Rectilinear Steiner Trees with Minimum Elmore Delay*

Kenneth D. Boese, Andrew B. Kahng, Bernard A. McCoy†, and Gabriel Robins†

CS Dept., University of California at Los Angeles, Los Angeles, CA 90024-1596
† CS Dept., University of Virginia, Charlottesville, VA 22903-2442

Abstract

We provide a new theoretical framework for constructing Steiner routing trees with minimum Elmore delay. Earlier work [3, 13] has established Elmore delay as a **high fidelity** *estimate of "physical", i.e., SPICE-computed, signal delay. Previously, however, it was not known how to construct an Elmore delay-optimal Steiner tree. Our main theoretical result is a generalization of Hanan's theorem [11] which limited the number of possible locations of Steiner nodes in an optimal delay rectilinear Steiner tree. Another theoretical result establishes a new decomposition theorem for constructing optimal-delay Steiner trees. We develop a branch-and-bound method, called BB-SORT-C, which exactly minimizes any linear combination of Elmore sink delays; BB-SORT-C is practical for routing small nets and for delimiting the space of achievable routing solutions with respect to Elmore delay.*

1 Introduction

Due to the scaling of VLSI technology, interconnection delay dominates the design of high-performance systems [8, 17]. Performance-driven routing has thus received considerable attention; a typical goal is to minimize average or maximum source-sink delay in a given signal net. Early work, e.g. [9], implicitly equated optimal routing with minimum-cost Steiner routing. More recent works recognize that delay minimization and wire length minimization can be far from synonymous. Cohoon and Randall [5] consider both the *cost* (total edge length) and the *radius* (longest source-sink path length) of the heuristic routing tree. Cong et al. [6] use a parameter ϵ to guide the tradeoff between cost and radius minimization; Alpert et al. [1] achieve a more direct cost-radius tradeoff between minimum spanning tree and shortest path tree constructions; and Cong et al. [7] propose the use of rectilinear Steiner arborescences [15].

*Partial support for this work was provided by NSF MIP-9110696, NSF Young Investigator Award MIP-9257982, ARO DAAK-70-92-K-0001, and ARO DAAL-03-92-G-0050.

Such previous routing methods have essentially "geometric" objectives which are difficult to tune to specific technology parameters. Boese et al. [2] have addressed this flaw with a construction that greedily optimizes Elmore delay directly. Supporting investigations in [3] demonstrate that Elmore delay has high fidelity to physical (SPICE-computed) delay (i.e., near-optimal Elmore delay implies near-optimal SPICE delay). This confirms earlier studies by Kim et al. [13] and Vlach et al. [19].

A natural question at this point is: How much better is possible? What is the *performance envelope* for routing tree constructions? Boese et al. [3] used branch-and-bound to construct optimal Elmore delay *spanning* trees and found that the *Elmore Routing Tree* (ERT) construction of [2] is on average only 2.3% above optimal for 7-pin nets. The more significant open question concerns the near-optimality of Steiner tree heuristics: the essential difficulty has been a potentially unbounded number of candidate Steiner node locations, which makes even branch-and-bound impossible.

In this paper, we present new theoretical results that allow construction of Elmore delay-optimal Steiner trees. Our key result restricts the Steiner nodes in an optimal Elmore delay rectilinear Steiner tree to the "Hanan grid," generalizing a theorem of Hanan for minimum cost Steiner trees [11]. Using this restriction and a new decomposition theorem (which also applies to minimum-cost Steiner trees) we show how branch-and-bound can construct a *Steiner Optimal Routing Tree* (SORT). Our results also give new restrictions on the structure of a SORT. Our experimental results establish that the SERT-C and SERT constructions of [2] are on average within only 5% of optimal for 5-pin nets and within 16% of optimal for 9-pin nets, depending on the technology parameters.

2 Preliminaries

Previous performance-driven routing constructions generally address *net-specific* objectives (cost, radius, cost-radius tradeoffs, etc.) rather than *sink-specific* objectives which exploit the *critical-path* information typically available from iterated placement and routing phases of performance-driven layout. [2] showed that a significant timing improvement is achieved by minimizing delay to a single critical sink, with only a small tree cost penalty as compared to the 1-Steiner algorithm of [12]. Thus, we use the critical-sink problem formulation of [2].

A *signal net* N consists of a set of pin locations $\{n_0, n_1, ..., n_k\}$ in the Manhattan plane, which are to be connected by a *routing tree* $T(N)$. Location n_0 is designated to be the *source*, with the n_i locations $(1 \leq i \leq k)$ denoting *sinks*. The *cost* of an edge in $T(N)$ is the Manhattan distance between its endpoints. The *cost* of a routing tree $T(N)$ is the sum of its edge costs. Elmore delay in $T(N)$ between n_0 and sink n_j is denoted by $t(n_j)$. Finally, each sink n_i is given an associated level of *criticality*, $\alpha_i \geq 0$. Our goal is to solve the

Critical-Sink Routing Tree (CSRT) Problem: *Given signal net N, construct $T(N)$ which minimizes*

$$\sum_{i=1}^{k} \alpha_i \cdot t(n_i).$$

2.1 Elmore Delay

Elmore delay [10, 16, 18] is a distributed RC delay approximation defined as follows. Given routing tree $T(N)$ rooted at the source n_0, let e_v denote the edge from node v to its parent in $T(N)$. The resistance and capacitance of edge e_v are denoted by r_{e_v} and c_{e_v}, respectively. Let T_v denote the subtree of T rooted at v, and let c_v denote the sink capacitance of v ($c_v = 0$ if v is a Steiner node). We use C_v to denote the *tree capacitance* of T_v, namely the sum of sink and edge capacitances in T_v. Using this notation, the Elmore delay along edge e_v is equal to $r_{e_v}(\frac{c_{e_v}}{2} + C_v)$. Let r_d denote the output driver resistance at the net's source. Then Elmore delay $t(n_i)$ at sink n_i is:

$$t(n_i) = r_d C_{n_0} \quad + \sum_{e_v \in path(n_0, n_i)} r_{e_v}(\frac{c_{e_v}}{2} + C_v)$$

2.2 The "Elmore Routing Tree" Approach

The greedy *Elmore routing tree* (ERT) approach of [2] minimizes Elmore delay *directly* during the construction of a routing tree. The ERT algorithm for spanning trees is analogous to Prim's minimum spanning tree construction [14]: starting with a trivial tree containing only the source, ERT iteratively finds a pin n_i in the tree and a sink n_j outside the tree so that adding edge (n_i, n_j) yields a tree with smallest delay in the growing tree[1].

ERT extends to Steiner routing by allowing each new pin to connect to an *edge* rather than to a pin in the existing tree. Connections to an edge are always made so that the induced Steiner node is located at the point on the edge closest to the new pin. (The exact placement/embedding of an edge is allowed to vary within its bounding box.) Very substantial delay savings for all of the ERT variants are reported in [2]; moreover, the ERT approach is efficient because Elmore delay to all sinks can be evaluated in linear time.

2.3 Summary of Algorithms

The rest of this paper will concentrate on the following high-performance Steiner routing methods.[2]

1-Steiner	best performing efficient heuristic for minimum-cost Steiner trees [12].
SERT	greedy heuristic for minimizing *maximum* sink Elmore delay [2].
SERT-C	modification of SERT to minimizing delay at a single critical sink [2].
BB-SORT	Branch-and-Bound for Steiner Optimal Routing Trees minimizing maximum sink delay.
BB-SORT-C	Branch-and-Bound for Steiner Optimal Routing Trees with Critical sinks for minimizing a linear combination of delays.

3 Theoretical Results

We use the following short-hand conventions. "Delay" will always be Elmore delay; "max delay" is the maximum source-sink delay in the net. Finally, a Steiner node is "on the Hanan grid" if it is located at the intersection of horizontal and vertical lines through pins in the net.

3.1 CSRT is NP-Hard

For any given set of circuit parameters[3], the minimum cost Steiner tree problem can be reduced to the CSRT problem for a single critical sink, as shown in Figure 1. The "generic" variant of CSRT, which seeks to minimize maximum sink delay, can use the the same reduction by setting n_c far away from n_0 so that the maximum delay occurs at n_c.

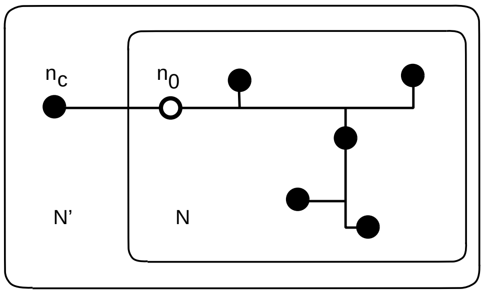

Figure 1: Proof that the CSRT problem is NP-Hard: minimum-cost Steiner tree instance (N) reduces to a CSRT instance (N') with critical sink n_c directly left of the pin n_0 in N with smallest x-coordinate. $t(n_c)$ is minimized by a tree with edge (n_0, n_c) plus the min-cost Steiner tree over N.

3.2 Branch-and-Bound for Optimal Delay Steiner Trees

The branch-and-bound method of [3] for optimal spanning trees starts with a tree containing the source and incrementally adds sinks to a growing tree while evaluating delay at each step. When the delay exceeds that of any *complete* tree seen so far, the search is

[1] For routing with a single critical sink, ERT starts with one edge between n_0 and the critical sink.

[2] Note that SERT and SERT-C can also be easily extended to handle multiple critical sinks by routing critical sinks by SERT and then non-critical sinks by SERT-C.

[3] Unit resistance, unit capacitance, loading capacitances, and driver resistance; wire sizing is not considered in our formulation.

pruned and the algorithm backtracks. The algorithm avoids redundant testing of topologies by adding sinks in breadth-first order, with sinks with the same parent connected in increasing index order. In this way, any tree topology will correspond to a unique ordering of the sinks and can be tested by the algorithm at most once.

We modify the method of [3] to find the optimal delay *Steiner* tree by assuming that an optimal tree can always be constructed iteratively by connecting a sink by a new edge directly to the source or by a *closest connection* to some edge in the current tree[4]. The modification is simply that connections are considered to each *edge* in the current tree (plus the source), rather than to each pin. Branch-and-bound pruning is used again to reduce the complexity of the search. Redundant testing of topologies is greatly (although not completely) avoided by restricting the order in which sinks can be added to construct any particular topology. Figure 2 gives details of our Branch-and-Bound method for Steiner Optimal Routing Trees with a single Critical sink (BB-SORT-C). A simple modification to Step 11 can minimize a linear combination of delays or, for BB-SORT, minimize the maximum sink delay.

```
BB-SORT-C Algorithm
Input: signal net N with critical sink n_1
Output: Steiner tree T* over N having optimal t(n_1)
1.  best = +∞
2.  for i = 1 to |N| − 1
3.      call Add_Sink(i,n_0)
4.  return T*
Procedure Add_Sink(Integer: i; Edge: e)
5.  while e ≠ NIL
6.      call Try_Connection(i,e)
7.      e = Next(e)
Procedure Try_Connection(Integer: i; Edge: e)
8.  T = Make_Connection(i,e,T)
9.  if (t(n_1) ≤ best)
10.     if (num_pins(T) == |N|)
11.         best = t(n_1); T* = T
12.     else
13.         for j = 1 to i − 1
14.             if (n_j ∉ T) call Add_Sink(j,Next(e))
15.         for j = i + 1 to |N| − 1
16.             if (n_j ∉ T) call Add_Sink(j,e)
17. T = Delete_Connection(i,e,T)
```

Figure 2: Pseudo-code for BB-SORT-C. Note that n_0 is treated like an edge in Step 3 because connections are considered to the source and to all edges in the current tree. Procedures not defined in the template: Next(e) returns the edge after e in a list of edges in T ordered by when they were added to T; Make_Connection(i,e,T) connects n_i to T by a closest connection to e; Delete_Connection(i,e,T) reverses the call to Make_Connection in Step 8.

3.3 Sub-Optimality of BB-SORT

Figure 3 gives an example in which BB-SORT constructs a sub-optimal tree in terms of maximum sink delay. Figure 3(a) shows what appears to be the optimal tree, with maximum delay $t(n_1) = t(n_2) = 28.625$. All Steiner nodes in this tree are on the vertical line $x = 1.5$, which is outside the Hanan grid. Part (b) shows the tree returned by BB-SORT, with maximum delay $t(n_1) = 28.641$.

Given that the example in Figure 3 was constructed carefully by hand, we believe that other counter-examples are rare and that BB-SORT almost always gives the optimal "generic" Steiner tree.

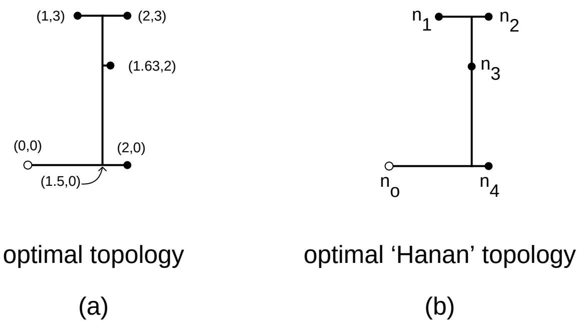

Figure 3: Counter-example for which BB-SORT returns a sub-optimal "generic" routing tree. Pin positions are shown in (a); driver resistance = 1.75; unit resistance and capacitance equal 1.0; loading capacitance = 0.37 for n_3 and 0.0 for other sinks; Part (a) appears to be optimal (max delay = $t(n_1) = t(n_2) =$ 28.625). Part (b) is returned by BB-SORT (max delay = $t(n_1) =$ 28.641).

3.4 Optimality of BB-SORT-C

For any *linear combination* of sink delays, our branch-and-bound method constructs the optimal tree. In this section we state the lemmas and theorem used to obtain this result, along with sketches of the proofs themselves. Complete proofs are contained in [4]. [5]

3.4.1 Definitions

Let T^* be a Steiner tree over net N minimizing $f = \sum_{i=1}^{k} \alpha_i \cdot t(n_i)$, with each $\alpha_i > 0$.[6] For convenience, we normalize time and distance so that unit wire resistance and unit wire capacitance both equal one. We consider any tree T as a set of nodes and edges, and so $v \in T$ for node v and $e \in T$ for edge e are well defined. A completely vertical or horizontal edge is called a *straight edge*; other edges are *L-shaped.*

The *closest connection* between three nodes is the location of the Steiner node in a minimum-cost Steiner tree over the nodes.[7] The *closest connection* between node v and edge e is the closest connection between v and the endpoints of e. Assume that a tree T is rooted at n_0. We define $T \backslash v$ to be the tree induced by removing node v and its descendants from T, then removing all degree-2 Steiner nodes. We say that node $v \in T$ is *connected to* edge $e \in T \backslash v$ if its parent node in T is

[4] A *closest connection* to a given edge is made by creating a Steiner node at the point on the edge closest to the new sink.

[5] Note that we allow n_0 to have degree > 4, which is physically impossible, but can be approximated by merging wires close to n_0. Also, the optimal tree is not always planar, as this is not required in the CSRT problem formulation, nor is it required in multi-layer routing.

[6] The case of $\alpha_i = 0$ for one or more i is effectively handled by setting α_i to a small positive value.

[7] This location is unique and has coordinates given by the medians of the x- and the y- coordinates of the three nodes.

located on edge e (including perhaps an endpoint of e). If $parent(v)$ is located at the closest connection between v and edge $e \in T \backslash v$, then v *makes a closest connection* to edge e.

3.4.2 Proof of Closest Connections in T^*

Lemma 1: Suppose node $a \in T^*$, $a \neq n_0$, is connected to edge $e \in T^* \backslash a$. Then either $parent(a) = n_0$ or a makes a closest connection to e in T^*.

Proof Sketch: Let $x = parent(a)$ and let c be the closest connection between node a and edge $e = (p, b)$, as in Figure 4. For convenience we overload x, a, b, c, and p to also represent the edge lengths from p to these respective nodes or locations. It is easy to see that $x \leq c$, since otherwise moving x to c will reduce tree cost *and* reduce or leave unchanged all path lengths. For $p \leq x \leq c$, application of the Elmore formula shows that delay f is a *concave* function of x.[8] Consequently, f can only be minimized at the boundaries $x = p$ or $x = c$. Further application of the Elmore formula shows that the capacitances of edge (p, d) and subtree T^*_d do not affect the concavity of f for x between q and c, and so $x \neq p$ (unless $p = n_0$). Thus, either $x = c$ or $x = p = n_0$. □

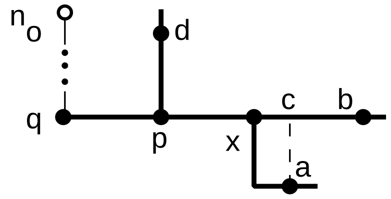

Figure 4: Lemma 1: Node $a \in T^*$ is connected to edge $(p, b) \in T^* \backslash a$ at node x; either $x = p = n_0$ or $x = c$, where c is the closest connection between a and (p, b).

By itself, Lemma 1 is not sufficient to prove optimality of BB-SORT-C. The tree in Figure 5 has all nodes $v \neq n_0$ either connected to n_0 or making a closest connection to an edge in $T \backslash v$; however, this tree cannot be constructed by BB-SORT-C.

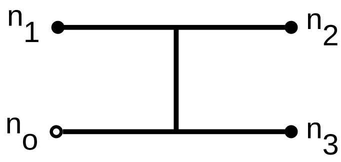

Figure 5: Lemma 1 is not sufficient to prove optimality of BB-SORT-C. This tree satisfies the conclusions of Lemma 1 but cannot be constructed by BB-SORT-C.

3.4.3 Hanan Grid Proof for T^*

Define a *segment* to be a contiguous set of straight edges in tree T which are either all horizontal or all vertical; a *maximal segment* (MS) is a segment not properly contained in any other segment. The node in an MS M closest to n_0 topologically is the *entry point* to M. A segment containing all edges in an MS M to one side of M's entry point is called a *branch.* In addition, all L-shaped edges in T are also *branches.* A branch b is a *branch off of* MS M' if M' and b are incident at a single node which is not the entry point to M'. An MS M divides the plane into two half-planes; the half-plane containing the edge between M's entry point and its parent is called the *near side* of M, while the other half-plane is called the *far side* of M. Branches off of M that are located on its near (resp. far) side are called *near* (resp. *far*) *branches.* In addition, a *sink* located on M is defined to be a *far branch* off of M if it is not the entry point to a larger far branch. We use $Near(M)$ (resp. $Far(M)$) to denote the set of near (resp. far) branches off of MS M. Figure 6 shows an example of an MS M with endpoints p_1 and n_3, entry point p_0, and four branches, including near branch b_1 and far branches b_2, b_4, and n_3.

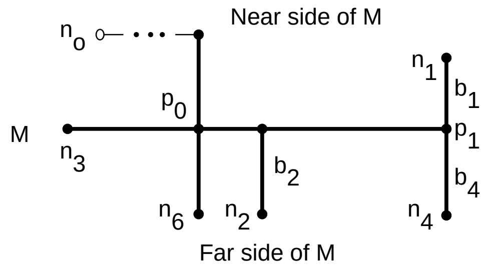

Figure 6: Example of a maximal segment M with entry point p_0, one near branch b_1, and three far branches. (Note that n_3 forms a far branch without edges and that edge (p_0, n_6) is *not* a far branch off of M).

Lemma 2: Let M be an MS in T^* not containing n_0. Then $|Far(M)| > |Near(M)|$.

Proof Sketch: Figure 7 shows that $|Far(M)| \geq |Near(M)|$: If $S \subseteq M$ is the smallest subsegment of M with M's entry point q_0 as an endpoint and with $|Far(S)| < |Near(S)|$, then S can be shifted to S' as in the figure, thereby reducing delay at some sinks while leaving delay at the others unchanged. Suppose that $|Far(M)| = |Near(M)|$ and that no subsegment S of M containing q_0 has $|Far(S)| < |Near(S)|$. Then Figure 8 shows how M can be shifted to M' so as to reduce delay at some sinks without increasing delay at any others. □

Lemma 3: Any maximal segment in T^* must contain either a sink or the source.

Proof Sketch: (See Figure 9.) Let M be a maximal segment in T^* not containing a pin and such that every MS below M topologically *does* contain a sink. Without loss of generality, assume that M is a vertical segment. Coordinates x_1 and x_2 in Figure 9 represent positions of M which would intersect nodes below M in the tree topologically (i.e., p_1 and p_2); x_0 represents the x coordinate of M. Application of the Elmore formula shows that delay function f is concave in x_0 between x_1 and x_2, and so $x_0 = x_1$ or $x_0 = x_2$ in T^*. If $x_0 = x_1$, then either p_1 is a sink or there is another vertical MS through

[8] We apply the Elmore formula for $t(n_j)$ to three cases of n_j: (i) $n_j \in T^*_a$; (ii) $n_j \in T^*_b$; and (iii) $n_j \in T^* \backslash x$. For case (iii) $t(n_j)$ is linear in x; otherwise it is quadratic in x, with a negative coefficient for x^2.

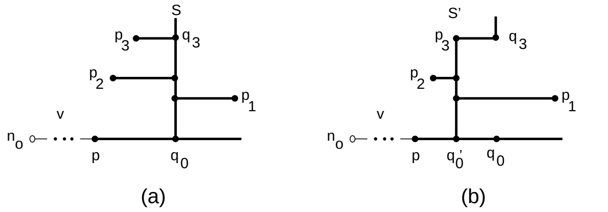

Figure 7: Lemma 2: (a) $|Near(S)| > |Far(S)|$ for segment S between q_0 and q_3; in (b) S is shifted to S' to reduce delay to all sinks in T_{q_0}, leaving all other delays unchanged.

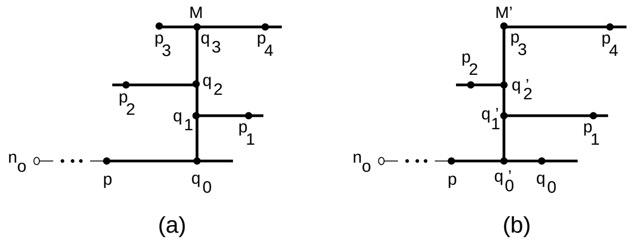

Figure 8: Lemma 2: in (a) $|Near(M)| = |Far(M)|$ for maximal segment M; in (b) M is shifted to M', reducing delay at all sinks in T_{q_0}.

p_1 containing a sink. Similarly for $x_0 = p_2$. Therefore, M must contain a sink if it is in T^*. □

An immediate corollary is a generalization of the classic result of Hanan [11] to the Elmore delay objective.[9]

Corollary 1: Any Steiner node in T^* is located on the Hanan grid. □

3.4.4 Decomposition Theorem for T^*

To prove the optimality of BB-SORT-C, we need to show that an optimal tree T^* can be constructed iteratively from tree $T_0 = \{n_0\}$ by successively adding some ordering of sinks $n_1, n_2, \ldots, n_k$ to create trees $T_1, T_2, \ldots, T_k = T^*$ with each n_i making a closest connection to some edge in tree T_{i-1}. We start with $T^* = T_k$ and successively "peel off" sinks. At each step, we find an interior node $q \in T_i$ whose children are all leaves and peel off one of q's children. Any of q's children may be peeled off *except* $Pin(q)$, which is defined so that pins peeled off later will still make a closest connection to some edge in the current tree (see [4]). Figure 10 gives an example of a possible pin ordering that could be used by the decomposition procedure. In [4] we use this peeling decomposition to prove the following:

[9] Hanan's original theorem may be viewed as a special case of this Corollary with the driver on-resistance $r_d \to \infty$.

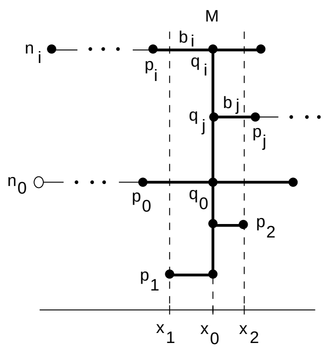

Figure 9: Lemma 3: because delay function f is concave in x_0 for $x_1 \leq x_0 \leq x_2$, f is minimized only when MS M passes through either x_1 or x_2.

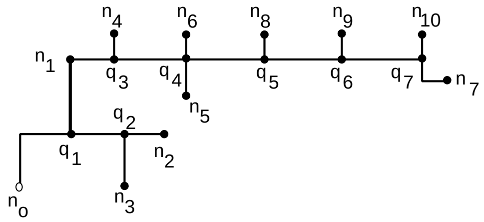

Figure 10: Example of a peeling decomposition of T^*. (Sinks n_i are removed in reverse order of their subscripts.)

Theorem 1: There exists a sequence of subtrees $T_0 = \{n_0\}, T_1, T_2, \ldots, T_k = T^*$ such that for each i, $1 \leq i \leq k$, (i) there is a sink $n_i \in T_i$ such that $T_{i-1} = T_i \backslash n_i$, and (ii) either edge $(n_0, n_i) \in T_i$ or n_i makes a closest connection to some edge in T_{i-1}. □

Corollary 2: BB-SORT-C is optimal for any positive linear combination of sink delays. □

4 Implications: Steiner ERT's Are Near-Optimal

We have implemented BB-SORT and BB-SORT-C in C on a Sun SPARC I ELC workstation, and compared them to the SERT and SERT-C heuristics of [2] and the 1-Steiner algorithm of [12]. Our results use four typical IC and MCM technologies (Table 1).

Name	IC1	IC2	IC3	MCM
Technology	2.0 μm	1.2 μm	0.5 μm	MCM
r_d (Ω)	164.0	212.1	270.0	25.0
unit R ($\Omega/\mu m$)	0.033	0.073	0.112	0.008
unit C ($fF/\mu m$)	0.234	0.0826	0.039	0.06
loading C (fF)	5.7	7.06	1.0	1000
chip size (cm^2)	1x1	1x1	1x1	10x10

Table 1: Technology parameters for three CMOS IC technologies and an MCM technology. IC1 and IC2 parameters are provided by MOSIS; IC3 is courtesy of MCNC; MCM parasitics are courtesy of Prof. Wayne W.-M. Dai of UC Santa Cruz from data provided by AT&T Microelectronics Division.

4.1 Near-Optimality of SERT-C

Table 2 compares Elmore delay of trees constructed by the SERT-C algorithm and optimal Elmore delay trees found by BB-SORT-C for each of the four technologies. Net sizes range from five to nine pins, limited by the exponential running time of BB-SORT-C. The table indicates that any future Elmore delay improvement by Steiner tree heuristics will be limited to between 0.0% and 4.9% for 5-pin nets and between 0.1% and 15.8% for 9-pin nets.

		Number of Pins				
		5	6	7	8	9
SERT-C Delay	IC1	4.2	6.2	8.3	10.5	11.2
	IC2	4.9	7.9	11.4	13.4	15.8
	IC3	4.6	7.8	11.2	13.5	15.7
	MCM	0.00	0.04	0.07	0.09	0.11
1-Stein Delay	IC1	11.7	15.4	20.1	22.9	26.1
	IC2	22.8	28.6	36.2	40.8	45.9
	IC3	27.5	34.1	42.9	48.1	54.1
	MCM	45.5	70.9	63.4	69.3	76.0
SORT-C Cost	IC1	11.1	11.5	12.4	11.8	12.2
	IC2	16.1	15.8	15.8	15.3	15.5
	IC3	17.5	17.1	16.5	16.2	16.1
	MCM	29.6	27.6	25.6	25.3	23.2
Run Time (sec)	SERT-C	.0004	.0006	.0008	.0010	.0012
	1-Stein	.0025	.0046	.0074	.011	.020
	BB-SORT-C	.006	.046	0.46	5.6	36.3

Table 2: Percent above optimum of Elmore delay to a single critical sink and wire length for three Steiner tree constructions (cost comparison is with 1-Steiner). Averages were taken over 200 random nets for each net size.

4.2 Elmore-Optimality of "Generic" SERT Algorithm

The counter-example in Section 3.3 showing that BB-SORT is not always optimal was carefully constructed by hand; even then, BB-SORT was only 0.06% above optimal. Thus, we believe that BB-SORT is within one percent of optimal in essentially all cases. In Table 3 we compare SERT and 1-Steiner with the "SORT" trees of BB-SORT. It appears that the SERT constructions are very nearly optimal: the worst case occurs for IC2 and IC3 for $|N| = 9$, where SERT delays are only 3.9% above those of BB-SORT.

		Number of Pins				
		5	6	7	8	9
SERT Delay	IC1	1.5	2.0	2.6	2.9	3.5
	IC2	1.3	1.6	2.4	3.2	3.9
	IC3	1.1	1.7	2.4	3.1	3.9
	MCM	1.5	2.1	2.7	3.3	3.7
1-Stein Delay	IC1	7.9	10.4	15.4	16.9	19.8
	IC2	13.1	16.6	24.1	26.8	31.0
	IC3	15.1	19.0	27.3	30.3	35.1
	MCM	43.6	52.9	65.1	71.2	78.5
SORT Cost	IC1	4.9	6.2	8.3	9.9	10.2
	IC2	10.6	12.0	13.6	14.4	16.2
	IC3	11.3	13.4	15.2	16.4	17.6
	MCM	50.7	70.6	80.5	89.6	97.5
Run Time (sec)	SERT	.0014	.0030	.0056	.010	.016
	BB-SORT	.015	.13	1.3	14.4	61.8

Table 3: Percent above optimum of maximum sink Elmore delay and wire length for three Steiner tree constructions (cost comparison is with 1-Steiner). 200 random nets are used for each net size.

5 Conclusions

Two main theoretical results show that the BB-SORT-C branch-and-bound method can be used to find Steiner trees that are *optimal* for any linear combination of sink Elmore delays. Our first result is a generalization of Hanan's theorem [11] to Elmore delay. We then establish a new decomposition theorem for optimal Elmore-delay trees. When the objective is to minimize the maximum Elmore delay in a net, we give a counter-example for which our BB-SORT does *not* return the optimal tree. Nevertheless, we believe that BB-SORT will almost always return a tree well within one percent of optimal.

BB-SORT-C and BB-SORT may be used for routing small nets; a more far-reaching implication of our results lies in delineating the *achievable space* of performance-driven routing solutions. Our simulations for the SERT-C heuristic of [2] indicate that it is within 5% of optimal on average for 5-pin nets and within 16% on average for 9-pin nets. The "generic" SERT constructions appear to be even closer to optimal (within 1.5% for $|N| = 5$ and 4% for $|N| = 9$).

6 Acknowledgements

We are grateful to Mr. Ashok Vittal of UC Santa Barbara for helpful comments on an earlier draft. Part of this work was performed during a sabbatical visit to UC Berkeley; support from NSF MIP-9117328 and the hospitality of Professor Ernest S. Kuh and his research group is gratefully acknowledged.

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 Global Routing", *technical report* CSD-920051, UCLA Department of Computer Science, 1992.

[2] K.D. Boese, A. B. Kahng and G. Robins, "High-Performance Routing Trees with Identified Critical Sinks", *Proc. ACM/IEEE Design Automation Conf.*, June 1993, pp. 182-187.

[3] K. D. Boese, A. B. Kahng, B. A. McCoy and G. Robins, "Fidelity and Near-Optimality of Elmore-Based Routing Constructions", *Proc. IEEE Intl. Conf. on Computers and Processors*, October 1993, pp. 81-84.

[4] K.D. Boese, A. B. Kahng, B. A. McCoy and G. Robins, "Near-Optimal Critical Sink Routing Tree Constructions", *technical report TR-930027*, UCLA CS Department, 1993.

[5] J. P. Cohoon and L. J. Randall, "Critical Net Routing", *Proc. IEEE Intl. Conf. on Computer Design*, 1991, pp. 174-177.

[6] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, "Provably Good Performance-Driven Global Routing", *IEEE Trans. on CAD* 11(6), June 1992, pp. 739-752.

[7] J. Cong, K.-S. Leung and D. Zhou, "Performance-Driven Interconnect Design Based on Distributed RC Delay Model", *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 606-611.

[8] W. E. Donath, R. J. Norman, B. K. Agrawal, S. E. Bello, S. Y. Han, J. M. Kurtzberg, P. Lowy and R. I. McMillan, "Timing Driven Placement Using Complete Path Delays", *Proc. ACM/IEEE Design Automation Conf.*, 1990, pp. 84-89.

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

[10] W. C. Elmore, "The Transient Response of Damped Linear Network with Particular Regard to Wideband Amplifiers", *J. Applied Physics* 19 (1948), pp. 55-63.

[11] M. Hanan, "On Steiner's Problem with Rectilinear Distance", *SIAM J. Appl. Math.*, 14 (1966), pp. 255-265.

[12] A. B. Kahng and G. Robins, "A New Class of Iterative Steiner Tree Heuristics with Good Performance", *IEEE Transactions on CAD* 11(7), July 1992, pp. 893-902.

[13] S. Kim, R. M. Owens and M. J. Irwin, "Experiments with a Performance Driven Module Generator", *Proc. ACM/IEEE Design Automation Conf.*, 1992, pp. 687-690.

[14] A. Prim, "Shortest Connecting Networks and Some Generalizations", *Bell System Tech. J.* 36 (1957), pp. 1389-1401.

[15] S. K. Rao, P. Sadayappan, F. K. Hwang and P. W. Shor, "The Rectilinear Steiner Arborescence Problem", *Algorithmica* 7 (1992), pp. 277-288.

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

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

[18] R. S. Tsay, "Exact Zero Skew", *Proc. IEEE Intl. Conference on Computer-Aided Design*, 1991, pp. 336-339.

[19] J. Vlach, J. A. Barby, A. Vannelli, T. Talkhan and C. J. Shi, "Group Delay as an Estimate of Delay in Logic", *IEEE Transactions on Computer-Aided Design*, 10(7), 1991, pp. 949-953.