

Timing Optimization for Multi-Source Nets: Characterization and Optimal Repeater Insertion*

John Lillis
Dept. of EECS, U.C. Berkeley
Berkeley, CA 94720

Chung-Kuan Cheng
CSE Dept., U.C.S.D.
La Jolla, CA 92093

Abstract

This paper presents new results in the area of timing optimization for multi-source nets. The Augmented RC-Diameter (ARD) is proposed as a natural and practical performance metric and a linear time algorithm for computing the ARD of a multi-source net is presented. Building on the ARD, an algorithm for optimal repeater insertion is presented: for a given multi-source topology the algorithm efficiently identifies an optimal assignment of repeaters to prescribed insertion points under the "min cost timing feasible" problem formulation. The algorithm has been implemented and preliminary experimental results are promising.

1 Introduction

Due to a technological shift in the relative importance of interconnect delay and logic delay, recent years have seen much research in automatic timing optimization for VLSI interconnect. Most of this work has focused on *single-source* interconnect optimization and includes buffer insertion (e.g., [16], [5], [10]), performance driven topology synthesis (e.g., [11]), and wire sizing (e.g., [2], [10], [14]).

After considering the single-source case, it is natural to ask what can be done for multi-source nets since buses are so prevalent in modern designs and are often a key determinant of system performance. This topic has only recently received attention (e.g., [4], [3], [15]) and is the topic of this paper. Our primary contributions are as follows.

- (1) We propose the *Augmented RC-Diameter (ARD)* as a natural timing metric for multi-source optimization.
- (2) We demonstrate that the ARD can be computed in $O(n)$ time and thus is no harder than computing an *RC-radius* (e.g., [13]).
- (3) We present an algorithm for a promising optimization technique in this area: optimal bi-directional repeater insertion. We adopt the objective of cost minimization

*This research was funded in part by grants from NSF project numbers MIP-9529077 and MIP-9625910, and the California MICRO Program.

Permission to make digital/hard copy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 97, Anaheim, California
(c) 1997 ACM 0-89791-920-3/97/06 ..\$3.50

subject to a given timing spec. The technique produces a suite of solutions giving a cost vs. performance trade-off and also subsumes the discrete driver sizing problem.

We focus on repeater insertion in part because the decoupling effect of repeaters makes the performance to cost benefit of the technique extremely favorable. Further, it has been shown to be a feasible design technique in the high-end community and, in fact, is commonly used in the design of high-speed commercial microprocessors [12].

Nevertheless, some design and technology issues arise in to bi-directional repeater insertion which do not appear in the single-source case. A primary issue is repeater design: should one use a pair of tri-state buffers controlled by a bus arbiter¹ or more sophisticated "direction sensing" autonomous repeaters (e.g., [8], [7])? Such decisions are largely dependent on designer preference [12] and are not addressed here. Fortunately, such issues appear to have little impact on the basic nature of the optimization problem and our techniques should remain relevant.

2 Preliminaries and Formulations

Throughout the remainder of this paper, the following are assumed to be given technology parameters.

- α and β which are respectively resistance and capacitance per unit wire length (e.g. 1 micron).
- A library L_b of repeaters. For notational simplicity, we assume that repeaters are *symmetric* (e.g., a repeater's drive capability is independent of signal direction). Thus, the following parameters will characterize a repeater b : **intrinsic delay** $i(b)$, **output resistance** $r(b)$, **input capacitance** $c(b)$ and **cost** $s(b)$.

Clearly, repeaters with *asymmetric* characteristics are of practical value. Generalizations of the presented algorithms to deal with the asymmetric case are straightforward and are used in our implementation.

For completeness, we review the basic delay models. In the typical basic model, the delay of a buffer b is given by $i(b) + r(b) \cdot c_{load}$ where c_{load} is the capacitive load on the b 's output. Similarly in the Elmore model, the delay of a signal transmitted along a wire of length l from u to v is approximated by $\alpha l(\frac{l\beta}{2} + c_v)$ where c_v is the lumped downstream capacitance at v . Subsequently when referring to the delay of a path, these models are assumed.

¹Note that a bus controller is typically necessary whether repeaters are inserted or not and thus the additional control overhead when repeaters are used can reasonably be expected to be modest, particularly when amortized over a multi-bit bus.

In addition to the above technology parameters, the following net-specific parameters are also given.

- Terminal set N in the plane with the following parameters associated with each terminal $v \in N$

$a(v)$: latest arrival time at pin v from a *latch output* (or PI) via a path *not* traversing net N

$d(v)$: maximum delay from pin v to a *latch input* (or PO) via a path *not* traversing net N

$c(v)$: the input capacitance of pin v when acting as a sink

$r(v)$: the output resistance of the driver of pin v when acting as a source

- A Rectilinear Steiner Tree T connecting N with prescribed, degree two, candidate buffer insertion points²

To assess the performance of a routing tree T with a driver assignment and a (perhaps null) buffer assignment we now formally define the Augmented RC-Diameter, $ARD(T)$.

Definition 1 Let $PD(u, v)$ be the RC path delay from source pin u to sink pin v in topology T spanning N including delay of the driver, buffers and wires on the path. We define the Augmented RC Diameter of T , $ARD(T)$ as

$$ARD(T) = \max_{u \in N} \max_{v \in N \setminus \{u\}} (a(u) + PD(u, v) + d(v))$$

The $ARD(T)$ gives the maximum delay from the circuit's PIs to its POs among all paths traversing N and thus, captures the impact of the net on, for example, system cycle time.³

The definition of $ARD(T)$ leads naturally to the *Optimal Repeater Insertion Problem* as follows.

Problem 1 Given routing topology T , a set of technology parameters and performance target D_{spec} , find an assignment of repeaters to the insertion points of T such that the resulting cost is minimized subject to $ARD(T) \leq D_{spec}$.

We present a solution to Problem 1 in Section 4.

3 Linear Time Computation of $ARD(T)$

Clearly, by performing n single-source computations we can compute $ARD(T)$ in $O(n^2)$ time. However, we will show that only $O(n)$ is needed and thus, the problem is no harder than the analogous single-source computation.

We first clarify some additional assumptions and notation. Without loss of generality, we assume that all terminals are also leaves in the topology T . For purposes of depth-first traversal, it will be useful to orient a topology T with respect to an arbitrary root vertex s . In such a rooted tree, we use $p(v)$ to indicate the parent of v ; we also note that edges $(u, v) \in T$ are now directed. We use $c(u, v)$ and $r(u, v)$ to respectively denote the capacitance and resistance of a wire $(u, v) \in T$.

²We assume insertion points are degree two to avoid ambiguity with respect to which side of the repeater a branch connects.

³No generality is lost by not explicitly specifying the sources and sinks of the net; for any non-sink (-source) v , we simply set $d(v) = -\infty$ ($a(v) = -\infty$).

The first step in computing $ARD(T)$ is the computation of capacitive values $c_L(u, v)$ and $c_L(v, u)$ for every edge $(u, v) \in T$ where $c_L(u, v)$ is the total capacitance at v as seen from u and likewise, $c_L(v, u)$ is the load at u from v 's perspective. We first compute $c_L(u, v)$ where $(u, v) \in T$ (i.e., forward edges) by the following recurrence.

$$c_L(u, v) = \begin{cases} c(v) & \text{if } v \text{ is a terminal} \\ c(b) & \text{else if buffer } b \text{ is placed at } v \\ \sum_{(v,w) \in T} c_L(v, w) + c(v, w) & \text{o.w.} \end{cases} \quad (1)$$

After this bottom-up process, we compute $c_L(v, u)$ where $(u, v) \in T$ (i.e., backward edges) top-down as follows.

$$c_L(v, u) = \begin{cases} c(u) & \text{if } u \text{ is the root} \\ c(b) & \text{else if buffer } b \text{ is placed at } u \\ c_L(u, p(u)) + c(u, p(u)) + \sum_{\substack{(u,w) \in T \\ w \neq v}} c_L(u, w) + c(u, w) & \text{o.w.} \end{cases} \quad (2)$$

Once $c_L(u, v)$ and $c_L(v, u)$ have been computed for edges $(u, v) \in T$, the remainder of the algorithm follows as in Fig. 1. In a depth-first traversal, the algorithm computes three values for each subtree T_v : a is the augmented arrival time at v via sources in T_v ; d is the augmented delay from v to sinks in T_v ; and d_I is the Augmented RC-Diameter among source/sink pairs in T_v . Once the root is reached, the ARD of the entire tree is easily derived.

4 The Repeater Insertion Algorithm

As in the previous section, we assume that the routing topology T has been reoriented with an arbitrary terminal as the root. Our repeater insertion algorithm is based on dynamic programming on this reoriented tree. The key component of our algorithm is concise characterization of the multitude of paths in the tree (e.g., in principle, all $O(n^2)$ paths are of interest as opposed to $O(n)$ in the single-source case).

To motivate the issues involved in this characterization, consider the example in Fig. 2. We have arrived at vertex v in our bottom-up traversal and need to be able to assess the maximum arrival time at v from u and w and the internal augmented path delays u to w and w to u . The key is that these values depend on the capacitance of the tree *outside* of T_v (i.e., $T \setminus T_v$) – the *external capacitance*, c_E . To assess the arrival time at v from terminal u , consider the case where $c_E = 0$; calculating the driver delay and interconnect delay of wire (u, v) , we give the arrival time at v from u as

$$a(v, u) = a(u) + 2(2 + 4 + 1) + 5(1 + 4 + 1) = 204.$$

Similarly, the arrival time at v via source w is given by $a(v, w) = 104$ (again, assuming $c_E = 0$). Thus, in general (i.e., for arbitrary c_E), we have

$$a(v, u) = 204 + 7c_E \quad \text{and} \quad a(v, w) = 104 + 12c_E$$

since the path resistance from u to v is 7 and that from w to v is 12. In Fig. 2(c), we see that not only does the arrival time

Routine: ARD(T)

Output: The Augmented RC-Diameter of T

Orient T with an arbitrary pin $s \in N$ as the root
 Compute $c_L(u, v)$ and $c_L(v, u)$ for all $(u, v) \in T$
 $(a, d, d_I) \leftarrow \text{ARD_R}(s)$
 Let $c_L(s)$ be the total load on pin s
 RETURN $\max(a + d(s), d + a(s) + r(s)c_L(s), d_I)$

Routine: ARD_R(v)

Output: triple (a, d, d_I) where

$$a = \max_{u \in T_v \cap N} a(u) + PD(u, v)$$

$$d = \max_{u \in T_v \cap N} PD(v, u) + d(u)$$

$$d_I = \max_{u \in T_v \cap N} \max_{w \in T_v \cap N, w \neq u} a(u) + PD(u, w) + d(w)$$

IF (v is a leaf)

$a \leftarrow a(v) + r(v)(c_L(v, p(v)) + c(v, p(v)))$
 RETURN $(a, d(v), -\infty)$

ELSE

$a \leftarrow d \leftarrow d_I \leftarrow -\infty$

FOREACH child u of v

$(a(u), d(u), d_I(u)) \leftarrow \text{ARD_R}(u)$

$a'(u) \leftarrow a(u) + r(u, v)(\frac{c(u, v)}{2} + c_L(u, v))$

$d'(u) \leftarrow d(u) + r(v, u)(\frac{c(v, u)}{2} + c_L(v, u))$

IF (buffer b placed at v)

$a'(u) \leftarrow a'(u) + r(b)(c_L(v, p(v)) + c(v, p(u)))$

$d'(u) \leftarrow d'(u) + r(b)(c_L(v, u) + c(v, u))$

$d_I \leftarrow \max(d_I, a + d'(u), d + a'(u), d_I(u))$

$a \leftarrow \max(a, a'(u))$

$d \leftarrow \max(d, d'(u))$

RETURN (a, d, d_I)

Figure 1: *Computation of ARD(T)*

depend on c_E , but so does the *critical source* - w dominating when $c_E < 20$ and u dominating when $c_E > 20$. Thus, the piece-wise *maximum* of $a(v, u)$ and $a(v, w)$ shown in Fig. 2(c) captures the arrival time at v from its descendants as a function of external capacitance c_E .

We must also consider the *internal paths* (u, v, w) and (w, v, u) as in Fig. 2(d). The augmented RC delay of path (u, v, w) includes $a(v, u)$, the scalar delay from v to sink w ($PD(v, w)$), and $d(w)$. Thus, noting that $PD(v, w) = 30$ and $d(w) = 20$, we have

$$d(u, v, w) = a(v, u) + PD(v, w) + d(w) = 254 + 7c_E.$$

Similarly, $d(w, v, u) = 274 + 12c_E$. These are shown by dashed lines in part (d) of the figure and amount to adding scalars to the y -intercepts of $a(v, u)$ and $a(v, w)$. Taking the piece-wise maximum of $d(u, v, w)$ and $d(w, v, u)$ gives us the *internal augmented RC diameter* of T_v (in this case $d(w, v, u)$ dominates $d(u, v, w)$ for all values of c_E .)

This discussion gives an idea of our approach: as we proceed bottom-up, we accumulate path resistance in the slopes of PWL functions and apply appropriate PWL operators when considering options such as repeater insertion.

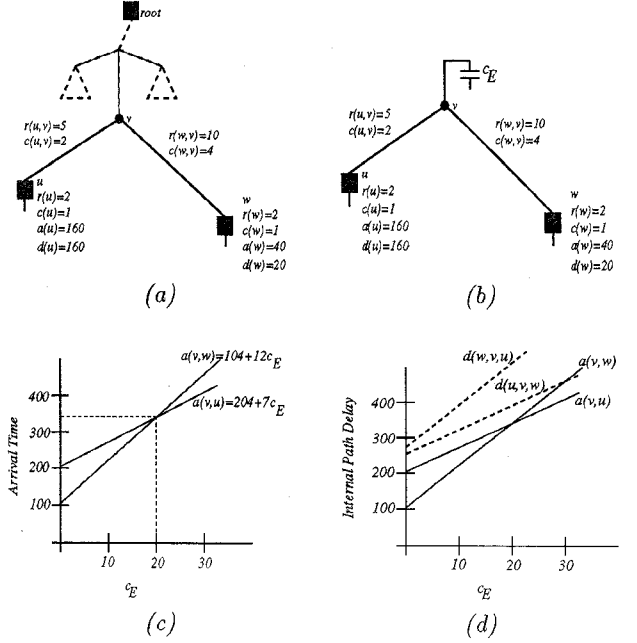


Figure 2: (a) *Example bottom-up computation*, (b) *abstraction of external capacitance c_E* , (c) *corresponding arrival time functions at v from u and w* , (d) *and augmented delay of internal paths*

4.1 Solution Characterization

A solution (repeater assignment) s for subtree T_v is fully parameterized as follows (using C-like structure syntax).

$s.cost$	scalar cost
$s.cap$	scalar capacitance
$s.d$	scalar augmented delay to sinks in T_v
$s.f_A$	PWL giving arrival time at v from sources in T_v as a function of c_E
$s.f_I$	PWL giving augmented RC diameter for source/sink pairs internal to T_v

The first three parameters are inherited from the single-source problem (see [10]) while the two PWL functions are unique to the multi-source problem. For each vertex v in the reoriented tree we compute a set $S(v)$ of *option* solutions for T_v where each $s \in S(v)$ is characterized as above.

4.2 PWL Primitives

In this section we define some basic PWL operators. We first give a formal definition of a PWL itself.

Definition 2 A piece-wise linear (PWL) function f is a set of quadruples $(y_0, slope, x_l, x_r)$ where each such quadruple represents a line segment and gives the y -intercept (y_0), slope and domain for which the segment is defined (i.e., all x where $x_l \leq x \leq x_r$). Additionally, no two segments may have overlapping domains.

As is typical in most applications of PWL functions, we store line segments in a linked list ordered by domain.

The following are the PWL primitives we require.

$$\begin{aligned}
f &\leftarrow \text{PWL_Max}(f_1, f_2) \\
&\iff f(x) = \max(f_1(x), f_2(x)) \quad \forall x \\
f &\leftarrow \text{PWL_AddScalar}(f_1, dy) \\
&\iff f(x) = f_1(x) + dy \quad \forall x \\
f &\leftarrow \text{PWL_ShiftLeft}(f_1, dx) \\
&\iff f(x) = f_1(x + dx) \quad \forall x \\
f &\leftarrow \text{PWL_AddSlope}(f_1, ds) \\
&\iff f(x) = f_1(x) + x \cdot ds \quad \forall x
\end{aligned}$$

Each of these operations is performed in time linear in the number of segments in the participating functions by appropriately stepping through the segment lists.

4.3 Solution Dominance

As in most multi-dimensional dynamic programming applications, a key issue in repeater insertion is the elimination of suboptimal solutions. Typically, a solution s is characterized by a set of k scalars $s[1], s[2], \dots, s[k]$; assume w.l.o.g. that minimization in each of these k dimensions is preferred. This induces a partial order \preceq on a set of solutions S ; for $s_1, s_2 \in S$, we have the following

$$s_1 \preceq s_2 \iff s_1[i] \leq s_2[i] \quad \forall i \in \{1..k\}.$$

We need not consider any solution $s_2 \in S$ if there exists $s_1 \in S$ such that $s_1 \preceq s_2$ (assume w.l.o.g. all solutions are distinct). Thus, for a given solution set S , we compute the *minima* of S and discard the remainder.

Definition 3 *The minimal or dominant subset of a set of k -dimensional points S is the largest subset $S' \subseteq S$ such that, for all $s_1, s_2 \in S'$, $s_1 \not\preceq s_2$ and $s_2 \not\preceq s_1$.*

The problem of finding a minimal point set has a long history dating back at least to [9]. A variety of algorithms have been proposed; some achieve excellent asymptotic complexity (e.g. [9]) while others have been tuned for ease of implementation or fast *expected* run time (e.g., [1]).

The reader may have noted that because of the PWL functions $s.f_A$ and $s.f_I$ a subsolution s does not correspond to a single point in k -dimensional space; rather, it corresponds to an infinite set of 5-dimensional points – one for every value of c_E . To deal with this complication we introduce the notion of a *minimal functional subset (MFS)*.⁴

Definition 4 *Let S be a set where each $s \in S$ is a k -tuple of PWL functions of a variable x . For each $s \in S$, let s' be defined (for $i \in \{1..k\}$) as*

$$s'[i](x) = \begin{cases} \infty & \text{if } \exists s_0 \in S \text{ s.t. } s_0 \neq s \text{ and} \\ & s_0[j](x) \leq s[j](x) \quad \forall j \in \{1..k\} \\ s[i](x) & \text{o.w.} \end{cases}$$

The minimal functional subset (MFS) of S is S' :

$$S' = \{s' | s \in S \text{ and } \exists x \text{ s.t. } s'[i](x) \neq \infty\}$$

Note that PWL functions in S' may no longer be “connected” since they may have suboptimal regions of the domain (i.e., ∞) separating useful regions of the domain. Finally, we point out that the fundamental pruning operation

⁴We use the notion of *subset* somewhat loosely here, however, the meaning should be clear.

of the definition – for a given s and s_0 , detect all ranges of x for which s_0 dominates s and alter s accordingly – can be implemented in time linear in k and the total number of segments in question. This is done by first detecting the regions in which $s_0[i]$ dominates $s[i]$ (similar to a *PWL_Max* operation) for each i and then computing the intersection of these regions and updating the PWLs accordingly.

In our application, solutions are characterized by three scalars (degenerate PWL functions) and two PWL functions. At each stage of dynamic programming, we would like to efficiently compute the FMS of a solution set S . We have devised an easy to implement algorithm for solving this problem which appears to work well in practice; it appears in pseudo-code in Fig. 3. The algorithm takes a divide and conquer approach motivated by the following intuition. Suppose that the FMS of S is much smaller than S itself.⁵ The idea is that many of the suboptimal solutions will be discarded at relatively deep levels of the recursion thus avoiding excessive pair-wise comparisons at higher levels of the recursion. Thus, the approach targets fast performance in practice but retains an $\Omega(|S|^2)$ worst case complexity in terms of the number of pair-wise comparisons.

Routine: MinimalFS_DC(S)

Output: The minimal functional subset of S

1. IF ($|S| = 1$)
2. RETURN S
3. Divide S into equal sized sets S_l and S_r
4. $S'_l \leftarrow \text{MinimalFS_DC}(S_l)$
5. $S'_r \leftarrow \text{MinimalFS_DC}(S_r)$
6. Compute $S' = \text{MFS}(S'_l \cup S'_r)$ by pair-wise comparison between S'_l and S'_r
7. RETURN S'

Figure 3: *Computing MFS(S) by Divide & Conquer*

4.4 The Overall Algorithm

We present a high-level view of our algorithm for optimal repeater insertion in Fig. 4. We recursively compute solution sets $S(v)$ from those of v 's children. The top-level code takes different actions depending on whether the current vertex v is a leaf, Steiner (branch) node, insertion point or the root itself; supporting subroutines appear in Fig. 5.

The algorithm follows the intuition presented in the preceding sections. Consider the computation of $s.f_A$ by *Join-Sets* in Fig. 5. Here we are joining two subtrees and the capacitance of each subtree will be seen by signals arriving at the root of the other tree. This gives rise to the *PWL_ShiftLeft* operators to compute f_A^1 and f_A^2 , the arrival time functions at the root *with respect to the external capacitance of the newly formed tree*. Taking their *PWL_Max* gives $s.f_A$. Computing $s.f_I$ is slightly more complex. We must consider not only pre-existing internal paths, but also internal paths traversing the newly formed root v . The effect of these newly formed paths is captured in f_I^{12} and f_I^{21} which give the augmented diameters of paths through v starting

⁵From experience, this is frequently the case, particularly when constructing solutions at a branch point

Routine: MSRI(v)
Purpose: Computes the minimal solution set $S(v)$ among repeater assignments to T_v

```

IF ( $v$  is a leaf)
   $s.cost \leftarrow 0$  ;  $s.cap \leftarrow c(v)$  ;  $s.d \leftarrow d(v)$ 
   $s.f_A \leftarrow a(v) + r(v) c_E$ 
   $s.f_I \leftarrow -\infty$  /* No internal path */
   $S(v) \leftarrow \{s\}$ 
ELSE IF ( $v$  is a Steiner node)
   $S(v) \leftarrow \emptyset$ 
  FOREACH (child  $u$  of  $v$ )
    MSRI( $u$ )
     $S'(u) \leftarrow \text{Augment}(S(u), (v, u))$ 
    IF ( $S(v) \neq \emptyset$ )
       $S(v) \leftarrow \text{JoinSets}(S(v), S'(u))$ 
       $S(v) \leftarrow \text{MinimalFS\_DC}(S(v))$ 
    ELSE
       $S(v) \leftarrow S'(u)$ 
ELSE IF ( $v$  is a candidate insertion point (degree-2))
  Let  $u$  be the child of  $v$ 
   $S'(u) \leftarrow \text{Augment}(S(u), (v, u))$ 
   $S(v) \leftarrow S'(u) \cup \text{RepeaterSolutions}(S'(u))$ 
   $S(v) \leftarrow \text{MinimalFS\_DC}(S(v))$ 
ELSE /*  $v$  is the root */
  Let  $u$  be the child of  $v$ 
   $S'(u) \leftarrow \text{Augment}(S(u), (v, u))$ 
  Compute global ARD for each  $s \in S'(u)$ 
  by pairing  $s$  with driver parameters.
  Keep only minima w.r.t. cost and ARD
  giving final tradeoff curve

```

Figure 4: *Top-level algorithm for Optimal Multi-Source Repeater Insertion*

in subtree 1 and subtree 2 respectively. Ultimately, $s.f_I$ is computed by a *PWL_Max* over pre-existing and new internal paths. The other subroutines follow similar reasoning.

Optimality of the algorithm follows inductively from its bottom-up structure.

Theorem 1 Consider $S(v)$ as computed by the MSRI algorithm in Fig. 4. Let $s(c_E)$ be the 5-tuple of scalars $(s.cost, s.cap, s.d, s.f_A(c_E), s.f_I(c_E))$. We conclude that $s \in S(v) \iff \exists c_E$ s.t. \forall repeater assignments $s' \neq s$ to T_v , $s'(c_E) \not\prec s(c_E)$

4.5 Discussion

A few additional issues relating to the algorithm are worth mentioning. First, by *discretizing* the dimensions of problem, it is possible to give a *pseudo-polynomial* run-time bound. However, the usefulness of such a bound is limited by its inherent pessimism; therefore, we defer to experimental results for a more meaningful run-time evaluation. Second, the algorithm also subsumes the driver sizing problem. Lastly, an extension allowing the use of inverters as repeaters is straightforward.

5 Experiments

We have implemented prototypes of our algorithms on a Sun SPARC 10 workstation and we report preliminary re-

Routine: JoinSets(S_1, S_2)
Input: Solution sets S_1 and S_2 from disjoint subtrees
Output: The non-minimal solution set S when the subtrees are joined at a common parent

```

 $S \leftarrow \emptyset$ 
FOREACH (pair of solutions  $s_1 \in S_1, s_2 \in S_2$ )
   $s.cost \leftarrow s_1.cost + s_2.cost$ 
   $s.cap \leftarrow s_1.cap + s_2.cap$ 
   $s.d \leftarrow \max(s_1.d, s_2.d)$ 
   $f_A^1 \leftarrow \text{PWL\_ShiftLeft}(s_1.f_A, s_2.cap)$ 
   $f_A^2 \leftarrow \text{PWL\_ShiftLeft}(s_2.f_A, s_1.cap)$ 
   $s.f_A \leftarrow \text{PWL\_Max}(f_A^1, f_A^2)$ 
   $f_I^{12} \leftarrow \text{PWL\_AddScalar}(f_A^1, s_2.d)$ 
   $f_I^{21} \leftarrow \text{PWL\_AddScalar}(f_A^2, s_1.d)$ 
   $f_I^v \leftarrow \text{PWL\_Max}(f_I^{12}, f_I^{21})$ 
   $f_I^1 \leftarrow \text{PWL\_ShiftLeft}(s_1.f_I, s_2.cap)$ 
   $f_I^2 \leftarrow \text{PWL\_ShiftLeft}(s_2.f_I, s_1.cap)$ 
   $s.f_I \leftarrow \text{PWL\_Max}(f_I^v, \text{PWL\_Max}(f_I^1, f_I^2))$ 
   $S \leftarrow S \cup \{s\}$ 
RETURN  $S$ 

```

Routine: RepeaterSolutions(S)
Output: Solution set in which all solutions are buffered with a repeater from the library L_b

```

 $S' \leftarrow \emptyset$ 
FOREACH ( $s \in S$  and  $b \in L_b$ )
   $s'.cost \leftarrow s.cost + s(b)$ 
   $s'.cap \leftarrow c(b)$ 
   $s'.d \leftarrow s.d + i(b) + r(b) \cdot s.cap$ 
   $a \leftarrow s.f_A(c(b)) + i(b)$ 
   $s'.f_A \leftarrow a + r(b) \cdot c_E$ 
   $s'.f_I \leftarrow s.f_I(c(b)) + 0c_E$ 
   $S' \leftarrow S' \cup \{s'\}$ 
RETURN  $S'$ 

```

Routine: Augment($S, (u, v)$)
Output: Solution set built from S taking wire (u, v) into account

```

 $S' \leftarrow \emptyset$ 
FOREACH ( $s \in S$ )
   $s'.cost \leftarrow s.cost$ 
   $s'.cap \leftarrow s.cap + c(u, v)$ 
   $s'.d \leftarrow s.d + r(u, v) \left( \frac{c(u, v)}{2} + s.cap \right)$ 
   $f'_A \leftarrow \text{PWL\_AddScalar}(s.f_A, r(u, v) \left( \frac{c(u, v)}{2} \right))$ 
   $s'.f_A \leftarrow \text{PWL\_AddSlope}(f'_A, r(u, v))$ 
   $s'.f_I \leftarrow \text{PWL\_ShiftLeft}(s.f_I, c(u, v))$ 
   $S' \leftarrow S' \cup \{s'\}$ 
RETURN  $S'$ 

```

Figure 5: *Subroutines of MSRI()*

sults in this section. The technology parameters are summarized as follows: **unit wire resistance and capacitance** are $\alpha = 0.12\Omega/\mu m$ and $\beta = 0.15fF/\mu m$ respectively; a “unit” unidirectional buffer/driver b_1 has **cost** $s(b_1) = 1$, **input capacitance** $c(b_1) = 0.05pF$ and **output resistance** $r(b_1) = 800\Omega$. Larger buffers were formed by a scaling

N	Avg. # i-points	D.S.: Min ARD		R.I.: Min Cost s.t.	R.I.: Min ARD		Avg. CPU (sec.)	
		D_{ds}	Cost	$ARD \leq D_{ds}$	D	Cost	D.S.	R.I.
10	30.4	0.73	1.45	1.18	0.55	1.73	0.9	6.9
20	36.6	0.77	1.26	1.06	0.44	1.51	6.0	28.9

Table 1: *Experimental Results. ARD and cost normalized to min-cost solution (min-sized drivers, no repeaters).*

factor k : a kX buffer b_k is parameterized by $s(b_k) = k$, $i(b_k) = 100ps$ (independent of size), $c(b_k) = k(0.05pF)$ and $r(b_k) = (800/k)\Omega$. These parameters are typical of those seen in the literature for sub-micron technology.

Experimental results appear in Table 1. In this set of experiments, we assume that all terminals serve as both sources and sinks and that all arrival times and downstream delay times are 0 (e.g., all terminals connect to latches). Ten random 10 terminal point sets were drawn from a $1cm \times 1cm$ grid; we did the same for 20 terminals. Steiner trees connecting the terminals were then generated using the P -Tree_A algorithm [11]. Repeater insertion points were then added to the topology such that consecutive insertion points were no more than approximately $800\mu m$ apart.

The repeater insertion algorithm was then applied to the topology using the repeater formed by a pair of $1X$ buffers. We then ran the algorithm in *driver sizing* mode on the same topology using a driver library built from $1X$, $2X$, $3X$ and $4X$ buffers, effectively giving a library of 16 drivers when orientation is considered. Additionally, it was assumed in both cases that the previous stage resistance of each terminal was 400Ω and that the subsequent stage capacitance of each terminal was $0.2pF$.

As can be seen from columns 3 and 7, far more substantial reductions in RC-diameter were observed for repeater insertion. Additionally, when the minimum RC-diameter achievable by driver sizing is used as a constraint for repeater insertion, we observe substantially lower cost overhead for equivalent or better diameter as shown by columns 4 and 5. Finally the average CPU times are given in the last two columns of the same table. As stated in the previous section, empirical evidence is the best way to judge the tractability of algorithms such as those proposed here. Indeed, the achieved run-times for our largely unoptimized prototype code are quite reasonable. Since the algorithm assures optimality, evidence of its tractability is a primary contribution of this paper.

6 Conclusions/Comments

It is clear that interconnect optimization will play a central role in future CAD tools for physical design; optimization of multi-source nets is no exception to this trend. As such, we propose that our results are of fundamental interest. First, the notion of the Augmented RC-Diameter gives a natural and practical performance metric for multi-source optimization. Second, we have proposed algorithm for optimal repeater insertion demonstrating that it is indeed a tractable problem. Natural topics for further research include a more detailed assessment of control overhead and application of similar techniques to multi-source topology synthesis.

In closing we thank Professor Majid Sarrafzadeh of

Northwestern University for pointing us to the point dominance literature and Dr. Kenneth Clarkson of AT&T Bell Labs for pointing out reference [1] and additional input on point dominance problems.

References

- [1] Clarkson, K.L. "More output-sensitive geometric algorithms," *Proc. 35th Annual Symposium on Foundations of Computer Science*, 1994 pp. 695-702
- [2] J.J. Cong, K.S. Leung, "Optimal Wiresizing Under Elmore Delay Model," *IEEE Trans. on CAD*, v. 14 no. 3 (1995) pp. 321-336.
- [3] J.J. Cong, P.H. Madden, "Performance driven routing with multiple sources," *Proc. IEEE Symposium on Circuits and Systems*, 1995 pp. 203-6
- [4] J.J. Cong, L. He, "Optimal wiresizing for interconnects with multiple sources," *Proc. Intl. Conf. Computer-Aided Design*, 1995 pp. 568-74
- [5] S. Dhar, "Delay Optimization Algorithms For Tree Models of MOS Circuits," Ph.D dissertation, Dept. of Electrical Engineering, Washington University Stevens Institute of Technology, 3 September 1987.
- [6] W.C. Elmore, "The Transient Response of Damped Linear Network with particular Regard to Wideband Amplifiers," *J. Applied Physics* 19 (1948), pp 55-63.
- [7] T. B. Huang, Y.C. Jeng et, al "Bidirectional Bus Repeater," U.S. Patent # 5,202,593, Apr. 13, 1993.
- [8] D.Y. Kao, C.C. Tsai, C.-K. Cheng, T.T. Lin, "New Design and Implementation for Signal Repeaters," *VLSI/CAD workshop*, Taiwan, pp. 173-176, Aug. 17-19, 1995.
- [9] H.T. Kung, F. Luccio, F.P. Preparata. "On finding the maxima of a set of vectors," *J. ACM* 22, 4 Oct. 1975, pp. 469-76
- [10] J. Lillis, C.-K. Cheng, T.-T. Lin, "Optimal Wire Sizing and Buffer Insertion for Low Power and a Generalized Delay Model," *IEEE Journal of Solid State Circuits*, vol. 31, no. 3, March 1996, pp. 437-447.
- [11] J. Lillis, C.-K. Cheng, T.-T. Lin, C.-Y. Ho, "New Techniques for Performance Driven Routing with Explicit Area/Delay Tradeoff and Simultaneous Wire Sizing," *Proc. Design Automation Conference*, Las Vegas, Jun. 1996, pp. 395-400.
- [12] N. Menezes, Intel Corp., *private communication*, Oct. 1996.
- [13] J. Rubinstein, P. Penfield, and M.A. Horowitz, "Signal Delay in RC Tree Networks," *IEEE Trans. on CAD* 2(3) (1983), pp 202-211.
- [14] S.S. Sapatnekar, "RC Interconnect Optimization under the Elmore Delay Model," *Proc. Design Automation Conf.*, 1994, pp. 387-391.
- [15] C.C. Tsai, D.Y. Kao, and C.-K. Cheng, "Performance Driven Bus Buffer Insertion," *IEEE Trans. on CAD*, April 1996, pp. 429-437
- [16] L.P.P.P van Ginneken, "Buffer Placement in Distributed RC-tree Networks for Minimal Elmore Delay," *Proc. International Symposium on Circuits and Systems*, 1990, pp 865-868.