Optimal Wiresizing for Interconnects with Multiple Sources *

Jason Cong and Lei He
Department of Computer Science
University of California, Los Angeles, CA 90095
cong@cs.ucla.edu helei@cs.ucla.edu

Abstract *The optimal wiresizing problem for nets with multiple sources is studied under the distributed Elmore delay model. We decompose such a net into a source subtree (SST) and a set of loading subtrees (LSTs), and show the optimal wiresizing solution satisfies a number of interesting properties, including: the LST separability, the LST monotone property, the SST local monotone property and the general dominance property. Furthermore, we study the optimal wiresizing problem using a variable grid and reveal the bundled refinement property. These properties lead to efficient algorithms to compute the lower and upper bounds of the optimal solutions. Experiment results on nets from an Intel processor layout show an interconnect delay reduction of up to 35.9% when compared to the minimum-width solution. In addition, the algorithm based on a variable grid yields a speedup of two orders of magnitude without loss of accuracy, when compared with the fixed grid based methods.*

1 Introduction

Interconnect optimization for delay minimization has drawn much attention recently. Previous work falls into two categories. One is topology optimization, such as the construction of bounded-radius bounded-cost trees[2], A-trees[6], and low-delay trees[1]. The other is wiresizing optimization, which was first introduced in [6, 7] to minimize a weighted average interconnect delay, then extended with a sensitivity-based heuristic in [11] to minimize the maximum interconnect delay. Moreover, both delay and power dissipation were optimized in [4] by simultaneous driver and interconnect sizing, and the circuit-level critical path delay (rather than the Elmore delay used in other work) was reduced in [10] by a sensitivity based, simultaneous gate and interconnect sizing.

All these methods assume that there is a *unique* source in each interconnect tree and minimize the delay between the source and a set of critical sinks. Thus, they are only applicable to *single-source interconnect trees (SSITs)*. However, there exist many interconnect trees with multiple potential sources, each driving the interconnect tree at a different time. None of the existing methods consider such *multi-source interconnect trees (MSITs)*, except a very recent work by Cong and Madden [5], where an *MSIT* topology optimization method based on the construction of min-cost min-diameter A-trees was developed.

In this paper, we study the optimal wiresizing problem for *MSIT*s under the distributed Elmore delay model, and the optimal wiresizing problem using a *variable* grid rather than a fixed grid used in previous work. The remainder of this paper is organized as follows: In Section 2, we present the formulation of the *MSIT* wiresizing problem. In Section 3 and 4, we study the properties of the optimal wiresizing solutions for *MSIT* designs, respectively under a fixed grid and a variable grid. These properties lead to efficient algorithms given in Section 5. Section 6 shows experimental results. Section 7 concludes the paper with discussions of future work. Proofs and more detailed experimental results can be found in [3]. The reader is strongly recommended to be familiar with the results in [7, 4], which are referred to several times in this paper.

2 Problem Formulation

2.1 Wiresizing for MSIT

Given an *MSIT*, each pin can be a source, or a sink, or both. Let *src(MSIT)* be the set of sources, and *sink(MSIT)* the set of sinks. We assume that no two sources are active at the same time.

A *node* refers to either a pin or a Steiner node

*This work is partially supported by ARPA/CSTO under contract J-FBI-93-112, the NSF Young Investigator Award MIP-9357582 and a grant from Intel Corporation under the NYI matching award program.

in the *MSIT*. A *segment* connects two nodes. Let $\{S_1, S_2, \cdots, S_m\}$ be the set of segments in the *MSIT*. In practice, a grid is superimposed on the routing plane. Each segment is divided into a sequence of grid edges (or simply called *edges*). Let $\{E_1, E_2, \cdots, E_n\}$ be the set of all *edges* in the *MSIT*. The wiresizing problem is to find a wire width from a set of given choices $\{W_1, W_2, \cdots, W_r\}$ ($W_1 < W_2 < \cdots < W_r$) for each edge. We assume that the wire width within an edge does not change and follow the same modeling technique used in [4], where an interconnect tree is modeled by a distributed RC circuit and a source is modeled by a fixed-value driver resistor (connected to a voltage source).

In order to handle multiple source-sink pairs, we introduce a weighted Elmore delay formulation.

$$t(MSIT, \mathcal{G}, \mathcal{W}) = \sum_{\substack{N_i \in src(MSIT) \\ N_j \in sink(MSIT)}} \lambda^{ij} \cdot t^{ij}(MSIT, \mathcal{G}, \mathcal{W}) \quad (1)$$

where $t^{ij}(MSIT, \mathcal{G}, \mathcal{W})$ is the Elmore delay [8] between the source N_i and the sink N_j. It is a function of the grid $\mathcal{G}$ and the wiresizing solution $\mathcal{W}$, with the penalty weight λ^{ij} to indicate its priority.

With these definitions, we give the general formulation of the *multi-source interconnect tree wiresizing (*MSWS*) problem* as follows:

Formulation 1 *Given an MSIT, a grid $\mathcal{G}$ and a set of possible wire width choices, the MSWS problem for delay minimization is to determine a wiresizing solution $\mathcal{W}$ which gives a width assignment w_E for every edge E under $\mathcal{G}$, such that the weighted delay $t(MSIT, \mathcal{G}, \mathcal{W})$ is minimized.*

When there is only one source in an interconnect tree, the *MSWS* problem becomes the single-source wiresizing (*SSWS*) problem studied in [7, 4]. Also, a slightly more general wiresizing problem, the multi-source wiresizing problem with a variable grid (*MSWS/G*) will be formulated in Section 4.

2.2 Weighted Delay Formulation

Given in [3], the Elmore delay t^{ij} between source N^i and sink N^j is:

$$\begin{aligned} & t^{ij}(MSIT, \mathcal{G}, \mathcal{W}) \quad (2) \\ = \; & \mathcal{K}_0^{ij} + \mathcal{K}_1 \cdot \sum_{E \in MSIT} l_E \cdot w_E \; + \\ & \mathcal{K}_2 \cdot \sum_{E,E' \in MSIT} f^{ij}(E,E') \cdot \frac{l_E \cdot l_{E'} \cdot w_{E'}}{w_E} \; + \\ & \mathcal{K}_3 \cdot \sum_{E,E' \in MSIT} f^{ij}(E,E') \cdot \frac{l_E \cdot l_{E'}}{w_E} \; + \\ & \mathcal{K}_4 \cdot \sum_{E} g^{ij}(E) \cdot \frac{l_E}{w_E} \; + \; \mathcal{K}_5 \cdot \sum_{E \in MSIT} h^{ij}(E) \cdot \frac{l_E^2}{w_E} \end{aligned}$$

where w_E and l_E are respectively the (wire) width and length of edge E, $\mathcal{K}_0^{ij}, \mathcal{K}_2, \cdots, \mathcal{K}_5$ are constants only depending on the IC or MCM fabrication technology, and $f^{ij}(E,E'), g^{ij}(E)$ and $H^{ij}(E)$ are constants only depending on the topology of the *MSIT*.

Assume that λ^{ij}'s are normalized, the objective function (1) becomes:

$$\begin{aligned} & t(MSIT, \mathcal{G}, \mathcal{W}) \quad (3) \\ = \; & \mathcal{K}_0 + \mathcal{K}_1 \cdot \sum_{E \in MSIT} l_E \cdot w_E \; + \\ & \mathcal{K}_2 \cdot \sum_{E,E' \in MSIT} F(E,E') \cdot \frac{l_E \cdot l_{E'} \cdot w_{E'}}{w_E} \; + \\ & \mathcal{K}_3 \cdot \sum_{E,E' \in MSIT} F(E,E') \cdot \frac{l_E \cdot l_{E'}}{w_E} \; + \\ & \mathcal{K}_4 \cdot \sum_{E \in MSIT} G(E) \cdot \frac{l_E}{w_E} \; + \; \mathcal{K}_5 \cdot \sum_{E \in MSIT} H(E) \cdot \frac{l_E^2}{w_E} \end{aligned}$$

where

$$\begin{aligned} \mathcal{K}_0 &= \sum_{N_i \in src(MSIT), N_j \in sink(MSIT)} \lambda^{ij} \cdot K_0^{ij} \\ F(E,E') &= \sum_{N_i \in src(MSIT), N_j \in sink(MSIT)} \lambda^{ij} \cdot f^{ij}(E,E') \\ G(E) &= \sum_{N_i \in src(MSIT), N_j \in sink(MSIT)} \lambda^{ij} \cdot g^{ij}(E) \\ H(E) &= \sum_{N_i \in src(MSIT), N_j \in sink(MSIT)} \lambda^{ij} \cdot h^{ij}(E) \end{aligned}$$

Although this weighted delay formulation for multiple sources and sinks is very similar to that for the single source and multiple sinks in [4], the coefficient functions F, G and H have very different properties, which lead to much higher complexity and very different properties for the *MSWS* problem when compared to the *SSWS* problem. These properties will be discussed in Section 3.

3 Properties of Optimal MSWS Solutions

3.1 Review of SSWS Properties

When there is only one source in the routing tree, each edge has a unique signal flow direction. We can define the “ancestors” and “descendants” with respect to the signal flow in the tree. The following properties of optimal *SSWS* solutions were given in [7].

A. Separability Given the wire width assignment of a path P originating from the source in an *SSIT*, the optimal wire width assignment for each subtree branching off from P can be carried out independently.

B. Monotone Property Given an *SSIT*, there exists an optimal wiresizing solution $\mathcal{W}$ such that $w_E \geq w_{E'}$ if edge E is an ancestor of edge E'.

Given two wiresizing solutions $\mathcal{W}$ and $\mathcal{W}'$, $\mathcal{W}$ *dominates* $\mathcal{W}'$ means that $w_E \geq w'_E$ for every edge E.

Given a wiresizing solution $\mathcal{W}$ on a routing tree, for any particular edge E in the routing tree, a *local refinement* on E is the operation to optimize the width of E, while keeping the assignment of $\mathcal{W}$ on the other edges.

C. Dominance Property Suppose that $\mathcal{W}^*$ is an optimal wiresizing solution for an *SSIT*. If a wiresizing solution $\mathcal{W}$ dominates $\mathcal{W}^*$, then any *local refinement* of $\mathcal{W}$ still dominates $\mathcal{W}^*$. Similarly, if $\mathcal{W}$ is dominated by $\mathcal{W}^*$, then any *local refinement* of $\mathcal{W}$ is still dominated by $\mathcal{W}^*$.

The presence of multiple sources greatly complicates the wiresizing problem. For example, with multiple sources, even a monotone wiresizing is not well defined. Nevertheless, our research have revealed a number of interesting properties of the optimal wiresizing solutions for an *MSIT*, some of which generalize the results on the *SSWS* problem, and others are unique for the *MSWS* problem.

3.2 Decomposition of an MSIT

In order to reduce the complexity with the *MSWS* problem, we decompose an *MSIT* into a *source subtree (SST)* and a set of *loading subtrees (LSTs)* (see Figure 1). The *SST* [1] is the subtree spanned by all source nodes in the *MSIT*. After we remove the *SST* from the *MSIT*, the remaining segments form a set of subtrees, each of them is called an *LST*. When every pin of an *MSIT* can be a source at different times, the *SST* is the entire *MSIT* and there is no *LST*.

Parallel to the ancestor-descendent relation in an *SSIT*, the left-right relation is introduced in an *MSIT*. We choose an arbitrary source as the *leftmost* node *Lsrc*. The direction of the signal (current) flowing out from *Lsrc* is the right direction along each edge E. Under such definitions, the signal in any *LST* always flows *rightward*, but the signal may flow either *leftward* or *rightward* in an edge in the *SST*.

[1] Note that *SST* defined in this paper is different from that defined in [7], where *SST* is used to denote a single stem tree.

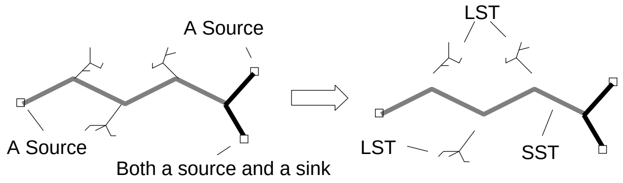

Figure 1: An *MSIT* can be decomposed into a source subtree *SST*, and a set of subtrees (three *LST*s here) branching off from the *SST*.

3.3 Properties of Optimal MSWS Solutions

A. LST Separability

Theorem 1 *Given the wire width assignment of the SST, the optimal width assignment for each LST branching off from the SST can be carried out independently. Furthermore, given the wire width assignment of both the SST and a path P originated from the root of an LST, the optimal wire width assignment for each subtree branching off from P can be carried out independently.*

B. LST Monotone Property

Theorem 2 *For an MSIT, there exists an optimal wiresizing solution $\mathcal{W}^*$ where the edge widths decrease monotonically rightward within each LST in the MSIT.*

C. SST Local Monotone Property

Given an *MSIT*, for any edges E_1 and E_2 within a segment S, it is not difficult to show that $F(E_1, E_2)$ is an invariant $F_l(S)$ if E_1 is left to E_2, and $F(E_1, E_2)$ is another invariant $F_r(S)$ if E_1 is right to E_2.

Theorem 3 *There exists an optimal wiresizing solution for an MSIT, such that the edge widths within each segment is monotone: (1) if $F_l(S) > F_r(S)$, the edge widths in S decrease monotonically rightward. (2) if $F_l(S) = F_r(S)$, the edges in S have the same width. (3) if $F_l(S) < F_r(S)$, the edge widths in S increase monotonically rightward.*

Of course, the *local monotone* property holds for segments in *LST*s, where the $F_l(S)$ is always greater than $F_r(S)$ (in fact, $F_r(S) = 0$) and the edge widths always decrease rightward, just as given by the *LST monotone* property.

D. Dominance Property

Theorem 4 *With respect to the definitions of the local refinement operation and the dominance relation in Section 3.1, the dominance property holds for the MSWS problem.*

Although the *dominance property* was proven based on the ancestor-descendant relation in [7] for the *SSWS* problem, we showed that it is a general property neither dependent on the ancestor-descendent relation, nor on the left-right relation.

Theorem 4 enables efficient computations of lower and upper bounds of the optimal wiresizing solution by the *GWSA* algorithm in [7]. It uses the *local refinement* operation iteratively to tighten the lower bound or the upper bound for one edge at a tree. A much more powerful refinement operation, *bundled refinement* operation, which may tighten the lower bound or the upper bound for a number of edges by only one operation, will be introduced in the next section.

4 Properties of Optimal MSWS/G Solutions

Up to now, both the *MSWS* problem defined in this paper and the *SSWS* problem defined in all previous work [7, 4, 11, 10] were only investigated and solved using a *fixed* grid. The grid controls how often the wire widths are allowed to change. However, it is difficult to choose a proper grid structure. For the best accuracy, a very fine, uniform grid is usually chosen, which results in very high memory usage and computation time due to the large number of edges. We now investigate methods to obtain the optimal wiresizing results using a *non-uniform* and *coarser* grid.

A novel contribution of our work is to introduce a *variable*-grid formulation for the *MSWS* problem. The grid maybe finer in some regions but coarser in others. Moreover, we begin with a coarser grid then proceed to a finer one. Theorem 5 to be presented in Section 4.2 justifies this strategy and leads to much more efficient algorithms with the same accuracy when compared with previous work.

All properties in this section hold for both the *MSWS* problem and the *SSWS* problem, but we shall concentrate on the *MSWS* problem because the *SSWS* problem can be treated as a special case.

4.1 Grid Refinement and Bundled Edges

Given an *MSIT*, let $\mathcal{G}_0$ be the grid with each segment in the *MSIT* as an edge, $\mathcal{G}_F$ the uniform grid with the finest grid unit δ everywhere (δ is determined by the design rules). If each edge in grid $\mathcal{G}$ corresponds to one or several edges in grid $\mathcal{G}'$, $\mathcal{G}'$ is a refinement of $\mathcal{G}$. A grid $\mathcal{G}$ is *valid* only if $\mathcal{G}$ is a refinement of $\mathcal{G}_0$ and the length of every edge is a multiple of δ. Then, among all valid grid, $\mathcal{G}_0$ is coarsest and $\mathcal{G}_F$ is finest.

With these definitions, the *variable-grid multi-source wiresizing (MSWS/G) problem*, can be formulated as follows:

Formulation 2 *Given an MSIT, the finest unit δ, and a set of possible wire width choices, the MSWS/G problem for delay minimization is to determine both a grid $\mathcal{G}$ and a wiresizing solution $\mathcal{W}$, such that the weighted delay $t(MSIT, \mathcal{G}, \mathcal{W})$ is minimized.*

The *dominance* relation can be extended to consider the variable-grid cases.

Definition 1 (**Dominance**) *Given two wiresizing solutions $\mathcal{W}$ and $\mathcal{W}'$, $\mathcal{W}'$* dominates *$\mathcal{W}$ if $w'_E \geq w_E$ for every edge E under $\mathcal{G}_{\mathcal{F}}$.*

Definition 2 (Bundled Edge) *Given an MSIT, a segment S and the finest grid $\mathcal{G}_F$, let $E_1, \cdots, E_p$ be a maximal sequence of successive edges under $\mathcal{G}_F$, within S and with the same wire width in the optimal wiresizing solution under $\mathcal{G}_F$, we say that they form a bundled edge.*

Corollary of Theorem 3 Each segment in an *MSIT* has at most r bundled edges where r is the number of possible wire width choices.

We shall compute the optimal width for each bundled edge directly, instead of treating it as a sequence of edges of length δ under the grid $\mathcal{G}_F$.

4.2 Bundled Refinement Operations

Let $\mathcal{W}$ be a wiresizing solution which dominates the optimal solution $\mathcal{W}^*$, and E be an edge under the current grid $\mathcal{G}$ and in segment S. Without loss of generality, we assume $F_l(S) \geq F_r(S)$ and treat E as two edges E_l and $\overline{E_l}$. E_l is the left end of E and with length δ, $\overline{E_l}$ is the remaining part of E (recall δ is the grid unit in the finest grid $\mathcal{G}_F$). Let $\tilde{w_{E_l}}$ be the optimized width for E_l based on the objective function (3) while keeping the assignment of $\mathcal{W}$ on $\overline{E_l}$ and any edge E' other than E. Then, $\tilde{w_{E_l}}$ is regarded as a refined upper bound of the *entire* edge E (not only E_l). This operation is called a *bundled refinement operation for the upper bound (BRU)*.

The rational for the *BRU* operation is as follows: if $F_l(S) \geq F_r(S)$, E_l is always wider than all edges under $\mathcal{G}_F$ in $\overline{E_l}$ (the *local monotone* property), and a refinement of an upper bound of E_l also gives an (possibly refined) upper bound for any edge under $\mathcal{G}_F$ in $\overline{E_l}$. Nevertheless, E will not be divided into E_l and $\overline{E_l}$ when performing the BRU operation on edges other than E.

Similarly, the *bundled refinement operation for the lower bound (BRL)* can be defined for a wiresizing solution $\mathcal{W}$ dominated by $\mathcal{W}^*$. Again, assuming $F_l(S) \geq F_r(S)$, we treat E as two edges E_r and $\overline{E_r}$. E_r is the right end of E and with length δ, $\overline{E_r}$ is the remaining part of E. Let $\tilde{w_{E_r}}$ be the optimized width for E_r based on the objective function (3) while keeping the assignment of $\mathcal{W}$ on $\overline{E_r}$ and any edge E' other than E. Then, $\tilde{w_{E_r}}$ is regarded as a refined lower bound of the *entire* edge E.

We have proven the *bundled refinement* property:

Theorem 5 *Let $\mathcal{W}^*$ be an optimal wiresizing solution under $\mathcal{G}_F$. If a wiresizing solution $\mathcal{W}$ dominates $\mathcal{W}^*$, then the wiresizing solution obtained by any BRU operation under any grid $\mathcal{G}$ on $\mathcal{W}$ still dominates $\mathcal{W}^*$. Similarly, if $\mathcal{W}$ is dominated by $\mathcal{W}^*$, then the wiresizing solution obtained by any BRL operation under any grid $\mathcal{G}$ on $\mathcal{W}$ is still dominated by $\mathcal{W}^*$.*

5 Optimal Wiresizing Algorithm

Given an *MSIT*, we first compute a lower bound and an upper bound of the optimal wiresizing solution. If the lower bound and the upper bound meet, which is very likely in practice, we get the optimal wiresizing solution immediately. Otherwise, a bounded enumeration technique combined with a dynamic programming technique is carried out between the lower and upper bounds where they do not meet.

5.1 OWBR Algorithm

Compared with the *GWSA* algorithm [7, 4] working on a *fixed* grid to refine the lower/upper bounds of the optimal wiresizing solution, our *Optimal Wiresizing algorithm with Bundled Refinement (OWBR)* refines both the wiresizing grid and the lower/upper bounds of the optimal wiresizing solution for an *MSIT*.

Starting with the coarsest grid $\mathcal{G}_0$, we perform BRU and BRL iteratively through an *MSIT* as follows. We first assign the minimum width to all edges (in this case, an edge is a segment in the *MSIT*), then traverse the *MSIT* and perform BRL operation on every edge. This process is repeated until no improvement is achieved on any edge in the last round of traversal. The result is a lower bound of the optimal wiresizing solution. Similarly, we assign the maximum width to all edges and perform BRU operations, obtain an upper bound of the optimal wiresizing solution. This is the first *pass* of *OWBR*.

After each *pass*, we check the lower and upper bounds. If there is a gap between the lower bound and the upper bound for an edge E (called an *unconvergent* edge) and its length is still larger than δ, we divide E into two edges of almost equal lengths (they may differ by δ in order to maintain a *valid* grid), each of which inherits the old lower and upper bounds. After the refinement of all unconvergent edges, another *pass* of lower/upper bound refinements is carried out on all unconvergent edges in the refined grid.

This process is repeated until we either have the identical lower and upper bound for all edges under current grid, or each unconvergent edge is of length δ.

Definition 3 *If a wiresizing solution $\mathcal{W}$ dominates the optimal solution $\mathcal{W}^*$ and can not be further refined by any local refinement operation under the finest grid $\mathcal{G}_F$, $\mathcal{W}$ is an $\mathcal{G}_F$-tight upper bound. Similarly, $\mathcal{W}$ is an $\mathcal{G}_F$-tight lower bound if $\mathcal{W}$ is dominated by $\mathcal{W}^*$ and can not be further refined by any local refinement operation under $\mathcal{G}_F$*[2].

Theorem 6 *The lower and upper bounds provided by OWBR are $\mathcal{G}_F$-tight.*

Theorem 7 *The worst-case complexity of GWSA is $O(r \cdot n^3)$. The average-case complexity of OWBR is $O(r \cdot m^3 \cdot \ln^4 n_0)$.*

Where, r is the number of wire-width choices, m is the number of segments in the *MSIT*, n is the total wire length of the *MSIT* and n_0 is the wire length of the longest segment in the *MSIT* (both in terms of δ). Note that $m \leq 2(k-1)$, where k is the number of pins. Since k is a constant bounded by the size of the largest multi-source net (normally no larger than 32), m is also a constant, plus $\ln n_0 \ll n$, *OWBR* is much more efficient than *GWSA*. This has been further confirmed by experimental results.

5.2 Bounded Enumeration

Because the *separability* and the *monotone property* hold in *LST*s, the optimal single-source wiresizing algorithm, named *OWSA* [7], can be used for *LST*s. The *OWSA* algorithm is a dynamic programming method

[2]There may exist more than one $\mathcal{G}_F$-tight upper (or lower) bounds of an $\mathcal{W}^*$

to compute an optimal wiresizing solution between the lower and upper bounds.

Without the *separability* in *SST*, the wire width assignment for unconvergent edges in the *SST* need to be enumerated subject to the *local monotone* property.

Our experiments show that *OWBR* gives the convergent bounds on all edges in an *MSIT* for most cases. For those cases which have unconvergent edges, the percentage of unconvergent edges is very small. Moreover, the gap between the lower and upper bounds on each unconvergent edge is also very small (usually being one in our experiments). Therefore, the enumeration procedure on unconvergent edges in the *SST* is very fast in practice.

6 Experimental Results

We have implemented the *OWBR* algorithm and tested it on a large number of *MSIT*s for both the MCM and the IC technologies. We shall present both the comparison of different wiresizing solutions and the comparison between the *OWBR* algorithm and the *GWSA* algorithm. The delays reported in this section are computed using HSPICE. The use of HSPICE simulation results, instead of calculated Elmore delay values, verifies not only the quality of our *MSWS* solutions but also the validity of our interconnect modeling and the correctness of our *MSWS* problem formulation.

Technology:	CMOS	MCM
Driver Resistance(Ω)	156	25
Wire Resistance($\Omega/\Box$)	0.044	0.02
Loading Capacitance(fF)	3.720	1000
Wire Capacitance($aF/\mu m^2$)	41.3	3.46
Fringing Capacitance($aF/\mu m$)	150	50.4
Finest Grid Unit(μm)	10	100

Table 1: Parameters for CMOS and MCM designs.

The parameters used in the optimal wiresizing algorithms and HSPICE are summarized in Table 1. The wire width choices are $\{W_1, 2W_1, 3W_1, 4W_1, 5W_1\}$, where W_1 is the minimum wire width ($0.95\mu m$ in the IC technology and $10\mu m$ in the MCM technology). Note that our algorithms are still valid if the wire widths are not integral multiples of the minimum width.

6.1 Comparison of Different Wiresizing Solutions

A. Results on Simple Artificial Nets Let *min_width* be the wiresizing solution with wire width W_1 everywhere, *opt_ssws* be the wiresizing solution given by the optimal *SSWS* algorithms[7, 4], and *opt_msws* be the wiresizing solution given by our optimal *MSWS* algorithms. Results on an 4-pin H-topology interconnect tree where each pin can be both a source and a sink are shown in Table 2. Compared to the *min_width* solutions, the *opt_ssws* solutions may have larger maximum delay (along a net) and consume larger routing area, while our *opt_msws* solutions reduce the maximum delay and the (weighted) average delay by up to 33% and 27%, respectively. The single-source wiresizing method is clearly not applicable to the *MSWS* problem.

B. Results on Industrial Nets We also tested our algorithms on several multi-source nets provided by Intel. These nets were extracted from the top-level floorplan of a high-performance microprocessor. Most pins of these nets can serve as both inputs and outputs, and all pairs between sources and sinks (excluding feedthrough pins) are considered to be timing critical. We use the 1-Steiner tree algorithm [9] to route these nets. As shown in Table 3, the *opt_msws* solutions consistently outperform the *min_width* solutions with as much as 36% and 17% reduction on the maximum delay and the average delay.

It is interesting to observe that although the (weighted) average delay is the objective of our algorithms, all experimental results show that this formulation reduces the maximal delay as well. Also, the delay reduction for nets with larger span is more significant.

6.2 Speed-up Using Variable Grid Computation

Ten 8-pin nets with pins randomly distributing in a 1000×1000 grid and routed by the 1-Steiner algorithm are wiresized. The CPU times used by the *OWBR* algorithm and the *GWSA* algorithm to obtain the lower bound and the upper bound of the optimal solution are reported in Table 4. We observed speed-ups ranging from a factor of 2 orders of magnitude to 3 orders of magnitude.

7 Conclusions and Future Work

The results in this paper have shown convincingly that proper sizing of the wire segments in multi-source nets can lead to significant reduction in the interconnect delay. We have also developed an efficient wiresizing algorithm using a *(coarse) variable* grid , which

Net Length	300 μm			3000 μm		
IC Wiresizing	min_width	opt_ssws	opt_msws	min_width	opt_ssws	opt_msws
Maximum Delay (ns)	0.345	0.347(+0.6%)	0.287(-16.8%)	2.58	2.47(-4.3%)	1.73(-32.7%)
Average Delay (ns)	0.303	0.290(-4.3%)	0.268(-11.5%)	2.08	1.89(-9.1%)	1.55(-25.5%)
Normalized Area	1	2.13	2.33	1	2.42	3.44

Net Length	3000 μm			30000 μm		
MCM Wiresizing	min_width	opt_ssws	opt_msws	min_width	opt_ssws	opt_msws
Maximum Delay (ns)	0.417	0.425(+1.9%)	0.355(-14.86%)	4.76	4.94(+4.3%)	3.23(-32.1%)
Average Delay (ns)	0.361	0.359(-0.6%)	0.321(-11.1%)	3.70	3.46(-6.5%)	2.71(-26.7%)
Normalized Area	1	2.39	2.00	1	2.49	3.44

Table 2: Comparison of *min_width*, *opt_ssws* and *opt_msws* solutions on a 4-source tree with H-topology

	net length	Normalized Area	Maximum Delay (ns)		Average Delay (ns)	
	(μm)	opt_msws	min_width	opt_msws	min_width	opt_msws
net1	3600	1.000	0.1435	0.1435 (0.0%)	0.1261	0.1261(0.0%)
net2	6600	1.000	0.2572	0.2572 (0.0%)	0.2004	0.2004(0.0%)
net3	10070	2.006	0.5230	0.4444 (-15.0%)	0.3504	0.3389(-3.28%)
net4	10570	2.020	0.5853	0.4992 (-14.7%)	0.5007	0.3906(-22.1%)
net5	31980	3.039	3.4505	2.2449 (-34.9%)	1.8968	1.3991(-26.2%)

Table 3: Multi-source wiresizing results on several nets in an Intel microprocessor layout

GWSA Run Time (s)	113.18	39.15	42.52	87.82	66.60	87.12	66.10	49.75	160.98	31.88
OWBR Run Time (s)	0.98	0.01	0.03	0.3	0.05	0.17	0.03	0.05	1.83	0.2
Speedup factor	$\geq$ 100	$\geq$ 1000	$\geq$ 1000	$\geq$ 100	$\geq$ 1000	$\geq$ 100	$\geq$ 1000	$\geq$ 100	$\geq$ 100	$\geq$ 100

Table 4: Performance comparison between *OWBR* and *GWSA*

achieves the same accuracy in the wiresizing solutions as the *fixed* grid based optimal wiresizing algorithms on the finest grid, but uses much less memory and computation time. To the best of our knowledge, it is the first work which presents an in-depth study of both the optimal wiresizing problem for multi-source interconnect trees and the optimal wiresizing problem using a *variable* grid.

In order to further reduce the interconnect delay in multi-source nets, we plan to study the simultaneous driver and wiresizing problem for multi-source nets. Also, we would like to develop efficient multi-source wiresizing algorithms for multiple-objective optimization to minimize delay, area, and power dissipation and explore the tradeoff among these objectives.

Acknowledgments The authors would like to thank Heming Chen at Intel Design Technology Department for providing the multiple source routing examples, and Cheng-Kok Koh and Patrick Madden at UCLA for their helpful discussions.

References

[1] K. D. Boese, A. B. Kahng, and G. Robins, "High-Performance Routing Trees With Identified Critical Sinks", *Proc. ACM/IEEE DAC*, 1993, pp. 182-187.

[2] 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.

[3] J. Cong and L. He, "Optimal Wiresizing for Interconnects with Multiple Sources", *UCLA Computer Science Dept. Tech. Report CSD-00031*, 1995 (available at http://ballade.cs.ucla.edu/:8080/ ~cong/publications.htm).

[4] J. Cong and C.-K. Koh, "Simultaneous Driver and Wire Sizing for Performance and Power Optimization", *IEEE Trans. on VLSI*, 2(4), December 1994, pp. 408-423.

[5] J. Cong and P. H. Madden, "Performance-Driven Routing with Multiple Sources", *Proc. IEEE ISCAS*, 1995, pp. 203-206.

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

[7] J. Cong and K. S. Leung, "Optimal Wiresizing Under the Distributed Elmore Delay Model", *IEEE Trans. on CAD*, 14(3), March 1995, pp. 321-336.

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

[9] A. B. Kahng and G. Robins, "A New Class of Iterative Steiner Tree Heuristics with Good Performance", *Proc. IEEE ICCAD*, July 1992, pp. 893-902.

[10] N. Menezes, S. Pullela, and L. T. Pilegi, "Simultaneous Gate and Interconnect Sizing for Circuit-Level delay Optimization", *Proc. ACM/IEEE DAC*, 1995, pp. 690-695.

[11] S. S. Sapatnekar, "RC Interconnect Optimization Under the Elmore Delay Model", *Proc. ACM/IEEE DAC*, 1994, pp. 387-391.