sadowska, and S. P. Lin, "Timing Driven Placement", *Proc. IEEE Intl. Conf. on Computer Aided Design*, 1989, pp. 94-97.

[12] Y. Ogawa, M. Pedram, and E. S. Kuh, "Timing-Driven Placement for General Cell Layout", *Proc. International Symp. on Circuits and Systems*, 1990, pp. 872-876.

[13] P. Ramanathan and K. G. Shin, "A Clock Distribution Scheme for Non-Symmetric VLSI Circuits", *Proc. IEEE Intl. Conf. on Computer Aided Design*, 1989, pp. 398-401.

[14] J. Rubinstein, P. Penfield and M. A. Horowitz, "Signal Delay in RC Tree Networks", *IEEE Trans. on CAD* 2(3) (1983), pp. 202-211.

[15] S. Sutanthavibul, and E. Shragowitz, "An Adaptive Timing-Driven Layout for High Speed VLSI", *Proc. IEEE Design Automation Conf.*, 1990, pp. 90-95.

8 Appendix I: Ratios of Heuristic Tree Radius to MST Radius

The tables in this appendix give the minimum, maximum, and average ratios of the heuristic tree radius to the MST radius. The data shown represents averages of 500 cases generated from a uniform distribution in the unit square. The source node was selected to be one of the points at random.

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	0.43	0.81	1.00	0.43	0.81	1.00	0.43	0.81	1.00	0.43	0.81	1.00	0.43	0.81	1.00
8	0.37	0.74	1.00	0.37	0.74	1.00	0.37	0.74	1.00	0.37	0.74	1.00	0.37	0.74	1.00
10	0.38	0.72	1.00	0.38	0.72	1.00	0.38	0.72	1.00	0.38	0.72	1.00	0.38	0.72	1.00
15	0.32	0.67	1.00	0.32	0.67	1.00	0.32	0.67	1.00	0.32	0.67	1.00	0.32	0.67	1.00
25	0.27	0.63	0.97	0.27	0.63	0.97	0.27	0.63	0.97	0.27	0.63	0.97	0.27	0.63	0.97
50	0.27	0.57	0.89	0.27	0.57	0.89	0.27	0.57	0.89	0.27	0.57	0.89	0.27	0.57	0.89
100	0.24	0.52	0.92	0.24	0.52	0.92	0.24	0.52	0.92	0.24	0.52	0.92	0.24	0.52	0.92
200	0.25	0.48	0.77	0.25	0.48	0.77	0.25	0.48	0.77	0.25	0.48	0.77	0.25	0.48	0.77

Table 2: Minimum, average, and maximum radius ratios for $\epsilon = 0.01$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	0.42	0.82	1.00	0.42	0.82	1.00	0.42	0.82	1.00	0.42	0.82	1.00	0.42	0.82	1.00
8	0.26	0.77	1.00	0.26	0.77	1.00	0.28	0.77	1.00	0.28	0.77	1.00	0.28	0.77	1.00
10	0.36	0.75	1.00	0.36	0.74	1.00	0.36	0.75	1.00	0.36	0.75	1.00	0.36	0.75	1.00
15	0.34	0.71	1.00	0.34	0.71	1.00	0.34	0.71	1.00	0.34	0.71	1.00	0.34	0.71	1.00
25	0.33	0.69	1.00	0.33	0.68	1.00	0.33	0.69	1.00	0.32	0.69	1.00	0.32	0.69	1.00
50	0.30	0.61	0.99	0.30	0.61	0.99	0.31	0.61	0.99	0.30	0.61	0.99	0.30	0.61	0.99
100	0.28	0.59	0.93	0.28	0.59	0.93	0.28	0.59	0.93	0.28	0.59	0.93	0.28	0.59	0.93
200	0.25	0.53	0.91	0.25	0.53	0.91	0.25	0.53	0.91	0.24	0.53	0.91	0.24	0.53	0.91

Table 3: Minimum, average, and maximum radius ratios for $\epsilon = 0.10$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	0.40	0.87	1.00	0.40	0.87	1.00	0.40	0.87	1.00	0.40	0.87	1.00	0.40	0.87	1.00
8	0.40	0.83	1.00	0.40	0.82	1.00	0.40	0.82	1.00	0.39	0.82	1.00	0.39	0.82	1.00
10	0.38	0.81	1.00	0.38	0.81	1.00	0.38	0.81	1.00	0.38	0.81	1.00	0.38	0.81	1.00
15	0.32	0.79	1.00	0.32	0.78	1.00	0.32	0.78	1.00	0.32	0.78	1.00	0.32	0.78	1.00
25	0.37	0.76	1.00	0.37	0.76	1.00	0.37	0.76	1.00	0.37	0.76	1.00	0.37	0.76	1.00
50	0.33	0.70	1.00	0.32	0.70	1.00	0.33	0.70	1.00	0.33	0.70	1.00	0.33	0.70	1.00
100	0.33	0.65	1.00	0.33	0.65	1.00	0.33	0.65	1.00	0.33	0.65	1.04	0.33	0.65	1.04
200	0.29	0.61	0.96	0.29	0.61	0.96	0.29	0.61	0.96	0.29	0.61	0.96	0.29	0.61	0.96

Table 4: Minimum, average, and maximum radius ratios for $\epsilon = 0.25$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	0.41	0.93	1.00	0.41	0.92	1.00	0.41	0.92	1.00	0.41	0.92	1.00	0.41	0.92	1.00
8	0.48	0.92	1.00	0.33	0.92	1.00	0.33	0.92	1.00	0.44	0.92	1.00	0.44	0.92	1.00
10	0.46	0.91	1.00	0.45	0.90	1.00	0.45	0.90	1.00	0.48	0.90	1.00	0.48	0.90	1.00
15	0.44	0.90	1.00	0.44	0.89	1.00	0.44	0.89	1.00	0.41	0.89	1.31	0.41	0.89	1.31
25	0.38	0.86	1.00	0.37	0.86	1.00	0.37	0.86	1.00	0.37	0.86	1.07	0.37	0.86	1.07
50	0.39	0.83	1.00	0.39	0.83	1.00	0.38	0.82	1.00	0.39	0.82	1.04	0.39	0.82	1.04
100	0.38	0.77	1.00	0.38	0.77	1.00	0.38	0.77	1.00	0.38	0.77	1.04	0.38	0.77	1.04
200	0.29	0.71	1.00	0.29	0.71	1.00	0.29	0.71	1.00	0.29	0.71	1.01	0.29	0.71	1.01

Table 5: Minimum, average, and maximum radius ratios for $\epsilon = 0.50$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	0.56	0.98	1.00	0.52	0.98	1.00	0.52	0.98	1.00	0.52	0.98	1.00	0.52	0.98	1.00
8	0.55	0.96	1.00	0.55	0.96	1.00	0.57	0.96	1.00	0.57	0.96	1.00	0.57	0.96	1.00
10	0.52	0.96	1.00	0.46	0.96	1.00	0.46	0.96	1.00	0.48	0.96	1.07	0.48	0.96	1.07
15	0.50	0.95	1.00	0.48	0.94	1.00	0.48	0.94	1.00	0.48	0.94	1.02	0.48	0.94	1.02
25	0.44	0.94	1.00	0.44	0.93	1.00	0.44	0.93	1.00	0.43	0.93	1.10	0.43	0.93	1.10
50	0.50	0.90	1.00	0.49	0.89	1.00	0.49	0.89	1.00	0.48	0.89	1.06	0.48	0.89	1.06
100	0.43	0.87	1.00	0.43	0.86	1.00	0.43	0.86	1.00	0.43	0.86	1.10	0.43	0.86	1.10
200	0.37	0.82	1.00	0.37	0.82	1.00	0.37	0.81	1.00	0.37	0.81	1.12	0.37	0.81	1.12

Table 6: Minimum, average, and maximum radius ratios for $\epsilon = 0.75$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	0.58	1.00	1.00	0.58	0.99	1.00	0.58	0.99	1.00	0.58	0.99	1.00	0.58	0.99	1.00
8	0.67	0.99	1.00	0.56	0.99	1.00	0.56	0.99	1.00	0.53	0.99	1.00	0.53	0.99	1.00
10	0.65	0.99	1.00	0.57	0.98	1.00	0.57	0.98	1.00	0.57	0.98	1.00	0.57	0.98	1.00
15	0.65	0.98	1.00	0.54	0.98	1.00	0.54	0.97	1.00	0.54	0.97	1.06	0.54	0.97	1.06
25	0.48	0.98	1.00	0.48	0.97	1.00	0.48	0.97	1.00	0.46	0.97	1.10	0.46	0.97	1.10
50	0.53	0.95	1.00	0.53	0.94	1.00	0.53	0.94	1.00	0.53	0.94	1.06	0.53	0.94	1.06
100	0.52	0.93	1.00	0.52	0.93	1.00	0.52	0.92	1.00	0.52	0.92	1.09	0.52	0.92	1.09
200	0.45	0.89	1.00	0.45	0.89	1.00	0.45	0.88	1.00	0.45	0.88	1.12	0.45	0.88	1.12

Table 7: Minimum, average, and maximum radius ratios for $\epsilon = 1.00$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	1.00	1.00	1.00	1.00	1.00	1.00	0.69	1.00	1.00	0.69	1.00	1.00	0.69	1.00	1.00
8	0.86	1.00	1.00	0.80	1.00	1.00	0.67	1.00	1.00	0.67	1.00	1.00	0.67	1.00	1.00
10	0.96	1.00	1.00	0.96	1.00	1.00	0.78	1.00	1.00	0.78	1.00	1.18	0.78	1.00	1.18
15	1.00	1.00	1.00	1.00	1.00	1.00	0.85	1.00	1.00	0.85	1.00	1.10	0.85	1.00	1.10
25	0.84	1.00	1.00	0.84	1.00	1.00	0.81	1.00	1.00	0.69	1.00	1.26	0.69	1.00	1.26
50	0.82	1.00	1.00	0.82	1.00	1.00	0.78	1.00	1.00	0.58	0.99	1.16	0.58	0.99	1.16
100	0.73	1.00	1.00	0.72	1.00	1.00	0.73	0.99	1.00	0.66	0.99	1.09	0.66	0.99	1.09
200	0.77	0.99	1.00	0.77	0.99	1.00	0.77	0.99	1.00	0.59	0.98	1.24	0.59	0.98	1.24

Table 8: Minimum, average, and maximum radius ratios for $\epsilon = 2.00$

9 Appendix II: Ratios of Heuristic Tree Cost to MST Cost

The tables in this appendix give the minimum, maximum, and average ratios of the heuristic tree cost to the MST cost. The data shown represents averages of 500 cases generated from a uniform distribution in the unit square. The source node was selected to be one of the points at random.

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	1.00	1.25	2.47	1.00	1.25	2.47	1.00	1.24	2.47	1.00	1.23	2.47	1.00	1.23	2.47
8	1.00	1.36	3.09	1.00	1.35	3.09	1.00	1.34	2.61	1.00	1.31	2.61	1.00	1.30	2.61
10	1.00	1.44	2.72	1.00	1.42	2.72	1.00	1.39	2.76	1.00	1.36	2.76	1.00	1.35	2.72
15	1.00	1.51	2.75	1.00	1.46	2.44	1.00	1.43	2.44	1.00	1.39	2.36	1.00	1.37	2.36
25	1.01	1.74	3.15	1.01	1.61	2.89	1.01	1.54	2.84	1.01	1.48	2.27	1.01	1.46	2.27
50	1.14	2.13	3.96	1.14	1.78	3.58	1.12	1.65	2.76	1.09	1.57	2.63	1.09	1.54	2.54
100	1.23	2.71	4.55	1.22	1.98	3.54	1.14	1.80	3.47	1.14	1.73	3.29	1.14	1.67	3.26
200	1.75	3.49	6.02	1.36	2.22	3.88	1.19	1.97	3.49	1.18	1.94	3.41	1.18	1.84	3.41

Table 9: Minimum, average, and maximum performance ratios for $\epsilon = 0.01$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	1.00	1.17	2.22	1.00	1.17	2.22	1.00	1.17	2.22	1.00	1.16	2.22	1.00	1.16	2.22
8	1.00	1.25	2.20	1.00	1.23	1.94	1.00	1.22	2.26	1.00	1.20	2.26	1.00	1.20	1.94
10	1.00	1.28	2.33	1.00	1.26	2.33	1.00	1.25	2.18	1.00	1.23	2.18	1.00	1.22	2.18
15	1.00	1.39	2.79	1.00	1.32	2.77	1.00	1.28	2.53	1.00	1.25	2.28	1.00	1.23	2.28
25	1.00	1.53	2.71	1.00	1.39	2.45	1.00	1.33	2.30	1.00	1.28	2.16	1.00	1.25	2.00
50	1.00	1.92	3.49	1.00	1.52	2.91	1.00	1.41	2.92	1.00	1.33	2.22	1.00	1.30	2.22
100	1.07	2.37	4.96	1.05	1.67	3.21	1.04	1.49	2.92	1.03	1.40	2.51	1.03	1.37	2.11
200	1.10	3.17	5.60	1.08	1.91	3.63	1.06	1.62	3.29	1.03	1.58	3.15	1.03	1.51	3.05

Table 10: Minimum, average, and maximum performance ratios for $\epsilon = 0.10$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	1.00	1.09	1.96	1.00	1.09	1.96	1.00	1.09	1.92	1.00	1.08	1.67	1.00	1.08	1.67
8	1.00	1.16	1.86	1.00	1.14	1.86	1.00	1.13	1.86	1.00	1.12	1.86	1.00	1.11	1.86
10	1.00	1.19	2.34	1.00	1.15	2.34	1.00	1.14	2.32	1.00	1.12	2.32	1.00	1.11	2.32
15	1.00	1.26	2.37	1.00	1.19	2.25	1.00	1.16	2.01	1.00	1.13	1.74	1.00	1.12	1.62
25	1.00	1.37	2.58	1.00	1.21	2.16	1.00	1.17	2.16	1.00	1.13	1.94	1.00	1.12	1.83
50	1.00	1.67	3.14	1.00	1.33	2.08	1.00	1.23	2.36	1.00	1.16	1.66	1.00	1.15	1.66
100	1.00	2.07	4.52	1.00	1.42	2.61	1.00	1.27	2.17	1.00	1.20	2.20	1.00	1.18	1.99
200	1.00	2.62	5.50	1.00	1.61	2.84	1.00	1.35	2.50	1.00	1.28	2.30	1.00	1.25	2.27

Table 11: Minimum, average, and maximum performance ratios for $\epsilon = 0.25$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	1.00	1.05	1.60	1.00	1.04	1.56	1.00	1.04	1.56	1.00	1.04	1.56	1.00	1.04	1.56
8	1.00	1.07	1.97	1.00	1.05	1.59	1.00	1.05	1.59	1.00	1.04	1.84	1.00	1.04	1.59
10	1.00	1.09	1.73	1.00	1.06	1.59	1.00	1.06	1.59	1.00	1.05	1.59	1.00	1.05	1.59
15	1.00	1.13	2.08	1.00	1.08	1.60	1.00	1.06	1.53	1.00	1.05	1.53	1.00	1.05	1.53
25	1.00	1.21	2.91	1.00	1.10	1.97	1.00	1.08	1.88	1.00	1.05	1.72	1.00	1.05	1.72
50	1.00	1.40	3.67	1.00	1.15	1.93	1.00	1.10	1.75	1.00	1.06	1.77	1.00	1.06	1.74
100	1.00	1.68	4.39	1.00	1.23	2.61	1.00	1.13	1.92	1.00	1.07	1.70	1.00	1.06	1.70
200	1.00	2.16	5.50	1.00	1.33	2.78	1.00	1.16	2.21	1.00	1.10	1.79	1.00	1.08	1.61

Table 12: Minimum, average, and maximum performance ratios for $\epsilon = 0.50$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	1.00	1.01	1.41	1.00	1.01	1.41	1.00	1.01	1.41	1.00	1.01	1.41	1.00	1.01	1.41
8	1.00	1.03	1.63	1.00	1.02	1.41	1.00	1.02	1.40	1.00	1.02	1.40	1.00	1.02	1.40
10	1.00	1.05	1.83	1.00	1.03	1.43	1.00	1.02	1.41	1.00	1.02	1.41	1.00	1.02	1.41
15	1.00	1.07	2.28	1.00	1.04	1.60	1.00	1.03	1.48	1.00	1.03	1.38	1.00	1.02	1.38
25	1.00	1.12	2.48	1.00	1.05	1.58	1.00	1.04	1.64	1.00	1.03	1.49	1.00	1.03	1.41
50	1.00	1.24	3.06	1.00	1.09	1.70	1.00	1.06	1.65	1.00	1.03	1.49	1.00	1.03	1.43
100	1.00	1.41	4.20	1.00	1.12	1.87	1.00	1.06	1.69	1.00	1.04	1.71	1.00	1.03	1.57
200	1.00	1.70	4.79	1.00	1.19	2.24	1.00	1.08	1.87	1.00	1.04	1.53	1.00	1.03	1.37

Table 13: Minimum, average, and maximum performance ratios for $\epsilon = 0.75$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	1.00	1.00	1.27	1.00	1.00	1.27	1.00	1.00	1.27	1.00	1.00	1.27	1.00	1.00	1.27
8	1.00	1.01	1.73	1.00	1.01	1.54	1.00	1.01	1.54	1.00	1.01	1.54	1.00	1.01	1.54
10	1.00	1.02	1.47	1.00	1.01	1.32	1.00	1.01	1.31	1.00	1.01	1.31	1.00	1.01	1.31
15	1.00	1.03	1.79	1.00	1.02	1.30	1.00	1.01	1.30	1.00	1.01	1.30	1.00	1.01	1.30
25	1.00	1.04	2.38	1.00	1.02	1.39	1.00	1.01	1.37	1.00	1.01	1.33	1.00	1.01	1.33
50	1.00	1.13	2.66	1.00	1.04	1.71	1.00	1.03	1.47	1.00	1.02	1.31	1.00	1.02	1.31
100	1.00	1.22	3.10	1.00	1.06	1.67	1.00	1.03	1.58	1.00	1.02	1.38	1.00	1.01	1.31
200	1.00	1.45	5.27	1.00	1.11	2.09	1.00	1.05	1.57	1.00	1.02	1.38	1.00	1.02	1.33

Table 14: Minimum, average, and maximum performance ratios for $\epsilon = 1.00$

# pts	BPRIM min	BPRIM ave	BPRIM max	H1 min	H1 ave	H1 max	H2 min	H2 ave	H2 max	H3 min	H3 ave	H3 max	H4 min	H4 ave	H4 max
5	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
8	1.00	1.00	1.34	1.00	1.00	1.07	1.00	1.00	1.07	1.00	1.00	1.07	1.00	1.00	1.07
10	1.00	1.00	1.08	1.00	1.00	1.08	1.00	1.00	1.08	1.00	1.00	1.08	1.00	1.00	1.08
15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
25	1.00	1.00	1.39	1.00	1.00	1.14	1.00	1.00	1.14	1.00	1.00	1.09	1.00	1.00	1.09
50	1.00	1.00	1.71	1.00	1.00	1.13	1.00	1.00	1.11	1.00	1.00	1.11	1.00	1.00	1.09
100	1.00	1.01	2.74	1.00	1.00	1.39	1.00	1.00	1.15	1.00	1.00	1.08	1.00	1.00	1.08
200	1.00	1.02	2.06	1.00	1.01	1.85	1.00	1.00	1.35	1.00	1.00	1.04	1.00	1.00	1.04

Table 15: Minimum, average, and maximum performance ratios for $\epsilon = 2.00$