# 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

In this paper, we study the optimal wiresizing problem for nets with multiple sources under the RC tree model and the Elmore delay model. We decompose the routing tree for a multi-source net into the source subtree (*SST*) and a set of loading subtrees (*LST*s), and show that 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 segment-division rather than an *a priori* fixed segment-division in all previous works and reveal the bundled refinement property. These properties lead to efficient algorithms to compute the optimal solutions. We have tested our algorithm on a large number of multi-source nets, including several nets extracted from the layout for a high performance Intel microprocessor. Accurate SPICE simulation shows that our methods reduce the interconnect delay by up to 36.3% for submicron CMOS technology and 32.1% for MCM technology, respectively, when compared to the minimal wire width solution. In addition, the algorithm based on the variable segment-division yields a speedup of over 100x time and does not lose any accuracy, when compared with the methods based on the *a priori* fixed segment-division.

*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.

# 1 Introduction

As the VLSI technology reaches submicron device dimensions, gigahertz clock frequencies, and highly integrated multi-chip module design, interconnect delay has become the dominating factor to determine system performance. Therefore, interconnect optimization for delay minimization has drawn much attention recently. Techniques for interconnect delay minimization falls into two categories. One is topology optimization, such as the constructions of bounded-radius bounded-cost trees[6], AHHK trees[1], A-trees[9], low-delay trees[2], and IDW/CFD trees[18]. In essence, these methods construct an interconnect tree to minimize both the total tree length and the paths between the input pin (also called the source) and a set of timing-critical output pins (also called critical sinks), while the conventional Steiner tree algorithms only minimize the total tree length. Besides, the non-tree routing was also explored in [21, 30].

The other type of interconnect optimization methods is wiresizing of interconnects, which was first introduced by Cong *et al* in [9, 11]. They modeled the routing tree as a distributed RC tree and formulated the wiresizing problem under the Elmore delay model [14, 29] to minimize a weighted average interconnect delay from the source to a set of critical sinks. The first polynomial-time optimal algorithm was developed based on the separability, the monotone property and the dominance property. Later on, since the Elmore delay along a RC tree is a posynomial function of wire widths as first pointed out in [15], the sensitivity-based heuristic like that used in TILOS [15] and the convex programming technique [13, 24] were applied in [23] and [22], respectively, to minimize the maximum interconnect delay. Besides, heuristics for performing greedy wiresizing during topology construction [17] and wiresizing for non-tree structures [30] have been proposed. Wiresizing heuristics to minimize clock skew have been used in [28, 31, 4].

Wiresizing can be conducted simultaneously with both the buffer inserting and the driver (buffer) sizing to further reduce the interconnect delay. The authors of [7] formulated the simultaneous driver and wire sizing (*SDWS*) problem under the Elmore delay model to minimize both delay and power dissipation. They revealed a dominance relationship between the driver sizing and the correspondent optimal wire sizing, and developed an efficient SDWS algorithm. Recently, the simultaneous gate and wire sizing was studied in [26, 27] under the higher-order delay model. Both compute the delay sensitivity with respect to wire width and gate size and are gradient-guided methods in essence.

Very recently, a dynamic programming wiresizing scheme for interconnect trees was proposed in [20]. Based on a generalization of the optimal buffer placement algorithm in [16], it handles simultaneous wiresizing and buffer insertion, for both performance and power optimization. It also takes the signal slew into account in the buffer delay computation.

Clearly, all these interconnect optimization 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. For example, in the interconnect tree designed for a bus-type net with multiple sources as shown in Figure 1.a (reproduced from [8]), each pin is controlled by tri-state gates and may become the source of the interconnect tree. None of the existing interconnect optimization methods consider such *multi-source interconnect trees (MSITs)*, except a very recent work by Cong and Madden [8], where an $MSIT$ topology optimization method based on the construction of min-cost min-diameter A-trees was developed. Their method can reduce the maximum delay along an $MSIT$ by an average of 6.1% for submicron CMOS and 19.9% for MCM, respectively, when compared with conventional routing methods.

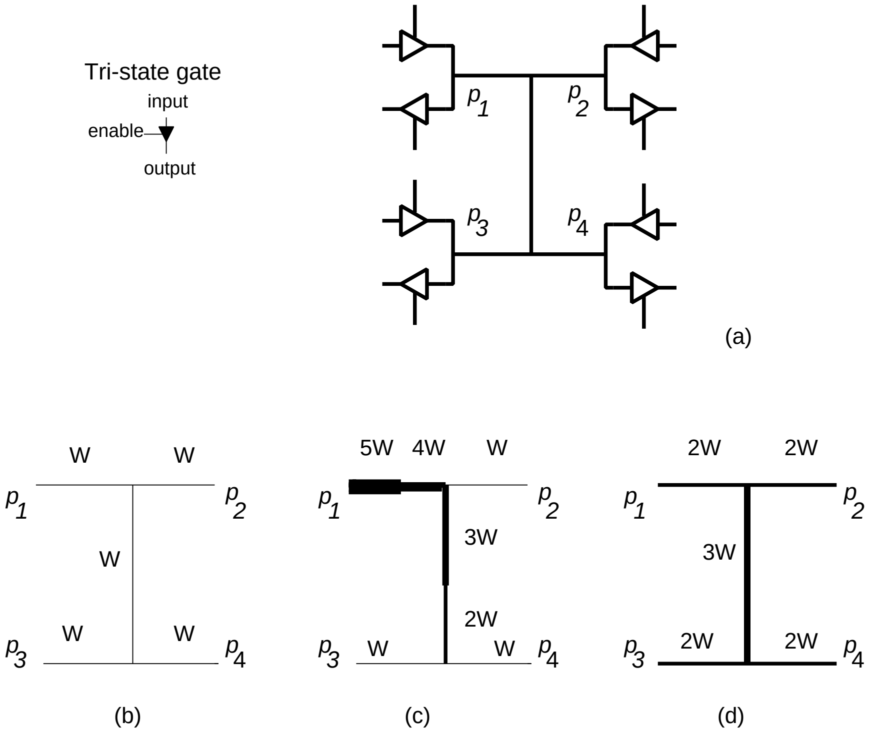
It is not difficult to see that the single-source wiresizing ($SSWS$) solution may perform poorly for wiresizing in $MSIT$ designs. For the optimal routing tree of the multi-source net in Figure 1.a, based on the parameters in Table 2 for a submicron CMOS technology, compared with the minimum-width wiresizing solution *min_width* (Figure 1.b), the optimal single-source wiresizing solution *opt_ssws* (Figure 1.c), by assuming that an arbitrary pin is the source and the others the critical sinks, in fact increases the maximum delay by 0.6%, while the multi-source optimal wiresizing solution *opt_msws* (Figure 1.d) reduces the maximum delay by 16.8%. Thus, multi-source wiresizing optimization method is necessary for $MSIT$ designs.

In this paper, we will study the optimal wiresizing problem for $MSIT$s under the distributed RC model and the Elmore delay model. We decompose an $MSIT$ into a source subtree($SST$) and a set of loading subtrees($LST$s), and show a number of interesting properties of the optimal wiresizing solutions under this decomposition, including: the LST separability, the LST monotone property, the SST local monotone property and the general dominance property. These properties lead to effective algorithms to compute the optimal wire width assignment for an $MSIT$. We have tested our algorithm on a large number of $MSIT$s in both CMOS and MCM technologies, including those for several multi-source nets extracted from the layout of a high performance Intel processor. Accurate SPICE simulation shows that our methods reduce the interconnect delay by up to 36.3% for submicron CMOS technology and 32.1% for MCM technology, respectively.

Furthermore, we study the optimal wiresizing problem using a *variable* segment-division rather than an *a priori* fixed segment-division used in all previous works. We show the *bundle refinement* property which leads to a very efficient wire sizing algorithm based on *bundled refinement* and recursively *refined* segment-division. The algorithm yields a speedup of over 100x time and does not lose any accuracy, when compared to the method based an *a priori* fixed segment-division. 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* segment-division.

We also investigate the fidelity of the Elmore delay model for wiresizing in this paper and compare our optimal solution under the Elmore delay model versus that selected by SPICE. We show that the Elmore delay model has high fidelity for wiresizing optimization, which convincingly justifies our formulation under the Elmore delay model.

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 $MSIT$




| | min_width | opt_ssws | opt_msws |
|---|---|---|---|
| Maximum Delay (ns) | 0.345 | 0.347 (+0.6%) | 0.287 (-16.8%) |
| Average Delay (ns) | 0.303 | 0.290 (-4.3%) | 0.268 (-11.5%) |
| Normalized Area | 1 | 2.13 | 2.33 |

Figure 1: A multi-source interconnect tree with three wiresizing solutions
(a) A signal net with four pins ($P_1, P_2, P_3$ and $P_4$). Each of them is controlled by tri-state gates and may become a source for the interconnect tree at a different time. (b) The minimum width wiresizing solution (min_width) with the minimum wire width $w$ being $0.9\mu m$. (c) The optimal single-source wiresizing solution (opt_ssws) under the assumption that pin $P_1$ is the unique source, and pins $\{P_2, P_3, P_4\}$ are the critical sinks. (d) The optimal multi-source wiresizing solution (opt_msws). The total wire length is 300 $\mu m$. The parasitic parameters are those in Table 2 for a submicron CMOS technology. More details can be found later in Section 6.1.A.

designs, under the *a priori* fixed and the variable segment-divisions, respectively. These properties lead to efficient algorithms given in Section 5. Section 6 shows experimental results, including the fidelity study of the Elmore delay model. Section 7 concludes the paper with discussions of future work. An extended abstract of this paper was presented in ICCAD'95 [5].

# 2 Problem Formulation

## 2.1 Wiresizing for MSIT

Given an MSIT, each pin can be a source (driver), or a sink (receiver), or both at different times. Let *src(MSIT)* be the set of pins which can be sources of the MSIT, and *sink(MSIT)* the set of pins which can be sinks of the MSIT. A pin belongs to both src(MSIT) and sink(MSIT) if it can be a source and a sink at different times.

We assume that no two sources in an MSIT are active at the same time.

A *node* refers to either a pin or a Steiner point in the MSIT. A *segment* connects two *node*s. Let $\{S_1, S_2, \cdots, S_m\}$ be the set of segments in the MSIT. In order to capture the distributed resistive property of the interconnect and achieve more optimized wiresizing solutions, a segment is divided into a sequence of *uni-segments*. The term of "uni-segment" is coined based on this assumption that the wire width is *uniform* within a uni-segment. The *segment-division* determines the set of all uni-segments, $\{E_1, E_2, \cdots, E_n\}$, in the MSIT. Our wiresizing problem is formulated to find a wire width for each uni-segment from a set of given choices $\{W_1, W_2, \cdots, W_r\}$ ($W_1 < W_2 < \cdots < W_r$). Note that the segment-division is implicitly applied by introducing artificial degree-2 Steiner points in [11] and treating each segment as a uni-segment, and explicitly applied by dividing each segment into a sequence of wires of unit length in [7] and treating each wire of unit length as a uni-segment. The segment-divisions are given *a priori* and fixed during the wiresizing procedure in [11, 7]. In our formulation, the segment-division is in fact defined by and variable during the wiresizing procedure, which will be discussed later in Section 4. For simplicity, just assume that we are given an *a priori* fixed segment-division in this section.

The modeling technique similar as those used in [11, 7] is applied. Each uni-segment is treated as a $\pi$-type RC circuit containing resistance $r_E$ and capacitance $c_E$, respectively. Let the unit-width unit-length wire have wire resistance $r_0$, wire area capacitance $c_0$ and wire fringing capacitance $c_1$, then $r_E = r_0 \cdot \frac{l_E}{w_E}$ and $c_E = c_0 \cdot w_E \cdot l_E + c_1 \cdot l_E$ for uni-segment $E$ with width $w_E$ and length $l_E$. The receiver at a sink is modeled by a loading capacitance, and a source together with its driver is modeled by a fixed-value driver resistor connected to a voltage source. So a given interconnect including its drivers and receivers is modeled by a distributed RC tree (see Figure 2 for the modeling for the routing tree of a two-source two-sink net).

Given the RC tree model, the Elmore delay [14] $t^{ij}$ from source $N_i$ to sink $N_j$ is a function of the segment-division $\mathcal{E}$ and the wiresizing solution $\mathcal{W}$, which can be written as the following (1) according to the Elmore delay formulation for RC trees given in [29].

$$t^{ij}(MSIT, \mathcal{E}, \mathcal{W}) \quad = \sum_{E \in P(N_i, N_j)} r_E \cdot (\frac{c_E}{2} + C_E) \tag{1}$$

where the summation is taken over all uni-segments on the unique path $P(N_i, N_j)$ from source $N_i$ to sink $N_j$, and $C_E$ is the total downstream capacitance of uni-segment $E$ with respect to source $N_i$.

In order to handle multiple source-sink pairs, we introduce the following weighted delay formulation (2).

$$t(MSIT, \mathcal{E}, \mathcal{W}) = \sum_{N_i \in src(MSIT), N_j \in sink(MSIT)} \lambda^{ij} \cdot t^{ij}(MSIT, \mathcal{E}, \mathcal{W}) \tag{2}$$

where $\lambda^{ij}$ is the penalty weight to indicate the priority of the Elmore delay $t^{ij}$ between source $N_i$ and sink $N_j$.

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

**Formulation 1** *Given an MSIT, a segment-division $\mathcal{E}$ and a set of possible wire width choices, the MSWS*

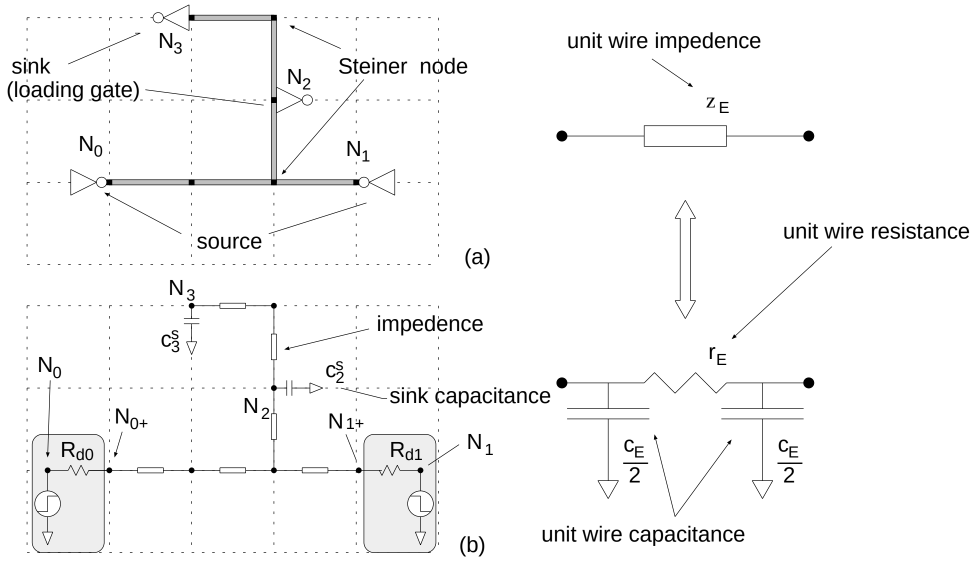



Figure 2: A two-source and two-sink interconnect tree with its distributed RC model under a uniform segment-division. (a) The layout of an MSIT with two sources $N_0$ and $N_1$, and two sinks $N_2$ and $N_3$. (b) The distributed RC model for this MSIT. Each uni-segment $E$ under the uniform segment-division is modeled as a $\pi$-type RC circuit containing a resistor of $r_E$ and two capacitors of $\frac{c_E}{2}$ each ($c_E$ includes both wire area capacitance and fringing capacitance). Each sink $v$ has an extra loading capacitance $c_v^s$ due to the receiver. Each source together with its driver is modeled by a driver resistance connected to an ideal voltage source.

*problem for delay minimization is to determine a wiresizing solution $\mathcal{W}$ which gives a wire width $w_E$ for every uni-segment $E$ under $\mathcal{E}$, such that the weighted delay $t(MSIT, \mathcal{E}, \mathcal{W})$ is minimized.*

When there is only one source in an interconnect tree, the MSWS problem degenerates into the *single source wiresizing (SSWS)* problem. Note that we assume a given segment-division in Formulation 1. A more general wiresizing problem, the multi-source wiresizing problem *without* an *a priori* given segment-division (*the MSWS/E problem)* will be presented and solved in Section 4.

## 2.2 Weighted Delay Formulation

Similar to [11, 7], the Elmore delay $t^{ij}$ between source $N^i$ and sink $N^j$ can be formulated as follows:

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

where $w_E$ and $l_E$ are respectively the (wire) width and length of the uni-segment $E$, respectively. When $\mathcal{E}$ is uniform and every uni-segment has the same length, the length can be omitted from this equation, just as given in [7].[1] Besides, $\mathcal{K}_0^{ij}, \mathcal{K}_2, \cdots, \mathcal{K}_5$ are constants depending only on the IC or MCM fabrication technology

[1]For the simplicity of presentation, we assume that the segment-division is uniform in all proofs of this paper, so we use the formulation without $l_E$'s later in our proofs. This simplification will be removed in Section 4.1.

(details concerning them can be found in [11, 7]), and $f^{ij}(E,E'), g^{ij}(E)$ and $H^{ij}(E)$ are defined below.

$$f^{ij}(E,E') = \begin{cases} 1 & \text{if } E \in P(N_i,N_j) \text{ and } E' \in Des^i(E) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$g^{ij}(E) = \begin{cases} \sum_{v \in sink^i(E)} c_v^s & \text{if } E \in P(N_i,N_j) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$h^{ij}(E) = \begin{cases} 1 & \text{if } E \in P(N_i,N_j) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $P(N_i,N_j)$ is the unique path from source $N_i$ to sink $N_j$, $Des^i(E)$ is the uni-segment set of all downstream uni-segments of $E$ with respect to the signal flow from source $N_i$, $sink^i(E)$ is the node set of all downstream sinks of $E$, again with respect to the signal flow from source $N_i$ and $c_v^s$ is the loading capacitance at sink $v$.

Assume that $\lambda^{ij}$'s are normalized, i.e.,

$$\sum_{N_i \in src(MSIT), N_j \in sink(MSIT)} \lambda^{ij} = 1$$

the objective function (2) becomes:

$$\begin{aligned} t(MSIT, \mathcal{E}, \mathcal{W}) &= \mathcal{K}_0 + \mathcal{K}_1 \cdot \sum_{E \in MSIT} l_E \cdot w_E + \mathcal{K}_2 \cdot \sum_{E,E' \in MSIT} F(E,E') \cdot \frac{l_E \cdot l_{E'} \cdot w_{E'}}{w_E} + \\ & \mathcal{K}_3 \cdot \sum_{E,E' \in MSIT} F(E,E') \cdot \frac{l_E \cdot l_{E'}}{w_E} + \mathcal{K}_4 \cdot \sum_{E \in MSIT} G(E) \cdot \frac{l_E}{w_E} + \mathcal{K}_5 \cdot \sum_{E \in MSIT} H(E) \cdot \frac{l_E^2}{w_E} \end{aligned} \tag{7}$$

where

$$\mathcal{K}_0 = \sum_{N_i \in src(MSIT), N_j \in sink(MSIT)} \lambda^{ij} \cdot K_0^{ij}$$

$$F(E,E') = \sum_{N_i \in src(MSIT), N_j \in sink(MSIT)} \lambda^{ij} \cdot f^{ij}(E,E') \tag{8}$$

$$G(E) = \sum_{N_i \in src(MSIT), N_j \in sink(MSIT)} \lambda^{ij} \cdot g^{ij}(E) \tag{9}$$

$$H(E) = \sum_{N_i \in src(MSIT), N_j \in sink(MSIT)} \lambda^{ij} \cdot h^{ij}(E) \tag{10}$$

Our MSWS problem aims to find the optimal $w_E$'s for the weighted delay formulation (7). Although this weighted delay formulation for multiple sources and sinks is very similar to that for the single source and multiple sinks in [7], 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

The single-source wiresizing problem (SSWS) under the an *a priori* fixed segment-division was studied in [11], and the polynomial-time optimal algorithm was developed based on the separability, the monotone property and the dominance property. Similar properties will be studied in the more general context of multi-source wiresizing problem (MSWS) in this section. Furthermore, new properties will be revealed in this section as well as the next section. These properties will enable us to apply the algorithms developed in [11] to the MSWS problem to certain extent and to develop even more efficient algorithms in Section 5.

### 3.1 Review of Optimal SSWS Solutions

When there is only one source in the routing tree, each uni-segment has a unique signal flow direction. We can define the "ancestors" and "descendants" with respect to the signal flow in the tree. Let $Ans(E)$ be the set of uni-segments which are ancestors of $E$ and $Des(E)$ be the set of uni-segments which are descendants of $E$. The following properties of optimal *SSWS* solutions were given in [11].

**A. Separability** Given the wire width assignment of a path $P$ originated 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 uni-segment $E$ is an ancestor of uni-segment $E'$.

Given two wiresizing solutions $\mathcal{W}$ and $\mathcal{W}'$, we define $\mathcal{W}$ *dominates* $\mathcal{W}'$ if $w_E \geq w'_E$ for every uni-segment $E$.

Given a wiresizing solution $\mathcal{W}$ for the routing tree, and any particular uni-segment $E$ in the tree, a *local refinement* on $E$ is defined to be the operation to optimize the width of $E$ while keeping wire width assignment of $\mathcal{W}$ on other uni-segments.

**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 MSITs, which will be presented in the next a few subsections. Some of these properties 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 the *source subtree (SST)* and a set of *loading subtrees (LSTs)* (see Figure 3). The SST [2] 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 entire MSIT becomes the SST 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

[2] Note that SST defined in this paper is different from that defined in [11], where SST is used to denote a single stem tree.

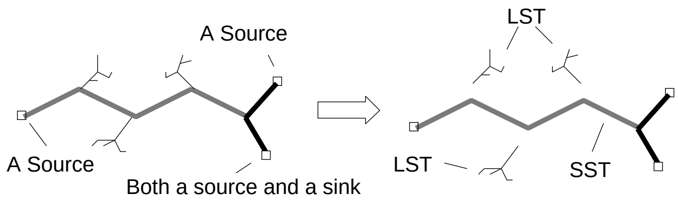



Figure 3: An $MSIT$ can be decomposed into the source subtree $SST$, and a set of loading subtrees (three $LST$s here) branching off from the $SST$. The dark uni-segments belong to the $SST$.

$Lsrc$ is the right direction along each uni-segment $E$. Under such definitions, the signal in any LST always flows *rightward*, but the signal may flow either *leftward* or *rightward* in an uni-segment in the SST.

Furthermore, careful study of the definitions of $f^{ij}$, $g^{ij}$ and $h^{ij}$ in (4)–(6) and those of $F, G$ and $H$ in (8)–(10) reveals the following properties for the coefficient functions F, G and H.

**Lemma 1** *Given an MSIT and a segment S in the MSIT, for any uni-segments $E_1$ and $E_2$ ($E_1 \neq E_2$) within segment S, $F(E_1, E_2)$ is an invariant (denoted $F_l(S)$) if $E_1$ is left to $E_2$, and $F(E_1, E_2)$ is another invariant (denoted $F_r(S)$) if $E_1$ is right to $E_2$.*

**Lemma 2** *Given an MSIT and a segment S in the MSIT, for any uni-segments $E$ and $E'$, if $E$ in segment $S$, $E'$ in segment $S'$ and $S \neq S'$, $F(E, E')$ is an invariant (denoted $F(S, S')$).*

**Lemma 3** *Given an MSIT and a segment S in the MSIT, for any uni-segment E is within segment S, G(E) and H(E) are invariants (denoted G(S) and H(S), respectively).*

Although the $F, G$ and $H$ coefficient functions are defined for the relation between uni-segments in (8)–(10), Lemmas 1–3 enable us to compute these functions based on segments rather than uni-segments. Because the number of segments in an MSIT may be much smaller than the number of uni-segments in the MSIT, we can compute these coefficient functions for much reduced costs. These coefficient functions will be computed before the sizing procedure and viewed as constants during wiresizing procedure. Furthermore, Lemmas 1–3 are helpful for us to present and prove the optimal MSWS solution properties later on in this paper.

### 3.3 Properties of Optimal MSWS Solutions

This subsection first presents Theorems 1–3, which are counterparts of the separability, the monotone property and the dominance property for single-source wiresizing [11]. Proofs for these theorems can be found in the appendix of this paper, which are similar to those in [11], except that our proofs can handle the multiple sources in the interconnect. Then, the *local monotone property* unique for the multi-source wiresizing is given as Theorem 4, just after Lemmas 4–7 leading to it. Theorem 4 is a must for the bundled wiresizing algorithm to be presented in Section 5.1.

**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.*

The first part of Theorem 1 is the separability between LSTs, and the second part is the separability within an LST. Because the separability may not hold within the $SST$, the $MSWS$ problem has much higher complexity than the $SSWS$ problem in general.

**B. LST Monotone Property**

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

In essence, an $LST$ is an $SSIT$ with its driver located at the branching node from the $SST$. By replacing the left-right relation in the $LST$ with the ancestor-descendent relation in an $SSIT$, the separability within an LST is just like the separability for single-source wiresizing, and the LST monotone property just like the monotone property for single-source wiresizing. Because the optimal wiresizing algorithm OWSA developed in [11] for single-source wiresizing is based on the separability and the monotone property, it can be applied for an LST when given the wire width assignments for the SST.

Note that the strict monotone is not needed by Theorem 2, and it is not needed in Theorem 4 presented later in this paper, either. Furthermore, it is worthwhile to mention that this monotone property just holds within each $LST$. The root uni-segment in an $LST$ may be wider than the uni-segment from which the $LST$ branches off (see Figure 4). This example also shows that the monotone property like that in the $SSWS$ problem does not hold for any particular source on an MSIT.

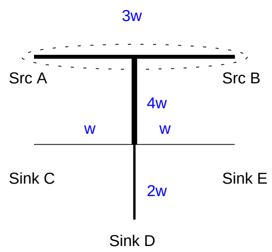



Figure 4: The optimal wire width assignments for a two-source net with $w$ being the minimum wire width. The $SST$ is surrounded by the dashed curve. Segments outside the curve belong to an $LST$. The wire width assignment is *monotone* in the $LST$. However, the root uni-segment of the $LST$ is wider than uni-segments in the $SST$.

**C. Dominance Property**

**Theorem 3** *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 [11] 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 3 enables efficient computations of lower and upper bounds of the optimal wiresizing solution by the *GWSA* algorithm in [11]. It applies the *local refinement* operation iteratively for each uni-segment to compute the lower bound or the upper bound of the optimal wiresizing solution. A much more powerful refinement operation, called the *bundled refinement* operation, which may compute the lower bound or the upper bound for a number of uni-segments in a single operation, will be introduced in Section 4.2.

**D. SST Local Monotone Property**

Although the signal direction is changeable in the uni-segments of the $SST$ when different sources are active, surprisingly, our study shows that optimal $MSWS$ solutions still satisfy a local monotone property (Theorem 4).

We shall prove the *SST local monotone property* based on a series of lemmas.

**Lemma 4** *Given an MSIT and a segment $S$ in the MSIT, if $E_l$ and $E_r$ are two adjacent edges within segment $S$ and $E_l$ is just left to $E_r$, then*

$$\begin{aligned} F(E_l, E) &= F(E_r, E) \quad if\ E \neq E_l \neq E_r \\ F(E, E_l) &= F(E, E_r) \quad if\ E \neq E_l \neq E_r \end{aligned}$$

**Proof:** There are two cases for Lemma 4.

Case 1: $E$ is in segment $S$, according to Lemma 1

if $E$ is left to $E_l$ (as well as $E_r$), $F(E_l, E) = F(E_r, E) = F_r(S)$ and $F(E, E_l) = F(E, E_r) = F_l(S)$.

if $E$ is right to $E_l$ (as well as $E_r$), $F(E_l, E) = F(E_r, E) = F_l(S)$ and $F(E, E_l) = F(E, E_r) = F_r(S)$.

Case 2: $E$ is in segment $S'(\neq S)$, according to Lemma 2

$F(E_l, E) = F(E_r, E) = F(S, S')$ and $F(E, E_l) = F(E, E_r) = F(S', S)$. □

**Lemma 5** *Given an MSIT and a segment $S$ in the MSIT, let $E_l$ and $E_r$ be uni-segments in segment $S$ with $E_l$ just left to $E_r$, concerning the optimal wiresizing solution, if $F_l(S) > F_r(S)$, then $E_l$ can not be narrower than $E_r$; if $F_l(S) < F_r(S)$, then $E_l$ can not be wider than $E_r$.*

**Proof:** Let $M = MSIT - E_l, E_r$ and $\overline{\mathcal{W}}$ be the wiresizing solution defined on $M$ by $\mathcal{W}$. The objective function (7) can be written as:

$$t(MSIT, \mathcal{E}, \mathcal{W})$$

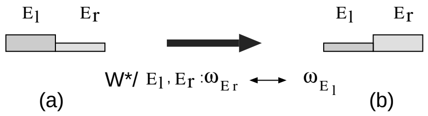



Figure 5: (a) The width assignments for $E_l$ and $E_r$ in the optimal solution $\mathcal{W}^*$. (b) The wiresizing obtained by swapping the width assignments for $E_l$ and $E_r$.

$$
\begin{aligned}
= \quad & t(M, \mathcal{E}, \overline{\mathcal{W}}) \; + \; \mathcal{K}_1 \cdot (w_{E_l} \; + \; w_{E_r}) \; + \\
& \mathcal{K}_2 \cdot \sum_{E' \in MSIT} F(E_l, E') \cdot \frac{w_{E'}}{w_{E_l}} \; + \; \mathcal{K}_2 \cdot \sum_{E \in MSIT} F(E, E_l) \cdot \frac{w_{E_l}}{w_E} \; + \\
& \mathcal{K}_2 \cdot \sum_{E' \in MSIT} F(E_r, E') \cdot \frac{w_{E'}}{w_{E_r}} \; + \; \mathcal{K}_2 \cdot \sum_{E \in MSIT} F(E, E_r) \cdot \frac{w_{E_r}}{w_E} \; + \\
& \mathcal{K}_3 \cdot \sum_{E' \in MSIT} F(E_l, E') \cdot \frac{1}{w_{E_l}} \; + \; \mathcal{K}_4 \cdot G(E_l) \cdot \frac{1}{w_{E_l}} \; + \; \mathcal{K}_5 \cdot H(E_l) \cdot \frac{1}{w_{E_l}} \; + \\
& \mathcal{K}_3 \cdot \sum_{E' \in MSIT} F(E_r, E') \cdot \frac{1}{w_{E_r}} \; + \; \mathcal{K}_4 \cdot G(E_r) \cdot \frac{1}{w_{E_r}} \; + \; \mathcal{K}_5 \cdot H(E_r) \cdot \frac{1}{w_{E_r}} \qquad (11)
\end{aligned}
$$

Let $\mathcal{W}^*$ be the optimal wiresizing solution. After swapping the width assignments for $E_l$ and $E_r$ with respect to $\mathcal{W}^*$ (see Fig. 5), we denote the resulted wiresizing solution $\mathcal{W}^*/E_l, E_r : w_{E_l} \leftrightarrow w_{E_r}$.

According to Lemmas 3 and 4

$$G(E_l) \; = \; G(E_r) \; = \; G(S) \qquad (12)$$

$$H(E_l) \; = \; H(E_r) \; = \; H(S) \qquad (13)$$

$$F(E_l, E) \; = \; F(E_r, E) \quad if \; E \neq E_r \neq E_l \qquad (14)$$

$$F(E, E_l) \; = \; F(E, E_r) \quad if \; E \neq E_r \neq E_l \qquad (15)$$

thus,

$$
\begin{aligned}
& t(MSIT, \mathcal{E}, \mathcal{W}^*/E_l, E_r : w_{E_l} \leftrightarrow w_{E_r}) \; - \; t(MSIT, \mathcal{E}, \mathcal{W}^*) \\
= \quad & \mathcal{K}_2 \cdot F(E_l, E_r) \cdot (\frac{w^*_{E_l}}{w^*_{E_r}} - \frac{w^*_{E_r}}{w^*_{E_l}}) \; + \; \mathcal{K}_2 \cdot F(E_r, E_l) \cdot (\frac{w^*_{E_r}}{w^*_{E_l}} - \frac{w^*_{E_l}}{w^*_{E_r}}) \; + \\
& \mathcal{K}_3 \cdot F(E_l, E_r) \cdot (\frac{1}{w^*_{E_r}} - \frac{1}{w^*_{E_l}}) \; + \; \mathcal{K}_3 \cdot F(E_r, E_l) \cdot (\frac{1}{w^*_{E_l}} - \frac{1}{w^*_{E_r}}) \\
= \quad & \{F(E_l, E_r) \; - \; F(E_r, E_l)\} \cdot \{w^*_{E_l} \; - \; w^*_{E_r}\} \cdot \{\mathcal{K}_2 \cdot \frac{(w^*_{E_l} \; + \; w^*_{E_r}) \; + \; \mathcal{K}_3}{w^*_{E_l} \cdot w^*_{E_r}}\} \qquad (16)
\end{aligned}
$$

We know that $\frac{(w^*_{E_l} \, + \, w^*_{E_r}) \, + \, \mathcal{K}_3}{w^*_{E_l} \cdot w^*_{E_r}} > 0$ and (16) $\geq 0$ since $\mathcal{W}^*$ is the optimal solution. Clearly, if $F_l(S) > F_r(S)$, according to Lemma 1, $F(E_l, E_r) > F(E_r, E_l)$, then we have $w^*_{E_l} \geq w^*_{E_r}$. Similarly, if $F_l(S) < F_r(S)$, then $F(E_l, E_r) < F(E_r, E_l)$, so we have $w^*_{E_l} \leq w^*_{E_r}$. As a result, Lemma 5 holds. □

By applying Lemma 5 to any adjacent uni-segments in a segment, we obtain Lemma 6.

**Lemma 6** *Given an MSIT and a segment $S$ in the MSIT, concerning the optimal wiresizing solution, if $F_l(S) > F_r(S)$, then the wire widths within segment $S$ decrease monotonically rightward. Similarly, they increase monotonically rightward if $F_l(S) < F_r(S)$.*

**Lemma 7** *Given an MSIT and any segment $S$ in the MSIT, if $F_l(S) = F_r(S)$, there exists an optimal wiresizing such that all uni-segments in segment $S$ have the same wire width.*

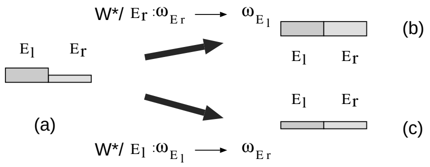



Figure 6: (a) The width assignments for $E_l$ and $E_r$ in the optimal solution $\mathcal{W}^*$. (b) The wiresizing obtained by replacing the width of $E_r$ with that of $E_l$. (b) The wiresizing obtained by replacing the width of $E_l$ with that of $E_r$.

**Proof:** Assume that Lemma 7 fails for an MSIT, then for any optimal solution $\mathcal{W}^*$, there must exist two uni-segments in a segment S of the MSIT such that $E_l$ is just left to $E_r$ and $w^*_{E_l} \neq w^*_{E_r}$. Since $\mathcal{W}^*$ is optimal, the increase in the objective function when we change the width of $E_r$ in $\mathcal{W}^*$ from $w^*_{E_r}$ to $w^*_{E_l}$ (see Figure 6.b), by using (11), is:

$$\begin{aligned}
\Delta t1 &= t(MSIT, \mathcal{E}, \mathcal{W}^*/E_r : w_{E_r} \rightarrow w_{E_l}) \;-\; t(MSIT, \mathcal{E}, \mathcal{W}^*) \\
&= \mathcal{K}_1 \cdot (w_{E_l} - w_{E_r}) \;+\; \mathcal{K}_2 \cdot \sum_{E' \in MSIT} F(E_r, E') \cdot (\frac{w_{E'}}{w_{E_l}} - \frac{w_{E'}}{w_{E_r}}) \;+ \\
&\quad \mathcal{K}_2 \cdot \sum_{E \in MSIT} F(E, E_r) \cdot (\frac{w_{E_l}}{w_E} - \frac{w_{E_r}}{w_E}) \;+\; \mathcal{K}_3 \cdot \sum_{E' \in MSIT} F(E_r, E') \cdot (\frac{1}{w_{E_l}} - \frac{1}{w_{E_r}}) \;+ \\
&\quad \mathcal{K}_4 \cdot G(E_r) \cdot (\frac{1}{w_{E_l}} - \frac{1}{w_{E_r}}) \;+\; \mathcal{K}_5 \cdot H(E_r) \cdot (\frac{1}{w_{E_l}} - \frac{1}{w_{E_r}}) \\
&\geq 0
\end{aligned} \tag{17}$$

Similarly, the increase in the objective function when change the width of $E_l$ in $\mathcal{W}^*$ from $w^*_{E_l}$ to $w^*_{E_r}$ (see Figure 6.c) is:

$$\begin{aligned}
\Delta t2 &= t(MSIT, \mathcal{E}, \mathcal{W}^*/E_l : w_{E_l} \rightarrow w_{E_r}) \;-\; t(MSIT, \mathcal{E}, \mathcal{W}^*) \\
&= \mathcal{K}_1 \cdot (w_{E_r} - w_{E_l}) \;+\; \mathcal{K}_2 \cdot \sum_{E' \in MSIT} F(E_l, E') \cdot (\frac{w_{E'}}{w_{E_r}} - \frac{w_{E'}}{w_{E_l}}) \;+ \\
&\quad \mathcal{K}_2 \cdot \sum_{E \in MSIT} F(E, E_l) \cdot (\frac{w_{E_r}}{w_E} - \frac{w_{E_l}}{w_E}) \;+\; \mathcal{K}_3 \cdot \sum_{E' \in MSIT} F(E_l, E') \cdot (\frac{1}{w_{E_r}} - \frac{1}{w_{E_l}}) \;+ \\
&\quad \mathcal{K}_4 \cdot G(E_l) \cdot (\frac{1}{w_{E_r}} - \frac{1}{w_{E_l}}) \;+\; \mathcal{K}_5 \cdot H(E_l) \cdot (\frac{1}{w_{E_r}} - \frac{1}{w_{E_l}}) \\
&\geq 0
\end{aligned} \tag{18}$$

It is not difficult to verify the following (19) by using (12)–(15) to summarize (17) and (18).

$$\Delta t1 \;+\; \Delta t2 \;=\; 0 \tag{19}$$

According to (17)–(19), $\Delta t1 = \Delta t2 = 0$. That is, both $\mathcal{W}^*/E_r : w_{E_r} \rightarrow w_{E_l}$ and $\mathcal{W}^*/E_l : w_{E_l} \rightarrow w_{E_r}$ are optimal!

Therefore, if there is an optimal solution $\mathcal{W}^*$ where the wire widths are not uniform in a segment S, let $w_l^*$ be the wire width of the leftmost uni-segment in S, from left to right, we can successively replace the wire width for every uni-segment in S by $w_l^*$, without increase in the objective function. That is, the resulting wiresizing solution is an optimal wiresizing solution such that the wire widths are uniform in any segment S. □

It is worthwhile to mention that Lemma 6 is more general than Lemma 7 in the sense that the former holds for any optimal solution while the later does not deny this possibility that there may exist an optimal solution with non-uniform wire widths in a segment $S$ when $F_l(S) = F_r(S)$. Nevertheless, Lemmas 6 and 7 are sufficient for us to draw the following important conclusion, which is needed by the bundled wiresizing algorithm to be presented in Section 5.1.

**Theorem 4 (Local Monotone Property)** *There exists an optimal wiresizing solution for an MSIT, such that the wire widths within each segments is monotone: (1) if $F_l(S) > F_r(S)$, the wire widths within $S$ decrease monotonically rightward. (2) if $F_l(S) = F_r(S)$, the wire within $S$ have a same width. (3) if $F_l(S) < F_r(S)$, the wire widths within $S$ increase monotonically rightward.*

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

# 4 Properties of Optimal MSWS/E Solutions

Up to now, both the *MSWS* problem defined in this paper and the *SSWS* problem studied in [11, 7, 23, 26, 27, 20, 30] are only studied in the context of an *a priori* fixed segment-division. The segment-division controls how often the wire width is allowed to change. However, it is difficult to choose a proper segment-division. For best accuracy, a very fine, uniform segment-division needs to be chosen, which results in much high memory usage and computation time due to the large number of uni-segments. We now investigate methods to obtain the optimal wiresizing results using a *non-uniform* and *coarser* segment-division.

A novel contribution of our work is to introduce an *MSWS* formulation based on a *variable* segment-division. The segment-division maybe finer in some regions but coarser in others. Moreover, we begin with a coarser segment-division 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 Segment-Division Refinement and Bundled-Segment

Assume that *minLength* is a constant determined by the users such that the wire widths are allowed to change every *minLength* long, in other words, *minLength* is the minimum length that a uni-segment can be. Clearly, *minLength* controls the optimality of the wiresizing solution.

Given an MSIT, let $\mathcal{E}_0$ be the segment-division where each uni-segment is a segment in the MSIT, and $\mathcal{E}_F$ the uniform segment-division where each uni-segment is *minLength* long.[3] Given two segment-divisions $\mathcal{E}$ and $\mathcal{E}'$, if each uni-segment in segment-division $\mathcal{E}$ corresponds to one or several uni-segments in segment-division $\mathcal{E}'$, we say that $\mathcal{E}'$ is a refinement of $\mathcal{E}$. An segment-division $\mathcal{E}$ is *valid* only if $\mathcal{E}$ is a refinement of $\mathcal{E}_0$ and the length of every uni-segment is a multiple of *minLength*. Clearly, among all valid segment-divisions, $\mathcal{E}_0$ is coarsest and $\mathcal{E}_F$ is finest. Furthermore, although we assume that the segment-division is *a priori* fixed and uniform in all proofs of this paper, by using the finest segment-division as this uniform segment-division, it is easy to find out that all lemmas and theorems still hold for any valid segment-division as long as they hold under the finest segment-division.

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

**Formulation 2** *Given an MSIT, the minimum uni-segment length minLength, and a set of possible wire width choices, the MSWS/E problem for delay minimization is to determine both an segment-division $\mathcal{E}$ and a wiresizing solution $\mathcal{W}$, such that the weighted delay $t(MSIT, \mathcal{E}, \mathcal{W})$ is minimized.*

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

**Definition 1** *Given two wiresizing solutions $\mathcal{W}$ and $\mathcal{W}'$, we define $\mathcal{W}'$* dominates *$\mathcal{W}$ if $w'_E \geq w_E$ for every uni-segment $E$ under the finest segment-division $\mathcal{E}_{\mathcal{F}}$.*

The concept of *bundled-segment* will be defined in order to achieve a segment-division as coarse as possible without loss of wiresizing accuracy.

**Definition 2** *Given an MSIT, a segment $S$ and the finest segment-division $\mathcal{E}_F$, let $E_1, \cdots, E_p$ be a maximal sequence of successive uni-segments in $S$ under $\mathcal{E}_F$ such that all uni-segments in this sequence have the same wire width in the optimal wiresizing solution under $\mathcal{E}_F$, We say that these uni-segments in the sequence form a bundled-segment.*

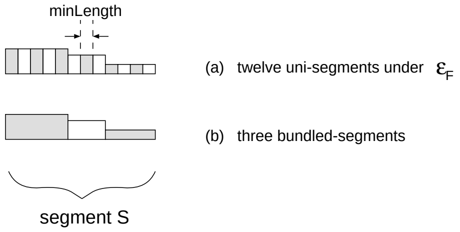



Figure 7: (a) The optimal wiresizing solution for segment $S$ with twelve uni-segments under the finest segment-division $\mathcal{E}_F$. (b) Segment $S$ contains only three bundled-segments which define a coarser segment-division with fewer computation costs to achieve the wiresizing accuracy same as that obtained by $\mathcal{E}_F$.

[3]For simplicity of presentation, we assume that the length of any segment in an MSIT is an integral multiple of *minLength*.

Figure 7 illustrates the concept of the bundled-segment by showing the optimal wiresizing solution for segment $S$ in an MSIT. It has twelve uni-segments under the finest segment-division $\mathcal{E}_F$ (see Figure 7.a), and just three bundled-segments (see Figure 7.b). Clearly, the segment-division defined by the bundled-segments can achieve the wiresizing accuracy same as that obtained by the finest segment-division $\mathcal{E}_F$.

For a long segment or a small *minLength* in order to achieve a more optimized wiresizing solution, the number of uni-segments under the finest segment-division tends to be quite large while the number of bundled-segments in the segment is always bounded by a really small constant, as given by the following Corollary 1 to the local monotone property (Theorem 4).

**Corollary 1** *Each segment in an MSIT has at most $r$ bundled-segments where $r$ is the number of possible wire width choices.*

Obviously, using the segment-division defined by the bundled-segments can achieve the required optimal solution for the most low costs. An operation which computes the optimal width for a bundled-segment directly, instead of treating it as a sequence of uni-segments under the finest segment-division $\mathcal{E}_F$, will be presented in the next subsection.

## 4.2 Bundled Refinement Operations

Let $\mathcal{W}$ be a wiresizing solution which dominates the optimal solution $\mathcal{W}^*$, and $E$ be a uni-segment under the current segment-division $\mathcal{E}$ and in segment $S$. Without loss of generality, we assume $F_l(S) \geq F_r(S)$ and treat $E$ as two uni-segments $E_l$ and $\overline{E_l}$. $E_l$ is the leftmost part of $E$, with length *minLength* (recall *minLength* is the length for a uni-segment in the finest segment-division $\mathcal{E}_F$); $\overline{E_l}$ is the remaining part of $E$. Let $\tilde{w}_{E_l}$ be the locally optimized width for $E_l$ based on the objective function (7) while keeping the width assignment of $\mathcal{W}$ on $\overline{E_l}$ and any uni-segment $E'$ other than $E$. Then, $\tilde{w}_{E_l}$ is regarded as a refined upper bound of the *entire* uni-segment $E$ (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 $F_l(S) \geq F_r(S)$, in the optimal solution $\mathcal{W}^*$, $E_l$ is always wider than all uni-segments under $\mathcal{E}_F$ in $\overline{E_l}$ (according to the local monotone property). The refinement of an upper bound of $w^*_{E_l}$ is still an upper bound of it (according to the dominance property), thus also gives an (possibly refined) upper bound of the optimal wire width assignments for any uni-segment under $\mathcal{E}_F$ in $\overline{E_l}$. Note that $E$ will not be divided into $E_l$ and $\overline{E_l}$ when performing the *BRU* operation on uni-segments 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 uni-segments $E_r$ and $\overline{E_r}$. $E_r$ is the rightmost part of $E$, with length *minLength*; $\overline{E_r}$ is the remaining part of $E$. Let $\tilde{w}_{E_r}$ be the locally optimized width for $E_r$ based on the objective function (7) while keeping the assignment of $\mathcal{W}$ on $\overline{E_r}$ and any uni-segment $E'$ other than $E$. Then, $\tilde{w}_{E_r}$ is regarded as a refined lower bound of the *entire* uni-segment $E$.

Concerning the bundled refinement, a property similar to the dominance property for the local refinement will be given as Theorem 5, which leads to the bundled wiresizing algorithm in Section 5.1. In the following, Lemmas 8–9 are first presented and proved. The reader may go to Theorem 5 directly by taking it for granted, without any difficulty to understand the algorithm in Section 5.1.

**Lemma 8** *Given an MSIT, a segment-division $\mathcal{E}$ and a wiresizing solution $\mathcal{W}$, for any particular uni-segment $E$ under $\mathcal{E}$, if we divide $E$ into a sequence of uni-segments $E_1, E_2, \cdots$, and $E_k$, let each new uni-segment inherits the wire width assignment of $E$, and denote the resulting segment-division and wiresizing solution $\mathcal{E}'$ and $\mathcal{W}'$, respectively, then the following relations with respect to (20)-(22) defined in Appendix hold for any uni-segment $E'$ other than $E$.*

$\Psi(MSIT, \mathcal{E}, E', \mathcal{W}) = \Psi(MSIT, \mathcal{E}', E', \mathcal{W}')$

$\Phi(MSIT, \mathcal{E}, E', \mathcal{W}) = \Phi(MSIT, \mathcal{E}', E', \mathcal{W}')$

$\Theta(MSIT, \mathcal{E}, E', \mathcal{W}) = \Theta(MSIT, \mathcal{E}', E', \mathcal{W}')$ [4]

**Proof:** It is not difficult to verify that Lemma 8 is true if the followings hold:

$$F(E_1, E') = F(E_2, E') = \cdots = F(E, E')$$
$$F(E', E_1) = F(E', E_2) = \cdots = F(E', E)$$
$$G(E_1) = G(E_2) = \cdots = G(E)$$
$$H(E_1) = H(E_2) = \cdots = H(E)$$

Assuming uni-segment $E$ is in segment $S$. There are two cases for uni-segment $E'$:

Case 1: $E'$ is also in the same segment S, according to Lemma 1, if $E$ is left to $E'$,

$$F(E_1, E') = F(E_2, E') = \cdots = F(E, E') = F_l(S)$$
$$F(E', E_1) = F(E', E_2) = \cdots = F(E', E) = F_r(S)$$

if $E$ is right to $E'$,

$$F(E_1, E') = F(E_2, E') = \cdots = F(E, E') = F_r(S)$$
$$F(E', E_1) = F(E', E_2) = \cdots = F(E', E) = F_l(S)$$

Case 2: $E'$ is in segment $S'$ different from segment $S$, according to Lemma 2

$$F(E_1, E') = F(E_2, E') = \cdots = F(E, E') = F(S, S')$$
$$F(E', E_1) = F(E', E_2) = \cdots = F(E', E) = F(S', S)$$

[4] In general, under the modeling method used in [11, 7, 5, 23], for the Elmore delay $t^{ij}$ between source $N_i$ and sink $N_j$, we have $t^{ij}(\mathcal{E}, \mathcal{W}) = t^{ij}(\mathcal{E}', \mathcal{W}')$. I.e., the Elmore delay is independent of the segment-division when given the wiresizing solution.

Again assuming uni-segment $E$ is in segment $S$, according to Lemma 3

$$G(E_1) = G(E_2) = \cdots = G(E) = G(S)$$
$$H(E_1) = H(E_2) = \cdots = H(E) = H(S)$$

As a result, Lemma 8 holds. $\square$

According to Lemma 8 and (23) defined in Appendix for the local refinement, we can conclude that the following Lemma 9 holds.

**Lemma 9** *When given the wiresizing solution $\mathcal{W}$ and any particular uni-segment $E$, the local refinement result for $E$ with respect to $\mathcal{W}$ is independent of the segment-division of uni-segments other than $E$.*

**Theorem 5 (Bundled Refinement Property)** *Let $\mathcal{W}^*$ be an optimal wiresizing solution under $\mathcal{E}_F$. If a wiresizing solution $\mathcal{W}$ dominates $\mathcal{W}^*$, then the wiresizing solution obtained by any BRU operation under any segment-division $\mathcal{E}$ 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 segment-division $\mathcal{E}$ on $\mathcal{W}$ is still dominated by $\mathcal{W}^*$.*

**Proof:** For any particular uni-segment $E$ under the current segment-division $\mathcal{E}$ in segment $S$ of an $MSIT$, it may be divided into $k$ uni-segments under the finest segment-division $\mathcal{E}_F$. From left to right, let them be $E_l = E_{F1}, E_{F2}, E_{F3}, ..., E_{Fk} = E_r$.

Without loss of generality, we assume $F_l(S) \geq F_r(S)$. For a wiresizing solution $\mathcal{W}$ which dominates the optimal solution $\mathcal{W}^*$, let $\mathcal{W}^b$ be the wiresizing solution after performing an $BRU$ operation of $\mathcal{W}$ on $E$ under $\mathcal{E}$. Then, we have $w_l^b = w_{F2}^b = w_{F3}^b = ... = w_r^b$ according to the definition of the $BRU$ operation. Meanwhile, let $\mathcal{W}^*$ be the optimal solution. We have $w_l^* \geq w_{F2}^* \geq w_{F3}^* \geq ... \geq w_r^*$ according to the local monotone property.

According to Lemma 9, $w_l^b$ is also the local refinement result for uni-segment $E_l$ under $\mathcal{E}_F$. Therefore, $w_l^b \geq w_l^*$. As a result, $w_l^b = w_{F2}^b = w_{F3}^b = ... = w_r^b \geq w_l^* \geq w_{F2}^* \geq w_{F3}^* \geq ... \geq w_r^*$. Recall that the bundled refinement of uni-segment $E$ does not change the wire width for any uni-segment $E'$ other than $E_l, E_{F2}, E_{F3}, ..., E_r$. Thus, $\mathcal{W}^b$ still dominates $\mathcal{W}^*$.

The $BRL$ case can be proved in a similar way. $\square$

# 5 Optimal Wiresizing Algorithm

## 5.1 Bundled Wiresizing Algorithm

The greedy wiresizing algorithm GWSA was given in [11]. Working on an *a priori* defined segment-division, GWSA can use *local refinement* operations to compute the lower/upper bounds of the optimal wiresizing solution starting with the minimum and maximum wire width assignments, respectively. Our *bundled wiresizing algorithm (BWSA)* (Table 1) computes the lower/upper bounds of the optimal wiresizing solution for an *MSIT*, also starting with the minimum and maximum wire width assignments, respectively, but using *bundled* *refinement* operations instead of *local refinement* operations, and a gradually refined segment-division rather than a fixed one. BWSA achieves the same optimal lower and upper bounds for much less computation costs when compared with GWSA.

### A. Overview

Starting with the coarsest segment-division $\mathcal{E}_0$, we perform $BRU$ and $BRL$ iteratively through an *MSIT*. We assign the minimum width to all uni-segments (in this case, each uni-segment is a segment), then traverse *MSIT* and perform $BRL$ operation on each uni-segment. This process is repeated until no improvement is achieved on any uni-segment in the last round of traversal. Because the minimum wire width assignment is dominated by the optimal wiresizing solution, the result wiresizing solution is still dominated by the optimal solution and is a tighter lower bound of the optimal wiresizing solution. Similarly, we assign the maximum width to all uni-segments and perform $BRU$ operations, obtain a tighter upper bound of the optimal wiresizing solution. This is the first *pass* of $BWSA$.

After each *pass*, we check the lower and upper bounds. If there is a gap between the lower and upper bounds for an uni-segment (which is called an *non-convergent* uni-segment) and it is still longer than the minimum uni-segment length $minLength$, we divide it into two uni-segments of the almost equal length (they may differ by $minLength$ in order to maintain a *valid* segment-division ), and let each uni-segment inherits the lower and upper bounds of their parent. After the refinement of all non-convergent uni-segments, another *pass* to tighten the lower/upper bounds is carried out by performing bundled refinement operations under the refined segment-division. Note that the bundled refinement is only needed for uni-segments who are just refined, because only these uni-segments are not convergent.

This BWSA algorithm iterates through a number of passes until we either have the identical lower and upper bound for all uni-segments under current division (in this case we get an optimal wiresizing solution), or each non-convergent uni-segment is $minLength$ long.

### B. Optimality

In order to discuss the optimality of the lower and upper bounds obtained by the BWSA algorithm, we define the following $\mathcal{E}_F$-tight lower and upper bounds.

**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 segment-division $\mathcal{E}_F$, $\mathcal{W}$ is an $\mathcal{E}_F$-tight upper bound. Similarly, $\mathcal{W}$ is an $\mathcal{E}_F$-tight lower bound if $\mathcal{W}$ is dominated by $\mathcal{W}^*$ and can not be further refined by any local refinement operation under $\mathcal{E}_F$.*

It is worthwhile to mention that there may be more than one $\mathcal{E}_F$-tight upper (or lower) bounds of an $\mathcal{W}^*$. An example of non-unique $\mathcal{E}_F$-tight bounds will be given in Section 6.2.B.

```
Function gBWSA_L/U(minLength, ℰ, 𝒲_lower/𝒲_upper)
  /*  Given the minimum uni-segment length minLength, an segment-division ℰ, a lower/upper
   *  bound 𝒲_lower/𝒲_upper of the optimal wiresizing solution, and a set of possible wire widths
   *  {W_1, W_2, ···, W_r}, return a tighter lower/upper bound.
   */
  𝒲 ← 𝒲_lower/𝒲_upper;
  do
      progress ← false;
      for each uni-segment E of ℰ do
          w ← BRL(minLength, ℰ, 𝒲_lower, E) or BRU(minLength, ℰ, 𝒲_upper, E);
          if w ≠ 𝒲(E) then
              progress ← true;    𝒲(E) ← w;
          end if
      end for;
  while progress = true;
  return 𝒲;
end Function;

Function SBSR(minLength, ℰ, 𝒲_lower, 𝒲_upper)
  /*  Given the minimum uni-segment length minLength, an segment-division ℰ, a lower bound
   *  and an upper bound of the optimal wiresizing solution, for each non-convergent uni-
   *  segment, if it is longer than minLength, divide it into two uni-segments, each inherits
   *  the bounds(Selective Binary Segment-division Refinement). Once the segment-division has
   */ been refined, set the flag not_done true.
  not_done ← false;
  for each uni-segment E of ℰ do
      if 𝒲_lower(E) ≠ 𝒲_upper(E), and E is longer than minLength
          divide E into two uni-segments with (nearly) equal lengths;
          not_done ← true;
  return ℰ and not_done;
end Function;

Function BWSA(MSIT, minLength)
  /*  Given an MSIT, the minimum uni-segment length minLength and a set of possible wire
   *  widths {W_1, W_2, ···, W_r}, return the lower and upper bounds of the optimal wiresizing
   *  solution of the MSIT by applying gBWSA_L() and gBWSA_U() with a recursively refined
   *  segment-division.
   */
  ℰ ← MSIT;    𝒲_lower ← W_1;    𝒲_upper ← W_r;
  do
      𝒲_lower ← gBWSA_L(minLength, ℰ, 𝒲_lower);
      𝒲_upper ← gBWSA_U(minLength, ℰ, 𝒲_upper);
      ℰ ← SBSR(minLength, ℰ, 𝒲_lower, 𝒲_upper);
  while not_done
  return 𝒲_lower and 𝒲_upper;
end Function;
```

Table 1: Bundled Wiresizing Algorithm (BWSA)

With this definition, we can prove the following important result concerning the optimality of the BWSA algorithm.

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

**Proof:** If $\mathcal{E}$ is the segment-division after BWSA, for any uni-segment $E$ under $\mathcal{E}$, $E$ is longer than minimum uni-segment length $minLength$ if and only if $E$ is an *convergent* uni-segment, whose bound can not be tightened any more.

If $E$ is $minLength$ long, according to Lemma 9, its lower and upper bounds obtained by bundled refinements are same as those obtained by local refinements under $\mathcal{E}_F$. Therefore, Theorem 6 holds. □

Basically, Theorem 6 suggests that the quality of the wiresizing solutions obtained by the BWSA algorithm starting from the *coarsest* segment-division is as good as those obtained by the GWSA algorithm under the *finest* segment-division $\mathcal{E}_F$.

### C. Complexity

In order to analyze the complexities for both GWSA and BWSA, the following notations will be introduced. Recall that our MSWS/E problem aims to find the optimal wiresizing solution for every wire which is $minLength$ long. In order to achieve the required accuracy, the finest segment-division $\mathcal{E}_F$ where each uni-segment is $minLength$ long must be used by GWSA, while BWSA can determine a proper, usually coarser, segment-division during the wiresizing procedure. For the simplicity of presentation, we will use *min-segment* exclusively for the uni-segment which is $minLength$ long. If we use $minLength$ as the wire length unit, the total wire length $n$ is a natural metric for the problem size.

We will prove the following Theorem 7.

**Theorem 7** *Given an MSIT and $r$ wire width choices, if the total wire length is $n$ when $minLength$ is regarded as the length unit, both GWSA and BWSA have the worst-case complexity of $O(n^3 \cdot r)$ for the MSWS/E problem.*

**Proof:** Two parts are to be presented with the first one for GWSA and the second one for BWSA.

**Part I:** Beginning with a wiresizing solution, GWSA traverses the MSIT to perform local refinement for every uni-segment. The *traversal* is repeated until there is no improvement on the last round of traversal. Note that if there are $m$ uni-segment in the MSIT, each traversal costs $O(m^2)$, because each uni-segment is locally refined exactly once, and each refinement takes $O(m)$. This complexity holds under the assumption that the coefficient functions F,G and H defined in (8)–(10) have been computed before the wiresizing procedure. Thus, the traversal has the same complexity for both SSWS and MSWS formulations except that it takes slightly higher costs to compute F, G and H in the MSWS formulation.

The finest segment-division $\mathcal{E}_F$ must be used for GWSA for the MSWS/E problem. Given an MSIT with total wire length of $n$, the total number of min-segments, i.e., uni-segments under the finest segment-division $\mathcal{E}_F$, is also $n$. Thus, each traversal in GWSA has complexity of $O(n^2)$.

Let all wiresizing solutions comprise a two-dimension space with one axis *wire width*, the other axis *wire length*. The worst-case complexity of GWSA is illustrated by Figure 8. Figure 8.a is the initial case with the wiresizing solution being the maximum wire width assignments, which is an upper bound of $\mathcal{W}^*$ and is represented by the dark line. Each traversal will generate a better solution that is closer to and still dominates $\mathcal{W}^*$. Figure 8.b is the worst case after a traversal where the area of solution space is reduced by one unit.

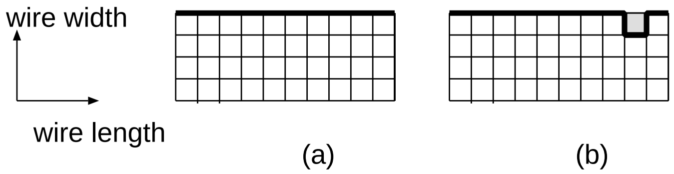



Figure 8: The wiresizing solutions (dark lines) before and after a traversal of local refinements on $n$ min-segment. (a). The initial wire sizing dominates the optimal solution $\mathcal{W}^*$ and the area of solution space is $(r-1)\cdot n$. (b). A worst case after a traversal where the area of solution space is only reduced by one unit (the pruned area is grey).

The overall worst case is that GWSA prunes the whole solution space into a line by total $(r-1)\cdot n$ traversals. Recall that a traversal costs $O(n^2)$, so the worst case complexity of GWSA is $O(n^3 \cdot r)$.

Note that the single-sided GWSA (either beginning with the maximum or the minimum wiresizing solutions) and the double-sided GWSA (beginning with the maximum and the minimum wiresizing solutions, respectively) have the same worst-case complexity, because the worst-case occurs when the whole solution space is pruned into a line (in this case, we obtain the optimal solution by just performing local refinements). Due to this reason, we will only consider the BRU (bundled refinement for upper bound) in Part II for the complexity analysis for BWSA.

**Part II:** In BWSA, the number of uni-segment $m$ is increased when the segment is refined recursively, clearly $m$ is always bounded from the above by the total wire length $n$, thus the complexity of a traversal in BWSA is $O(m^2) = O(n^2)$.

However, there maybe two types of traversals in BWSA. One is *effective traversal* which prunes the solution space by at least one unit, thus the total number of effective traversal is less than $(r-1)\cdot n$. The other is *non-effective traversal* which does not prune the solution space and its total number is less than $\ln n$ due to our binary segment refinement scheme. Thus, the worst case complexity of BWSA is $O((r-1)\cdot n + \ln n)\cdot O(n^2) = O(n^3 \cdot r)$.
□

It is worthwhile to mention that one bundled refinement will prune the solution space by $n_0$ units, if the uni-segment is $n_0$ long. In practice, the final uni-segment is often much longer than min-segment, thus BWSA runs much faster than GWSA. Such experiments will be presented in Section 6.2. In fact, because BWSA runs much faster than GWSA and obtains the lower/upper bounds same tight as those obtained by GWSA, we always use BWSA instead of GWSA, except for the comparison aim.

### 5.2 Optimal Wiresizing Algorithm Using Bundled Refinement

Based on the BWSA algorithm, an efficient optimal wiresizing algorithm using bundled refinement (*OWBR* algorithm) has been developed. Given an *MSIT*, OWBR first computes a lower bound and an upper bound of the optimal wiresizing solution using BWSA. If the lower and upper bounds meet, which is very likely in practice, we get the optimal wiresizing solution immediately. Otherwise, the optimal solution shall be found between the lower and upper bounds where them do not meet.

Because the separability and the monotone property hold in LSTs, the optimal single-source wiresizing algorithm, named *OWSA* [11], can be used for LSTs. The OWSA algorithm is a polynomial-time algorithm based on the dynamic programming method. However, since the separability in SST does not hold in general, the wire width assignments for non-convergent uni-segments in the SST need to be enumerated subject to the local monotone property.

Thus, if the lower and upper bounds obtained by BWSA do not meet for all uni-segments, OWBR first enumerates the wire width assignments for the SST between the lower and upper bounds and subject to the local monotone property. Then, OWSA is applied independently to each LST to compute an optimal wiresizing solution between the lower and upper bounds during each enumeration of the *SST*.

Our experiments show that BWSA gives the convergent bounds on all uni-segments in an MSIT for almost all cases. For those cases which have non-convergent uni-segments, the percentage of non-convergent uni-segments is very small. Moreover, the gap between the lower and upper bounds on each non-convergent uni-segment is also very small (usually being one in our experiments). Therefore, OWBR runs very fast in practice.

## 6 Experimental Results

We have implemented the OWBR algorithm in ANSI C for the Sun SPARC station environment and tested our algorithm on a large number of *MSIT*s for MCM and submicron IC technologies, including several multi-source nets extracted from the layout of an Intel high performance microprocessor. In this section, we shall present both the comparison of different wiresizing solutions and the comparison between the BWSA algorithm and the GWSA algorithm.

The parameters used in our experiments are summarized in Table 2. The IC technology parameters are based on the $0.5\mu m$ CMOS process from North Carolina Microelectronic Center (MCNC)[25], and the MCM technology parameters were obtained from [12]. The loading capacitance in IC technology is the total gate capacitance of a minimize size transistor in the MCNC $0.5\mu m$ CMOS, while the minimum loading capacitance in MCM technology is the pad capacitance of $1000fF$ on MCM substrate. The driver resistance is assumed to be fixed during wiresizing.

The wire width choices are $\{W, 2W, 3W, 4W, 5W\}$, where $W$ 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. The minimum uni-segment lengths are 10 $\mu m$ for the IC technology and 100 $\mu m$ for the MCM technology, respectively.

| Technology: | Integrated Circuits (ICs) | Multi-Chip Modules (MCMs) |
|---|---|---|
| Driver Resistance: | 156 $\Omega$ | 25 $\Omega$ |
| Wire Resistance ($\Omega/\Box$): | 0.044 | 0.02 |
| Loading Capacitance: | 3.720 $fF$ | 1000 $fF$ |
| Wire Capacitance (area) ($aF/\mu m^2$): | 41.3 | 3.46 |
| Fringing Capacitance (2 sides) ($aF/\mu m$): | 150 | 50.4 |
| Minimum Uni-segment Length $minLength$ : | 10 $\mu m$ | 100 $\mu m$ |

Table 2: Technology parameters based on submicron CMOS and MCM designs.

## 6.1 Comparison between Different Wiresizing Solutions

In this subsection, we will report SPICE-computed delay instead of calculated Elmore delay values. The use of SPICE simulation results not only shows the quality of our *MSWS* solutions, but also verifies the validity of our interconnect modeling and the correctness of our *MSWS* problem formulation.

Let *min_width* be the wiresizing solution with minimum wire width $W$ everywhere, *opt_ssws* be the wiresizing solution given by the optimal *SSWS* algorithms[11, 7], and *opt_msws* be the wiresizing solution given by our optimal *MSWS* algorithms. Also, let *wire length* denote the total wire length of a routing tree, and *normalized area* denote the area ratio of wiresizing solution versus the *min_width* wiresizing solution, which is equivalent to the average wire with if the minimum wire width $W$ is scaled to 1. Both *maximum delay* and *average delay* are only in terms of critical source-sink pairs. In the following experiments, we assign $\lambda^{ij} = 1$ for a critical source-sink pair and $\lambda^{ij} = 0$ otherwise, thus, the objective in equation (7) is equivalent to the *average delay* among critical source-sink pairs.

### A. Results on Simple Artificial Nets

Wiresizing results on the four-pin H-topology interconnect tree in Figure 1 are shown in Table 3, for both the CMOS technology and the MCM technology. Compared to the *min_width* solutions, the *opt_ssws* solutions may have larger maximum delay and consume larger routing area, while our *opt_msws* solutions reduce the maximum delay and the average delay by up to 32.7% and 25.5% for the CMOS technology, respectively, and

| Total Wire Length | 300 $\mu m$ | | | 3000 $\mu m$ | | |
|---|---|---|---|---|---|---|
| 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 |
| Total Wire Length | 3000 $\mu m$ | | | 30000 $\mu m$ | | |
| MCM Wiresizing | min_width | opt_ssws | opt_msws | min_width | opt_ssws | opt_msws |
| Maximum Delay (ns) | 0.417 | 0.425(+1.9%) | 0.406(-2.63%) | 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 | 1.83 | 1 | 2.49 | 3.44 |

Table 3: Comparison of *min_width*, *opt_ssws* and *opt_msws* solutions on a four-source tree with H-topology

up to 32.1% and 26.7% for the MCM technology, respectively. The single-source wiresizing method is clearly not applicable to the *MSWS* problem.

### B. Results on Random Nets

| IC ($\mu m$) | Normalized Area opt_msws | Maximum Delay ($ns$) min_width | Maximum Delay ($ns$) opt_msws | Average Delay ($ns$) min_width | Average Delay ($ns$) opt_msws |
|---|---|---|---|---|---|
| 23650 | 2.618 | 3.5595 | 3.1487 (-11.54%) | 2.3474 | 2.1153(-9.88%) |
| 18430 | 1.674 | 4.1752 | 4.0000 (-4.20%) | 2.5582 | 2.3982(-1.84%) |
| 13820 | 2.321 | 6.5797 | 5.4534 (-17.1%) | 2.2206 | 2.0319(-8.50%) |
| 12550 | 1.775 | 7.7205 | 6.3024 (-18.4%) | 4.5778 | 4.0538(-11.4%) |
| **MCM ($\mu m$)** | **Normalized Area opt_msws** | **Maximum Delay ($ns$) min_width** | **Maximum Delay ($ns$) opt_msws** | **Average Delay ($ns$) min_width** | **Average Delay ($ns$) opt_msws** |
| 236500 | 4.855 | 5.1556 | 4.1462 (-19.57%) | 3.4335 | 2.9410(-14.3%) |
| 184300 | 2.153 | 5.5607 | 4.6929 (-15.60%) | 3.2055 | 2.8979(-9.60%) |
| 138200 | 2.333 | 8.4482 | 6.9962 (-17.2%) | 2.7345 | 2.4349(-10.96%) |
| 125500 | 2.106 | 9.1050 | 8.2007 (-9.93%) | 5.4380 | 5.0314(-7.47%) |

Table 4: Multi-source wiresizing results on random nets

Ten eight-pin nets with pins randomly distributing in a $1000 \times 1000$ grid and routed by the 1-Steiner algorithm [19] are used in our experiments. For each eight-pin random net, eight timing-critical source-sink pairs are chosen randomly. Wiresizing is carried out under both the CMOS technology and the MCM technology by treating each grid edge as a uni-segment. Table 4 shows the comparison of the *min_width* and *opt_msws* wiresizing solutions for four of such nets. Clearly, *opt_msws* solutions consistently outperform the min_width solutions. On average, we observe 12.8% and 7.9% reduction of the maximum delay and the average delay among eight critical source-sink pairs in each net for the submicron IC technology, respectively, and 15.6% and 10.6% for the MCM technology, respectively.

### C. Results on Industrial Nets

We also tested our algorithms on several real multi-source nets provided by Intel [3]. 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 1-Steiner tree algorithm [19] to route these nets. Table 5 summarizes the routing trees for these nets.

| | total pin number | total segment number | total wire length ($\mu m$) |
|---|---|---|---|
| net1 | 3 | 4 | 3600 |
| net2 | 4 | 6 | 6600 |
| net3 | 9 | 13 | 10070 |
| net4 | 4 | 5 | 10570 |
| net5 | 11 | 10 | 16980 |
| net6 | 19 | 19 | 31980 |

Table 5: Routing trees for multi-source nets extracted from the layout for a high performance Intel microprocessor

We applied our OWBR algorithm to these routing trees and wiresizing results are shown in Table 6. The *opt_msws* solutions consistently outperform the *min_width* solutions with as much as 36.3% and 23.5% reduction on the maximum delay and the average delay, respectively.

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

| | Normalized Area | Maximum Delay ($ns$) | | Average Delay ($ns$) | |
|---|---|---|---|---|---|
| | opt_msws | min_width | opt_msws | min_width | opt_msws |
| net1 | 1.000 | 0.1435 | 0.1435 (0.0%) | 0.1260 | 0.1260(0.0%) |
| net2 | 1.044 | 0.2572 | 0.2567 (-0.2%) | 0.2004 | 0.1994(-0.50%) |
| net3 | 1.475 | 0.5230 | 0.4025 (-23.0%) | 0.3504 | 0.3241(-7.5%) |
| net4 | 2.000 | 0.5853 | 0.4873 (-16.7%) | 0.5007 | 0.3846(-23.2%) |
| net5 | 1.775 | 0.9496 | 0.7635 (-19.6%) | 0.6375 | 0.5711(-10.4%) |
| net6 | 2.706 | 3.4505 | 2.1979 (-36.3%) | 1.8968 | 1.4512(-23.5%) |

Table 6: Multi-source wiresizing results on several nets in an Intel microprocessor layout

## 6.2 Speed-up Using Variable Edge-Division

### A. Results on Random Nets

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

Table 7: Performance comparison between GWSA and BWSA

The ten eight-pin nets used in the above subsection were test for both CMOS and MCM technologies. We compared the CPU times to obtain the lower and upper bounds of the optimal wiresizing solution, used by GWSA under the finest edge-division and by BWSA under the refined edge-division, respectively. Results for multi-source wiresizing in the CMOS technology are reported in Table 7 with the speed-up ranging from a factor of two orders of magnitude to a factor of three orders of magnitude. In all these experiments, both *BWSA* and *GWSA* provide the convergent lower/upper bounds, i.e., the optimal solutions are achieved just by lower and upper bound computations. Since BWSA can be used for single-source wiresizing as well, we also test both algorithms again by assuming there is only one source together with eight timing critical sinks in each net. Similar speed-up factors were obtained.[5]

### B. Results on Industrial Nets

Furthermore, we applied both BWSA and GWSA algorithms to the set of Intel nets. Because the time to compute the lower/upper bounds for most nets in this set is too small to measure, we compared the total

[5] In the theoretical sense, BWSA tends to achieve even higher speed-up factor in case of a single source where the monotone wiresizing range is even larger, when compared with the multi-source case.

running time. In Table 8, the *BWSA-based* algorithm is just OWBR, i.e., BWSA to compute lower and upper bounds, followed by enumerating for the SST and OWSA for LSTs. The *GWSA-based* algorithm is just to replace BWSA by GWSA in the OWBR scheme. OWBR is observed to run more than 100x faster than the *GWSA*-based algorithm.

| | net1 | net2 | net3 | net4 | net5 | net6 |
|---|---|---|---|---|---|---|
| GWSA-based algorithm (s) | 0.07 | 8.18 | 172.37 | 15.67 | 38.10 | 227.92 |
| BWSA-based algorithm (s) | 0.07 | 0.15 | 0.37 | 0.37 | 0.97 | 3.37 |
| Speedup factor of BWSA-based algorithm | 1 | 54.5 | 465.8 | 42.3 | 39.3 | 67.63 |

Table 8: Total running time comparison between GWSA-based and BWSA-based algorithms

It is worthwhile to mention that BWSA gives identical lower and upper bounds for net3 while GWSA *does not*, which is also an example of existence of multiple $\mathcal{E}_F$-tight bounds for the optimal solution as mentioned in Section 5.1.B. Also note that, in case of OWBR, the total running time is *not* dominated by the time to compute lower and upper bounds, one reason is that the current implementation builds the data structure for the finest edge-division even if the *bundled refinement* does not need it at all. Thus, the total running time still can be further reduced in future implementation without building the data structure for the finest edge-division.

### 6.3 Fidelity of the Elmore Delay Model

The concept of *fidelity* of the Elmore delay model was introduced by Boese *et al* in [2] for the routing tree topology optimization, which was used to measure if an optimal or near-optimal solution according to the Elmore delay model is also also be nearly optimal according to the actual delay (e.g., computed using SPICE). We shall investigate the fidelity of the Elmore delay model for the optimal wiresizing problem to find out how good the solution given by our optimal wiresizing solution is in terms of the real delay.

We measured the *fidelity* on the real multi-source nets given in Table 5. We compared both the weighted average Elmore delay and the weighted average 50% delay computed by SPICE. The ranking technique similar to [2] was used: We first rank wiresizing solutions according to their Elmore delays, then rank them according to their SPICE-computed delay, and find the average difference between the two rankings. Because the number of total wiresizing solutions is prohibitively large to enumerate, we randomly generated 1,000 wiresizing solutions by assuming $minLength = 10\mu m$, computed both SPICE and Elmore delays of these solutions and ranked them.

Table 9 shows the fidelity of the Elmore delay model against the SPICE-computed 50% delay for wiresizing optimization. Besides the *ranking difference*, another metric *suboptimality* is presented as well. It is computed in the following way: Let the average ranking difference is $d$. For a wiresizing solution whose SPICE-computed delay ranking is $i$, we compute the relative difference between the $(i+d)$-th and $i$-th SPICE-computed delays, together with that between the $(i-d)$-th and $i$-th SPICE-computed delays. The larger one of the two relative differences is regarded as the possible suboptimality of this wiresizing solution (if it is regarded as optimum

| | Overall | | Best-100 (Elmore Delay) | | Best-10 (Elmore Delay) | |
|---|---|---|---|---|---|---|
| | Ranking Difference | Suboptimality | Ranking Difference | Suboptimality | Ranking Difference | Suboptimality |
| net1 | 10.07/1000 | 0.55% | 4.51/100 | 0.0195% | 0.70/10 | 0.0022% |
| net2 | 10.05/1000 | 0.12% | 4.41/100 | 0.0197% | 0.80/10 | 0.0023% |
| net3 | 2.068/1000 | 0.03% | 0.72/100 | 0.0039% | 0.00/10 | 0.0000% |
| net4 | 37.30/1000 | 0.26% | 12.6/100 | 0.0410% | 2.20/10 | 0.0150% |
| net5 | 56.95/1000 | 0.33% | 25.9/100 | 0.0580% | 3.00/10 | 0.0110% |
| net6 | 9.570/1000 | 0.04% | 3.95/100 | 0.0069% | 0.60/10 | 0.0018% |

Table 9: Average difference in ranking and the corresponding suboptimality for Intel nets based on $0.5\mu m$ CMOS technology

according to the Elmore delay model). We report in Table 9 the suboptimality average over all solutions in each column. Because it tells us how much the relative delay difference in terms of the SPICE-computed delay is for certain ranking difference, it is a more direct metric for fidelity.

Over the 1,000 random solutions for each net, the average ranking differences are between 2.068 and 56.95 of 1,000, and the average suboptimalities between 0.03% and 0.55%. Thus, on average, the optimal wiresizing solution according to the Elmore delay model may be 0.03%–0.55% worse than the SPICE-computed optimal solution.

We also choose the best 100 and 10 wiresizing solutions according to the Elmore delay model for each random solution set, respectively. Then, we compute the average ranking differences and the average suboptimalities. It is interesting to find that the better the wiresizing solutions according to the Elmore delay model, the less the suboptimality they have. For example, among the 1,000 random wiresizing solutions for net1, the average suboptimality is 0.55% for all 1,000 solutions, while only 0.0195% for the best 100 and, even less, 0.0022% for the best 10. If we assume that these random solutions are uniformly distributed in the overall solution space, it suggests that, in general, in the area near the optimum in the solution space, the Elmore delay model has a even higher fidelity.

Based on data of the best 10 for every random solution set, the optimal wiresizing solution according to the Elmore delay model is less than 0.015% worse than the optimal solution according to the SPICE-computed delay.[6] Thus, we believe that the Elmore delay model has really high fidelity for wiresizing optimization. Furthermore, this conclusion is applicable to all other wiresizing works [11, 7, 23] where the Elmore delay model is used as well.

In our SPICE simulation, the inductance is not taken into consideration under this assumption that the inductive effect is negligible under the current CMOS technology. Even though the higher-order delay model is used in [26, 27], the same assumption is made there. The fidelity will be studied in the future for the MCM technology where the inductive effect tends to be no longer negligible.[7]

[6] We also enumerate the wiresizing solutions for net1 and net2 by assuming that each segment in the routing tree has a uniform wire width. Even higher fidelity is observed when compared with this set of random wiresizing experiments. For these two nets, the Elmore delay model gives the best 5 solutions same as those according to the SPICE-computed delay.

[7] It was pointed out in [2] that the Elmore delay model still has high fidelity for topology optimization even when the inductance is taken into consideration in MCM technologies.

## 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 *(coarse) variable* segment-division, which achieves the same optimal wiresizing solutions as the optimal wiresizing algorithms based on the finest segment-division, but uses much less computational time.

In order to further reduce the interconnect delay in multi-source nets, we plan to study the simultaneous driver and wire sizing 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.


## Acknowledgements

The authors would like to thank Heming Chan 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.


## Appendix: Proofs of Theorems 1–3

### 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.*

**Proof**: In essence, the first part of the LST separability is the separability between $LSTs$, the second part is the separability within an $LST$. We shall first prove the the separability between $LSTs$.

Let $\mathcal{L} = \{LST_1, LST_2, \cdots LST_m\}$ be the set of loading subtrees branching off from the source subtree $SST$. To simplify the notations, we define:

$$\begin{aligned}
\mu(E,E') &= \mathcal{K}_2 \cdot F(E,E') \\
\omega(E,E') &= \mathcal{K}_3 \cdot F(E,E') \\
\nu(E) &= \mathcal{K}_4 \cdot G(E) \\
\phi(E) &= \mathcal{K}_5 \cdot H(E)
\end{aligned}$$

We then rewrite Eqn. (7) as a function of the loading subtrees in $\mathcal{L}$ and the source subtree $SST$:

$$\begin{aligned}
t(MSIT, R_d, \mathcal{W}) = & \sum_{E \in MSIT} \mathcal{K}_1 \cdot w_E + \sum_{E,E' \in MSIT, E \neq E'} \mu(E,E') \cdot \frac{w_{E'}}{w_E} + \sum_{E,E' \in MSIT, E \neq E'} \omega(E,E') \cdot \frac{1}{w_E} + \\
& \sum_{E \in MSIT} \nu(E) \cdot \frac{1}{w_E} + \sum_{E \in MSIT} \phi(E) \cdot \frac{1}{w_E}
\end{aligned}$$

$$
\begin{aligned}
= & \left[\sum_{E \in SST} \mathcal{K}_1 \cdot w_E + \sum_{LST \in \mathcal{L}} \left\{ \sum_{E \in LST} \mathcal{K}_1 \cdot w_E \right\}\right] + \\
& \left[\sum_{E,E' \in SST, E \neq E'} \mu(E,E') \cdot \frac{w_{E'}}{w_E} + \sum_{LST \in \mathcal{L}} \left\{ \sum_{E \in SST, E' \in LST} \mu(E,E') \cdot \frac{w_{E'}}{w_E} \right\} + \right. \\
& \left. \sum_{LST \in \mathcal{L}} \left\{ \sum_{E,E' \in LST, E \neq E'} \mu(E,E') \cdot \frac{w_{E'}}{w_E} \right\}\right] + \\
& \left[\sum_{E,E' \in SST, E \neq E'} \omega(E,E') \cdot \frac{1}{w_E} + \sum_{LST \in \mathcal{L}} \left\{ \sum_{E \in SST, E' \in LST} \omega(E,E') \cdot \frac{1}{w_E} \right\} + \right. \\
& \left. \sum_{LST \in \mathcal{L}} \left\{ \sum_{E,E' \in LST, E \neq E'} \omega(E,E') \cdot \frac{1}{w_E} \right\}\right] + \\
& \left[\sum_{E \in SST} \nu(E) \cdot \frac{1}{w_E} + \sum_{LST \in \mathcal{L}} \left\{ \sum_{E \in LST} \nu(E) \cdot \frac{1}{w_E} \right\}\right] + \\
& \left[\sum_{E \in SST} \phi(E) \cdot \frac{1}{w_E} + \sum_{LST \in \mathcal{L}} \left\{ \sum_{E \in LST} \phi(E) \cdot \frac{1}{w_E} \right\}\right]
\end{aligned}
$$

We define:

$$
\begin{aligned}
t_{SST,1} = & \sum_{E \in SST} \mathcal{K}_1 \cdot w_E + \sum_{E,E' \in SST, E \neq E'} \mu(E,E') \cdot \frac{w_{E'}}{w_E} + \sum_{E,E' \in SST, E \neq E'} \omega(E,E') \cdot \frac{1}{w_E} + \\
& \sum_{LST \in \mathcal{L}} \left\{ \sum_{E \in SST, E' \in LST} \omega(E,E') \cdot \frac{1}{w_E} \right\} + \sum_{E \in SST} \nu(E) \cdot \frac{1}{w_E} + \sum_{E \in SST} \phi(E) \cdot \frac{1}{w_E} \\
t_{SST,2}(LST) = & \sum_{E \in LST} \mathcal{K}_1 \cdot w_E + \sum_{E \in SST, E' \in LST} \mu(E,E') \cdot \frac{w_{E'}}{w_E} + \sum_{E,E' \in LST, E \neq E'} \mu(E,E') \cdot \frac{w_{E'}}{w_E} + \\
& \sum_{E,E' \in LST, E \neq E'} \omega(E,E') \cdot \frac{1}{w_E} + \sum_{E \in LST} \nu(E) \cdot \frac{1}{w_E} + \sum_{E \in LST} \phi(E) \cdot \frac{1}{w_E}
\end{aligned}
$$

We then have:

$$
t(T, R_d, \mathcal{W}) = t_{SST,1} + \sum_{LST \in \mathcal{L}} t_{SST,2}(LST)
$$

where $t_{SST,1}$ is a constant because it depends only on the width assignment of the source subtree $SST$. For any $LST \in \mathcal{L}$, $t_{SST,2}(LST)$ is a function whose only variables are the wire widths of the loading subtree $LST$ (assuming that the wire widths in the $SST$ are given). Since the contribution from each loading subtree $LST$ to the summation is independent of each other, $t(MSIT, R_d, \mathcal{W})$ is optimized if and only if for each $LST \in \mathcal{L}$, $t_{SST,2}(LST)$ is optimized. The separability between $LSTs$ follows as a consequence.

In order to prove the separability within an $LST$, let $\mathcal{B} = \{b_1, b_2, ..., b_n\}$ be the set of subtrees branching off from the path $P$. We define:

$$
\begin{aligned}
\mathcal{H}_{P,1} = & \sum_{E \in P} \mathcal{K}_1 \cdot w_E + \sum_{E \in SST, E' \in P} \mu(E,E') \cdot \frac{w_{E'}}{w_E} + \sum_{E,E' \in P, E \neq E'} \mu(E,E') \cdot \frac{w_{E'}}{w_E} + \\
& \sum_{E,E' \in P, E \neq E'} \omega(E,E') \cdot \frac{1}{w_E} + \sum_{E \in P} \nu(E) \cdot \frac{1}{w_E} + \sum_{E \in P} \phi(E) \cdot \frac{1}{w_E}
\end{aligned}
$$

$$\begin{aligned}\mathcal{H}_{P,2}(b) &= \sum_{E\in b}\mathcal{K}_1\cdot w_E + \sum_{E\in SST+P,E'\in b}\mu(E,E')\cdot\frac{w_{E'}}{w_E} + \sum_{E,E'\in b,E\neq E'}\mu(E,E')\cdot\frac{w_{E'}}{w_E} + \\ &\quad \sum_{E\in P,E'\in b}\omega(E,E')\cdot\frac{1}{w_E} + \sum_{E,E'\in b,E\neq E'}\omega(E,E')\cdot\frac{1}{w_E} + \sum_{E\in b}\nu(E)\cdot\frac{1}{w_E} + \sum_{E\in b}\phi(E)\cdot\frac{1}{w_E}\end{aligned}$$

We then have:

$$t_{SST,2}(LST) = \mathcal{H}_{P,1} + \sum_{b\in\mathcal{B}}\mathcal{H}_{P,2}(b)$$

Because the wire widths in both $SST$ and $P$ are given, $\mathcal{H}_{P,1}$ is a constant which depends only on the wire widths in $SST$ and $P$. For any $b\in\mathcal{B}$, $\mathcal{H}_{P,2}(b)$ is a function whose variables are the widths of the edges in $SST$, $P$ and $b$. Therefore, the contribution of $b$ to the objective function (7) is independent of any other subtree in $\mathcal{B}$. Clearly, the separability within an $LST$ holds. □

## B. LST Monotone Property

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

**Proof**: Based on the definition of the coefficient functions $F$, $G$ and $H$, it is not difficult to show for any particular $LST$:

$$\begin{aligned}F(E_1,E_2) &\geq F(E_1,E_2') && \text{if } E_2,E_2'\in LST,\ E_2\in Right(E_2')\\ F(E_1,E_2) &\geq F(E_1',E_2) && \text{if } E_1,E_1'\in LST,\ E_1\in Left(E_1')\\ G(E_1) &\geq G(E_1') && \text{if } E_1,E_1'\in LST,\ E_1\in Left(E_1')\\ H(E_1) &\geq H(E_1') && \text{if } E_1,E_1'\in LST,\ E_1\in Left(E_1')\\ H(E_1) &= F(E_1,E_2) && \text{if } E_1,E_2\in LST,\ E_2\in Right(E_1)\end{aligned}$$

where $E\in Right(E')$ if $E$ is right to $E'$ and $E\in Left(E')$ if $E$ is left to $E'$.

In this case, the left-right relationship in an $LST$ is the same as the ancestor-descendant relationship in an $SSIT$. If we replace the ancestor-descendant relationship in the proof of the $SSIT$ monotone property in [7] by the left-right relationship, the proof is applicable to the $LST$ monotone property. □

## D. Dominant Property

**Theorem 3** *If a wire width assignment $\mathcal{W}$ dominates $\mathcal{W}^*$ for an MSIT, then any* local refinement *of $\mathcal{W}$ still dominates $\mathcal{W}^*$. Similarly, if a wire width assignment $\mathcal{W}$ is dominated by $\mathcal{W}^*$, then any* local refinement *of $\mathcal{W}$ is dominated by $\mathcal{W}^*$.*

**Proof:** Similar to the proof of the dominance property in [11], we define the following equations for any particular edge $E$:

$$\Psi(MSIT,\mathcal{E},E,\mathcal{W}) = \mathcal{K}_1\cdot\sum_{E'\in MSIT-\{E\}}w_{E'} + \mathcal{K}_2\cdot\sum_{E',E''\in MSIT-\{E\},E'\neq E''}F(E',E'')\cdot\frac{w_{E''}}{w_{E'}} +$$

$$\mathcal{K}_3 \cdot \sum_{E',E'' \in MSIT-\{E\}, E' \neq E''} F(E',E'') \cdot \frac{1}{w_{E'}} \; +$$

$$\mathcal{K}_4 \cdot \sum_{E' \in MSIT-\{E\}} G(E') \cdot \frac{1}{w_{E'}} \; + \; \mathcal{K}_5 \cdot \sum_{E' \in MSIT-\{E\}} H(E') \cdot \frac{1}{w_{E'}} \tag{20}$$

$$\Phi(MSIT,\mathcal{E},E,\mathcal{W}) \; = \; \mathcal{K}_1 \; + \; \mathcal{K}_2 \cdot \sum_{E' \in MSIT-\{E\}} F(E',E) \cdot \frac{1}{w_{E'}} \tag{21}$$

$$\begin{aligned}\Theta(MSIT,\mathcal{E},E,\mathcal{W}) \; &= \; \mathcal{K}_2 \cdot \sum_{E' \in MSIT-\{E\}} F(E,E') \cdot w_{E'} \; + \; \mathcal{K}_3 \cdot \sum_{E' \in MSIT-\{E\}} F(E,E') \\ &+ \; \mathcal{K}_4 \cdot G(E) \; + \; \mathcal{K}_5 \cdot H(E)\end{aligned} \tag{22}$$

We can then rewrite the objective function (7) as follows:

$$t(MSIT,\mathcal{E},\mathcal{W}) \; = \; \Psi(MSIT,\mathcal{E},E,\mathcal{W}) \; + \; \Phi(MSIT,\mathcal{E},E,\mathcal{W}) \cdot w_E \; + \; \Theta(MSIT,\mathcal{E},E,\mathcal{W}) \cdot \frac{1}{w_E} \tag{23}$$

Since (20)–(22) depend only on the wire width assignment for edges other than $E$, for simplicity, we use $\Psi(\mathcal{W},\overline{E}), \Phi(\mathcal{W},\overline{E})$ and $\Theta(\mathcal{W},\overline{E})$ instead of $\Psi(MSIT,\mathcal{E},E,\mathcal{W}), \Phi(MSIT,\mathcal{E},E,\mathcal{W})$ and $\Theta(MSIT,\mathcal{E},E,\mathcal{W})$, respectively.

Because $\Psi(\mathcal{W},\overline{E})$, $\Phi(\mathcal{W},\overline{E})$, and $\Theta(\mathcal{W},\overline{E})$ are independent of $w_E$, they are considered as constants for local refinement of $E$. Minimizing (23) gives the local refinement $\tilde{w}_E$ of segment $E$. Notice that for any pair of wiresizing solutions $\mathcal{W}$ and $\mathcal{W}'$ such that $\mathcal{W}'$ dominates $\mathcal{W}$. We have $\Phi(\mathcal{W},\overline{E}) \geq \Phi(\mathcal{W}',\overline{E})$ and $\Theta(\mathcal{W},\overline{E}) \leq \Theta(\mathcal{W}',\overline{E})$.

Let $w_E^*$ be the width for segment $E$ in the optimal assignment $\mathcal{W}^*$. Since $\tilde{w}_E$ is the local refinement for $E$ with respect to $\mathcal{W}$, we have:

$$\Psi(\mathcal{W},\overline{E}) \; + \; \Phi(\mathcal{W},\overline{E}) \cdot \tilde{w}_E \; + \; \Theta(\mathcal{W},\overline{E}) \cdot \frac{1}{\tilde{w}_E} \; \leq \; \Psi(\mathcal{W},\overline{E}) \; + \; \Phi(\mathcal{W},\overline{E}) \cdot w_E^* \; + \; \Theta(\mathcal{W},\overline{E}) \cdot \frac{1}{w_E^*} \tag{24}$$

Since $w_E^*$ is also the *locally* optimal width assignment for edge $E$ with respect to the rest of the width assignment in $\mathcal{W}^*$, we have:

$$\Psi(\mathcal{W}^*,\overline{E}) \; + \; \Phi(\mathcal{W}^*,\overline{E}) \cdot w_E^* \; + \; \Theta(\mathcal{W}^*,\overline{E}) \cdot \frac{1}{w_E^*} \; \leq \; \Psi(\mathcal{W}^*,\overline{E}) \; + \; \Phi(\mathcal{W}^*,\overline{E}) \cdot \tilde{w}_E \; + \; \Theta(\mathcal{W}^*,\overline{E}) \cdot \frac{1}{\tilde{w}_E} \tag{25}$$

Summing up (24) and (25), we obtain:

$$\begin{aligned}&\left\{\Phi(\mathcal{W},\overline{E}) - \Phi(\mathcal{W}^*,\overline{E})\right\} \cdot \{\tilde{w}_E - w_E^*\} \; + \; \left\{\Theta(\mathcal{W},\overline{E}) - \Theta(\mathcal{W}^*,\overline{E})\right\} \cdot \left\{\frac{1}{\tilde{w}_E} - \frac{1}{w_E^*}\right\} \; \leq \; 0 \\ &\Rightarrow \left\{\Phi(\mathcal{W},\overline{E}) - \Phi(\mathcal{W}^*,\overline{E}) \; + \; \frac{\Theta(\mathcal{W}^*,\overline{E}) - \Theta(\mathcal{W},\overline{E})}{\tilde{w}_E \cdot w_E^*}\right\} \cdot \{\tilde{w}_E - w_E^*\} \; \leq \; 0\end{aligned}$$

If $\mathcal{W}$ dominates $\mathcal{W}^*$, we have $\Phi(\mathcal{W},\overline{E}) \leq \Phi(\mathcal{W}^*,\overline{E})$ and $\Theta(\mathcal{W}^*,\overline{E}) \leq \Theta(\mathcal{W},\overline{E})$, and therefore $\tilde{w}_E - w_E^* \geq 0$.[8] That is, the refinement of $\mathcal{W}$ still dominates $\mathcal{W}^*$. Similarly, if $\mathcal{W}^*$ dominates $\mathcal{W}$, then $\mathcal{W}^*$ dominates the local refinement of $\mathcal{W}$. □

[8] When both $\Phi(\mathcal{W},\overline{E}) = \Phi(\mathcal{W}^*,\overline{E})$ and $\Theta(\mathcal{W}^*,\overline{E}) = \Theta(\mathcal{W},\overline{E})$, the only possible is that $w_{E'} = w_{E'}^*$ for $E' \neq E$ because $\mathcal{W}$ dominates $\mathcal{W}^*$. Thus, according to the definition of local refinement, $\tilde{w}_E = w_E^*$. So we always can say that $\tilde{w}_E - w_E^* \geq 0$.

# References


[1] C. J. Alpert, T. C. Hu, J. H. Huang, and A. B. Kahng, "A Direct Combination of the Prim and Dijkstra Constructions for Improved Performance-Driven Routing", *Proc. IEEE Int'l Symp. on Circuits and Systems*, May 1993, pp. 1869-1872.

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

[3] H. Chan, Private Communication, 1995.

[4] J. Chung, and C. K. Cheng, "Skew Sensitivity Minimization of Buffered Clock Tree", *Proc. IEEE Int'l. Conf. on Computer-Aided Design*, 1994, pp. 280-283.

[5] J. Cong and L. He, "Optimal Wiresizing for Interconnects with Multiple Sources", *Proc. IEEE Int'l. Conf. on Computer Design*, Nov. 1995.

[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, 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.

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

[9] 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.

[10] J. Cong and K. S. Leung, "Optimal Wiresizing Under the Distributed Elmore Delay Model", *Proc. IEEE Int'l. Conf. on Computer-Aided Design*, 1993, pp. 634-639.

[11] 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.

[12] W. Dai, Private Communication, 1992.

[13] J. G. Ecker, "Geometric Programming: Methods, Computations and Applications", *SIAM Review*, Vol. 22, No. 3, July 1980, pp. 338-362.

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

[15] J. P. Fishburn and A. E. Dunlop, "TILOS: A Psoynomial Programming Approach to Transistor Sizing", *Proc. IEEE Int'l. Conf. on Computer-Aided Design*, 1985, pp. 326-328.

[16] L. P. P. P. van Ginneken, "Buffer Placement in Distributed RC-tree Networks for Minimal Elmore Delay", *Proc. ISCAS*, 1990, pp. 865-868.

[17] T. D. Hodes, B. A. McCoy, and G. Robins, "Dynamically-Wiresized Elmore-Based Routing Constructions", *Proc. ISCAS*, 1994, pp. 463-466.

[18] X. Hong, T. Xue, E. S. Kuh, C. K. Cheng, and J. Huang, "Performance-Driven Steiner Tree Algorithms For Global Routing", *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 177-181.

[19] A. B. Kahng and G. Robins, "A New Class of Iterative Steiner Tree Heuristics with Good Performance", *Proc. IEEE Int'l. Conf. on Computer-Aided Design*, July 1992, pp. 893-902.

[20] J. Lillis, C. K. Cheng and T. T. Y. Lin, "Optimal Wire Sizing and Buffer Insertion for Low Power and a Generalized Delay Model", *Proc. IEEE Int'l. Conf. on Computer-Aided Design*, Nov. 1995, pp. 138-143.

[21] B. A. McCoy and G. Robins, "Non-Tree Routing", *Proc. IEEE ISCAS*, 1994, pp. 430-434.

[22] P. K. Sancheti and S. S. Sapatnekar, "Interconnect Design Using Convex Optimization", *Proc. IEEE Custom Integrated Circuits Conference*, 1994, pp. 549-552.

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

[24] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya, and S. M. Kang, "An Exact Solution to the Transistor Sizing Problem for CMOS Circuits Using Convex Optimization", *IEEE Tran. on CAD*, November 1993, pp. 1621-1634.

[25] *MCNC Designers' Manual*, MCNC.

[26] 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.

[27] N. Menezes, R. Baldick, and L. T. Pilegi, "A Sequential Quadratic Programming Approach to Concurrent Gate and Wire Sizing", *Proc. ACM/IEEE ICCAD*, 1995, pp. 144-151.

[28] S. Pullela, N. Menezes, and L. T. Pillage, "Reliable Non-Zero Skew Clock Trees Using Wire Width Optimization", *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 165-170.

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

[30] T. Xue and E. S. Kuh, "Post Routing Performance Optimization via Multi-Link Insertion and Non-Uniform Wiresizing", *Proc. IEEE Int'l Conf. on Computer-Aided Design*, 1995, pp. 575-580.

[31] Q. Zhu, W. W. M. Dai, and J. G. Xi, "Optimal Sizing of High-Speed Clock Networks Based on Distributed RC and Lossy Transmission Line Models", *Proc. IEEE Int'l. Conf. on Computer-Aided Design*, 1993, pp. 628-633.