

Interconnect Sizing and Spacing with Consideration of Coupling Capacitance *

Jason Cong, Lei He, Cheng-Kok Koh[†], and David Zhigang Pan
Department of Computer Science
University of California, Los Angeles, CA 90095

[†]Department of Electrical and Computer Engineering
Purdue University, West Lafayette, Indiana 47907

Abstract

This paper studies *interconnect sizing and spacing* (ISS) problem with consideration of coupling capacitance in addition to area and fringing capacitances, for performance optimization of a single critical net (denoted as SISS) and for global optimization of multiple critical nets (denoted as GISS). We first introduce the formulation of *symmetric* and *asymmetric* wire sizing and spacing. We reveal several important properties for optimal ISS solutions, namely *effective-fringing property* and *extended dominance property* for a single net, and the more general *extended dominance property* for multiple nets, under several interconnect resistance and capacitance models. Based on these properties, we develop efficient algorithms to compute lower and upper bounds of optimal ISS solutions, using *local refinement* operation for SISS and *bound refinement* operation for GISS, respectively. Our experiments show that these bound computation algorithms are very effective, i.e., the resulting lower and upper bounds for each wire segment are identical or very close. When the lower and upper bounds are not identical after bound computation, we then use a bottom-up dynamic programming algorithm to compute final ISS solutions. Very promising experimental results are shown.

* This work is partially supported by the NSF Young Investigator Award MIP-9357582 and a grant from Intel Corporation. This research makes use of the software donated from Avant! Corporation, whose generous donation is greatly appreciated.

1 Introduction

It is well known that for deep sub-micron (DSM) VLSI designs, interconnect has become the dominating factor in determining the overall circuit performance [1]. Among various interconnect optimization techniques, wire sizing is the one that determines proper width for each wire segment in a routing structure.

It was first shown by [2, 3] that when wire resistance becomes significant, as in DSM designs, proper wire sizing can effectively reduce the interconnect delay. In their study, [2] used an upper-bound-of-Elmore delay model while [3] used the Elmore delay model directly, and both minimized the sum of weighted delays from source to all timing-critical sinks. Assuming that each wire segment has a set of *discrete* widths, their work presented the first optimal wire sizing algorithm for a single-source RC interconnect tree. The algorithm is based on a local refinement operation, so it is called *greedy wire sizing algorithm* (GWSA). It was later on extended for a routing tree with multiple sources and without *a priori* wire segmentation [4], and for the maximum delay objective using Lagrangian relaxation [5]. An alternative approach to GWSA is through bottom-up dynamic programming [6] and it can be combined easily with routing tree construction and buffer insertion [7]. Another formulation of wire sizing optimization is to determine the *continuous* wire shaping functions. The closed-form wire shaping functions were derived to minimize the Elmore delay, first without fringing capacitance [8, 9], later on with fringing capacitance [10, 11], and recently for a bi-directional (i.e., multiple-source) wire [12]. Wire sizing optimization is also studied using high-order delay model such as [13, 14]. Comprehensive survey and tutorial of these optimization techniques can be found in [15, 1].

However, most of these studies only sized a *single* net and did not explicitly consider the coupling capacitance between adjacent nets. As VLSI technology advanced to DSM feature size, the coupling capacitance between adjacent wires indeed has become the dominating component in the total interconnect capacitance, since tall and narrow wires are placed closer to each other. The coupling capacitance will lead to excessive signal delay and crosstalk noise which, if not optimized, will affect circuit performance and even cause circuit malfunction. In this paper, we focus on the effect of coupling capacitance on interconnect delay. We refer the reader to recent papers in [16, 17, 18, 19, 20, 21] for noise control and minimization.

In order to give a quantitative depiction of the importance of coupling capacitance on delay, we conducted a set of experiments using a $0.35\mu m$ industrial technology on certain netlists extracted from an advanced microprocessor chip. We first obtained routing tree and wire sizing solutions by the simultaneous topology

construction and wire sizing algorithm [7] with consideration of only area and fringing capacitances. We then ran HSPICE simulations on these routing solutions under different spacing assumptions with consideration of coupling capacitance. Table 1 shows the maximum delays of the same routing solution under four different scenarios: the infinite (∞) spacing means that no neighboring wire is present; the $0.5\mu m$ for layer M1 and $1.5\mu m$ for layer M5 mean that neighboring wires have fixed spacing of $0.5\mu m$ or $1.5\mu m$ away from these nets; *random1* and *random2* mean that neighboring wires have random spacings (from $0.32\mu m$ to $2.72\mu m$ for layer M1, and from $1.28\mu m$ to $4.48\mu m$ for layer M5, with step to be $0.2\mu m$) away from these nets. The coupling capacitance is determined according to these spacings, except that there is no coupling capacitance in the infinite spacing case. From the table, the delay increases drastically when the coupling capacitance is taken into account. For example, the maximum delay for Net5 on layer M1 at spacing *random1* is increased by 38.7% when compared with that at the infinite spacing. Therefore, it is unlikely that an optimal wire sizing solution which considers only area and fringing capacitances would remain optimal when coupling capacitance is taken into account.

spacing	All wires in M1 layer				All wires in M5 layer			
	∞	$0.5\mu m$	<i>random1</i>	<i>random2</i>	∞	$1.5\mu m$	<i>random1</i>	<i>random2</i>
Net1	0.1356	0.3750	0.2386	0.2363	0.1617	0.2890	0.2496	0.2017
Net2	0.2542	0.7143	0.4494	0.3930	0.2713	0.4711	0.3903	0.3544
Net3	0.7376	1.9168	0.8964	0.9613	0.5870	1.0029	0.7927	0.7502
Net4	0.8851	2.2896	1.3343	1.4783	0.5895	1.0071	0.8417	0.8038
Net5	2.7637	5.7931	3.7653	3.4119	1.7838	3.6393	2.8732	2.7871

Table 1: Maximum delays (ns) for 5 industrial nets with neighboring wires at different spacings (the driver resistance is 270Ω)

In this paper, we study the *interconnect sizing and spacing* (ISS) problem with explicit consideration of coupling capacitance, for performance optimization of a single or multiple critical nets (for optimization of single net, it is denoted as SISS; for global optimization of multiple nets, it is denoted as GISS). This paper is an extensive outgrowth of our technical digest presented in ICCAD'97 [22]. In [22], we developed a bottom-up dynamic programming algorithm for SISS, and we revealed the *effective-fringing property* (EFP) and developed a dynamic-programming bound computation algorithm based on EFP for GISS. This bound computation algorithm for GISS, however, did not take into account of different net weights, thus may fail to get the optimal GISS solution in some weighted cases. Such a problem is fixed in this paper (to be discussed in more detail in Section 4). In addition, we introduce two new theorems, named *extended dominance property for a single net*, and *extended dominance property for multiple nets*. They lead to very efficient wire width and spacing

bound computation algorithms, namely *local refinement* (LR) algorithm for SISS, and *bound refinement* (BR) algorithm for GISS, respectively. It shall be noted that in ICCAD'97, there is another work [23] on wire sizing and spacing, which used the *dominant time constant* as a measure of signal propagation delay and formulated the problem into a semidefinite programming. The dominant time constant is an approximation of the maximum delay of all sinks. However, it cannot measure the delays to other critical sinks in the same or other critical nets. In our study, we use the Elmore delay model to guide wire sizing and spacing optimization to minimize delays to all timing-critical sinks. Also, [23] used a rather simple capacitance model where the ground capacitance of a wire is proportional to its width and the coupling capacitance of a wire to its neighboring wire is inversely proportional to their spacing. In this study, we use a much more accurate 2.5D capacitance model than [23]. In [36], the GISS problem was also studied, but by a different approach. It was shown that the GISS problem can be formulated as a *bounded CH-program*, thus the *extended local refinement* (ELR) operation can be used for bound computation. The ELR essentially uses some minimum and maximum resistance/capacitance coefficients conservatively to guide bound computation. It is usually less effective than BR operation since BR directly works on the current lower and upper bound widths. Also, the mathematical complexity in the *bounded CH-program* formulation tends to hide the optimizational intuition and complicate the implementation. As to be shown in Section 5, our approach in this paper provides much better results for delay, wiring area, as well as running time.

The organization for the rest of this paper is as follows. Section 2 presents the problem formulation for *symmetric* and *asymmetric* interconnect sizing and spacing for both single and multiple nets, and introduces the interconnect resistance and capacitance models to be used in this work. Section 3 reveals two important properties (*effective-fringing property* and *extended dominance property*), and develops efficient algorithms for symmetric and asymmetric SISS optimization. Section 4 reveals the *extended dominance property for multiple nets*, from which we obtain efficient algorithm for GISS optimization for *all* nets under consideration, not just *one* net as in the SISS formulation. Experimental results in Section 5 show substantial improvement of our SISS and GISS algorithms over previous wire sizing/spacing algorithms. The conclusion follows in Section 6. The proofs of main theorems are presented in Appendix at the end of this paper.

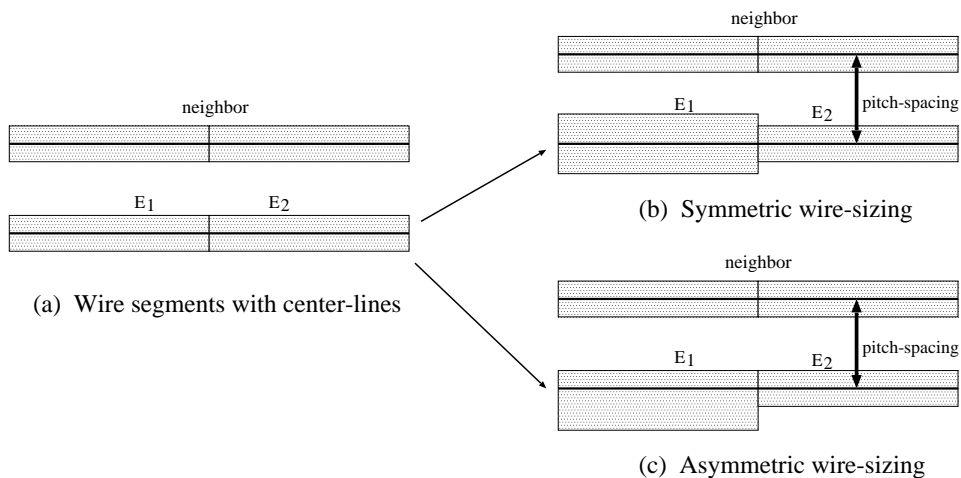


Figure 1: (a) Wire segments with center-lines. (b) Symmetric wire sizing. (c) Asymmetric wire sizing.

2 Problem Formulation

2.1 Symmetric and Asymmetric Wire Sizing

Given a layout of n nets, denoted as \mathcal{N}_i for $i = 1 \dots n$. For each net \mathcal{N}_i , it consists of $n_i + 1$ terminals $\{s_0^i, \dots, s_{n_i}^i\}$ connected by a routing tree, denoted T_i . The terminal s_0^i denotes the source of \mathcal{N}_i , with effective driving resistance of R_i . The rest of the terminals are sinks. The terminals (source and sinks) of T_i are at fixed locations, and T_i consists of m_i wire segments denoted by $\{E_1^i, \dots, E_{m_i}^i\}$. The *center-line* for a wire segment E is defined to be a line that divides E in the initial layout evenly. In Figure 1(a), for example, two horizontal wire segments E_1 and E_2 are shown with their center-lines. We assume that the center-line for each wire segment is fixed during wire sizing and spacing optimization. We call the distance between adjacent center-lines as the *pitch-spacing*.

For wire sizing and spacing optimization, each wire segment has a set of discrete choices of wire widths $\{W_1 = W_{min}, W_2, \dots, W_r\}$. We use w_E to denote the width of the wire segment E . All previous works implicitly assumed *symmetric wire sizing* around the center-lines, so each wire segment needs only one width to describe it. An example of symmetric wire sizing of the two wire segments E_1 and E_2 with a neighboring net is shown in Figure 1(b).

However, symmetric wire sizing may be too restrictive when taking coupling capacitance into account. We thus propose an *asymmetric wire sizing* scheme in which one may size around the center-lines of initial wire

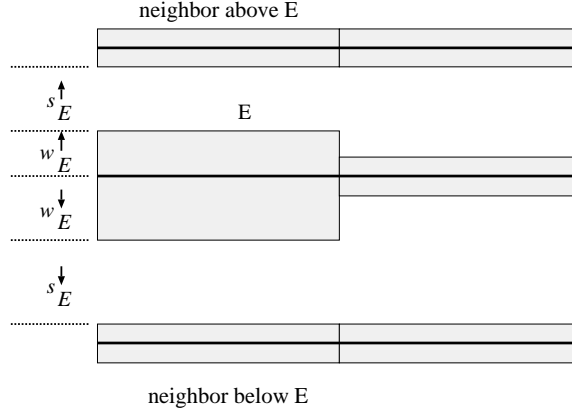


Figure 2: Illustration of two-piece wire widths and edge-to-edge wire spacings for a wire segment E .

segment asymmetrically. In this case, each wire segment needs two pieces of wire widths to describe it. Using the same example as in Figure 1(b), we would like E_1 to be farther away from its upper neighbor. As a result, we increase only the bottom half piece of the wire segment, keeping the top half intact, as shown in Figure 1(c). Let w_E^\downarrow (w_E^\uparrow) be wire width below (above) a horizontal line segment, and s_E^\downarrow (s_E^\uparrow) be the edge-to-edge spacing from E to the neighbor wire segment below (above) it, as illustrated in Figure 2. The new lumped wire width for wire segment E is then the sum $w_E = w_E^\downarrow + w_E^\uparrow$. An asymmetric wire sizing solution is valid if $w_E^\downarrow \geq W_{min}/2$ and $w_E^\uparrow \geq W_{min}/2$. Note that for symmetric wire sizing, $w_E^\downarrow = w_E^\uparrow = w_E/2$. To avoid introducing additional notations, we also use w_E^\downarrow (w_E^\uparrow) to denote the wire width for the left (right) part of a vertical wire segment, and s_E^\downarrow (s_E^\uparrow) to denote the spacing from E to its left (right) neighboring wire, respectively.

2.2 Single-net Interconnect Sizing and Spacing (SISS)

Given a layout of n routing trees T_i 's, the interconnect sizing and spacing problem for a single net is to find a symmetric wire assignment $\mathcal{W} = \{w_{E_1^i}, \dots, w_{E_{m_i}^i}\}$ or an asymmetric wire assignment $\mathcal{W} = \{w_{E_1^i} = (w_{E_1^i}^\downarrow, w_{E_1^i}^\uparrow), \dots, w_{E_{m_i}^i} = (w_{E_{m_i}^i}^\downarrow, w_{E_{m_i}^i}^\uparrow)\}$ for a routing tree of interest, say T_i , in order to optimize the following weighted delay objective with consideration of *the area*, *fringing* and *coupling* capacitances:

$$t_{T_i}(\mathcal{W}) = \sum_{j=1}^{n_i} \lambda_j^i \cdot t_{T_i}(s_j^i, \mathcal{W}) \quad (1)$$

where λ_j^i is the criticality of sink s_j^i in net \mathcal{N}_i , and $t_{T_i}(s_j^i, \mathcal{W})$ is the delay from source s_0^i to sink s_j^i in the routing tree T_i with wire sizing solution \mathcal{W} . Note that although only T_i appears in the above delay formula, other nets

provide neighborhood structure and constraints for T_i during the optimization of T_i , and thus affect the value of $t_{T_i}(\mathcal{W})$ as well.

We model the routing tree by an RC circuits and use the distributed Elmore delay model [24, 25] to measure the interconnect delays for performance optimization, similar to [26, 27, 4]. For clarity of presentation, we assume that a uniform grid structure is superimposed on the routing plane, and each wire segment in the routing plane is divided into a sequence of wires of unit length, defined to be the *grid edges*¹. Nonetheless, the results presented in this paper can be easily extended to cases where the wire segments are of non-uniform lengths in the same way as in [26].

Given a grid edge E , we use r_E for the wire resistance of E , c_E for the wire capacitance (including the area, fringing and coupling capacitances) of E , $Des(E)$ for the set of grid edges in the subtree rooted at E (excluding E), and $Ans(E)$ for the set of grid edges $\{E' | E \in Des(E')\}$ (again, excluding E). We will present the wire resistance and capacitance models in Section 2.4. In addition, we use $sink(E)$ to denote the set of sinks in the subtree rooted at E , and $C_{down}(E)$ to denote the *total* downstream capacitance in the subtree rooted at E (including both the wire capacitances and the sink capacitances):

$$C_{down}(E) = \sum_{s_j^i \in sink(E)} c_{s_j^i} + \sum_{E' \in Des(E)} c_{E'}$$

Furthermore, $sink(T_i)$ denotes the set of sinks in net \mathcal{N}_i , $P_{T_i}(u, v)$ denotes the unique path from u to v for any grid points u, v in T_i , and R_i denotes the driver resistance of the routing tree T_i under optimization. The distributed Elmore delay from source s_0^i to a certain sink s_j^i is then given by:

$$t_{T_i}(s_j^i, \mathcal{W}) = R_i \cdot \left(\sum_{E \in T_i} c_E + \sum_{s_j^i \in sink(T_i)} c_{s_j^i} \right) + \sum_{E \in P_{T_i}(s_0^i, s_j^i)} r_E \cdot \left(\frac{c_E}{2} + C_{down}(E) \right). \quad (2)$$

Let $\lambda(s_0^i) = \sum_{s_k^i \in sink(T_i)} \lambda_k^i$ and $\lambda(E) = \sum_{s_k^i \in sink(E)} \lambda_k^i$, then the performance measure $t_{T_i}(\mathcal{W})$ of T_i in (1) can be rewritten as:

$$\begin{aligned} t_{T_i}(\mathcal{W}) &= \sum_{s_k^i \in sink(T_i)} \lambda_k^i \cdot t_{T_i}(s_k^i, \mathcal{W}) \\ &= \lambda(s_0^i) \cdot R_i \cdot \left(\sum_{E \in T_i} c_E + \sum_{s_k^i \in sink(T_i)} c_{s_k^i} \right) + \sum_{E \in T_i} \lambda(E) \cdot r_E \cdot \left\{ \frac{c_E}{2} + C_{down}(E) \right\}. \end{aligned} \quad (3)$$

Note that if we treat $\lambda(s_0^i)R_i$ as the effective driver resistance, and $\lambda(E)r_E$ as the effective wire resistance of edge E , then Eqn. (3) is exactly of the same form of Eqn. (2). In this paper, we focus on the objective of

¹The “edge” or “wire segment” in the rest of the paper refers to the “grid edge”, unless specified.

minimizing the weighted sum of sink delays as in [26]. A previous work [5] showed that by assigning appropriate criticality/weight of each sink based on Lagrangian relaxation, the weighted-sum formulation can be used iteratively to minimize the maximum delay.

2.3 Global Interconnect Sizing and Spacing (GISS) for Multiple Nets

In the global interconnect sizing and spacing problem for multiple nets, we again assume that an initial layout of n routing trees T_i 's is given. However, the wire widths of all nets may be changed. With consideration of the area, fringing and coupling capacitances, the GISS problem for multiple nets is to find a symmetric wire assignment $\mathcal{W} = \{w_{E_1^1}, \dots, w_{E_{m_1}^1}, \dots, w_{E_1^n}, \dots, w_{E_{m_n}^n}\}$ or an asymmetric wire assignment

$$\mathcal{W} = \{w_{E_1^1} = (w_{E_1^1}^\downarrow, w_{E_1^1}^\uparrow), \dots, w_{E_{m_1}^1} = (w_{E_{m_1}^1}^\downarrow, w_{E_{m_1}^1}^\uparrow), \dots, \\ w_{E_1^n} = (w_{E_1^n}^\downarrow, w_{E_1^n}^\uparrow), \dots, w_{E_{m_n}^n} = (w_{E_{m_n}^n}^\downarrow, w_{E_{m_n}^n}^\uparrow)\}$$

for all T_i 's such that, the summation of the weighted performance measure of all n nets, i.e.,

$$t(\mathcal{W}) = \sum_{i=1}^n \delta_i t_{T_i}(\mathcal{W}) \quad (4)$$

is minimized, where δ_i indicates the criticality of net i .

2.4 Interconnect Resistance and Capacitance Models

Before presenting our SISS and GISS solutions, we introduce in this subsection a series of interconnect resistance and capacitance models, based on which we develop theoretical foundations and algorithms. These models span from simple ones (like those widely used in previous wire sizing studies in [3, 9]) to very general ones (e.g., implicit formula based on a look-up-table for DSM technologies).

For interconnect resistance, it has been a common approach (e.g., [28, 26, 27, 4, 9, 22]) to compute it from a constant sheet resistance r . For a grid edge E with wire width w_E , its resistance $r(w_E) = \frac{r}{w_E}$.

Definition 1 Simple Resistance Model (SRM): *Under the SRM, the wire resistance is inversely proportional to its width, i.e., $r(w_E) = \frac{r}{w_E}$, where r is the sheet resistance (some constant depending on technology).*

While SRM for VLSI interconnect is correct to the first order, it may not be accurate enough for DSM designs. For DSM designs, one may choose to use more accurate resistance formula or even very accurate table-look-up. Therefore, we introduce the following more general resistance model:

Definition 2 Monotone Resistance Model (MRM): *Under the MRM, the wire resistance monotonically decreases as wire width increases, i.e., $r(w'_E) < r(w_E)$ if $w_E > w'_E$.*

It is obvious that the SRM is a special case of MRM, which usually holds for interconnect resistance.

For the capacitance model, the table-based 2.5D model presented in [29] is used in our study, where the lumped capacitance for a wire contains the following components: area and fringing capacitances, lateral coupling capacitance, and cross-over and cross-under capacitances. This model was shown to be accurate and efficient for layout optimization. For ease of presentation, we consider only area, fringing and lateral coupling capacitances, which are the major parts of the lumped capacitance. That is, we use a 2D capacitance model simplified from the original 2.5D model in [29]. It shall be noted, however, that our algorithms developed in this paper can incorporate the cross-over and cross-under capacitances easily. We first use a 3D field solver Fastcap [30] to build tables for area, fringing and lateral coupling capacitances under different width and spacing combinations. During layout optimization, we compute area, fringing and coupling capacitances based on these pre-built tables. Details and justification of this method can be found in [29].

We denote the unit area capacitance c_a , the unit length fringing capacitance c_f , and the unit length coupling capacitance c_x . Then

$$c_E = c_a \cdot w_E + c_f + c_x.$$

where c_a , c_f and c_x , in general, can be computed based on three variables $(w_E, s_E^\downarrow, s_E^\uparrow)$ using table-lookup and interpolation/extrapolation.

Definition 3 Simple Capacitance Model (SCM): *Under the SCM, the unit area capacitance c_a and unit length fringing capacitances c_f are constants, and the coupling capacitance monotonically decreases as the edge-to-edge spacing increases (or equivalently, wire width decreases under a fixed pitch-spacing).*

Most previous works on interconnect sizing assumed fixed unit area and fringing capacitance coefficients. They either did not consider coupling capacitance explicitly (such as [26, 4]) or used very simple coupling capacitance model (e.g., c_x is inversely proportional to spacing as in [23]). These models all belong to our Simple Capacitance Model.

To cope with coupling capacitance, we define for each grid edge E , the *effective-fringing capacitance coefficient* $c_{ef}(E) = c_f + c_x$. In practice, effective-fringing capacitance often satisfies the following property:

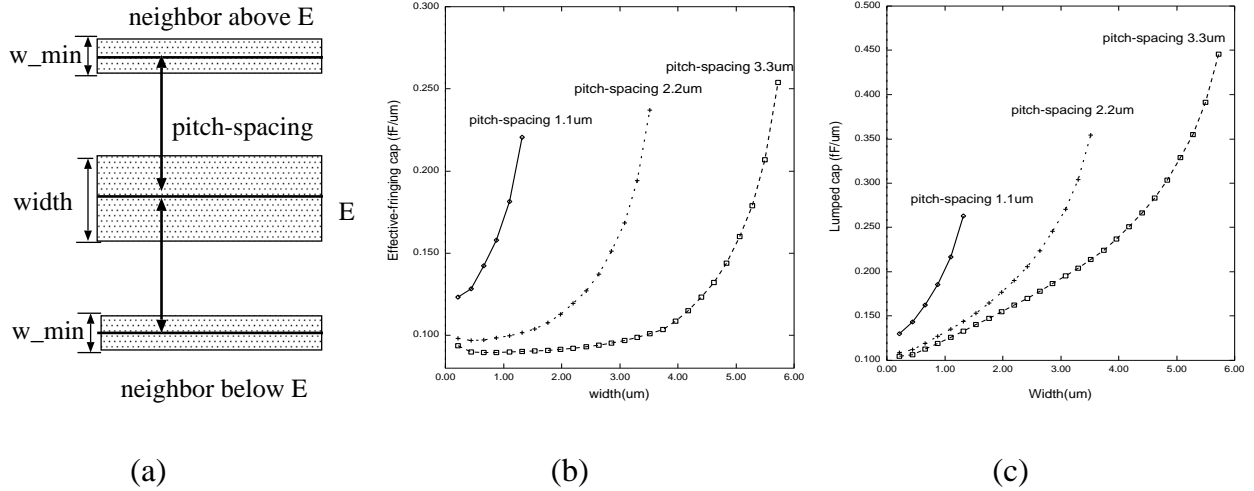


Figure 3: (a) A wire segment E with symmetric width and two neighboring wire segment with certain pitch-spacings. (b) Unit-length effective-fringing capacitance for wire segment E under different width and pitch-spacing scenarios. MEFCM holds for almost all width and pitch-spacing combinations. (c) Similar to (b), but shown is the unit-length lumped capacitance. MLCM always holds for all width and pitch-spacing combinations.

Definition 4 Monotone Effective-Fringing Capacitance Model (MEFCM): Under the MEFCM, the unit area capacitance c_a is constant, and the effective-fringing capacitance coefficient monotonically increases as wire width increases², i.e., $c_{ef}(w'_E) > c_{ef}(w_E)$ if $w'_E > w_E$, given fixed neighboring structures.

The unit length effective-fringing capacitance under a few center-line pitch-spacings in a $0.18\mu\text{m}$ technology [31] is shown in Figure 3 (b) using the geometric pattern of Figure 3 (a). The capacitance values are obtained from the 2D capacitance model. It shows that MEFCM holds for almost all wire width and pitch-spacing combinations. The only exceptions happen at the minimum wire width ($0.22\mu\text{m}$) under large pitch-spacings ($2.2\mu\text{m}$ and $3.3\mu\text{m}$).

Definition 5 Monotone Lumped Capacitance Model (MLCM): Under the MLCM, the lumped capacitance for a wire segment E monotonically increases as wire width increases, i.e., $c(w'_E) > c(w_E)$ if $w'_E > w_E$, given fixed neighboring structures.

It is obvious that SCM is a special case of MEFCM, which by itself is a special case of MLCM. Our study shows that MLCM always holds for interconnect capacitance. As an example, Figure 3 (c) shows the unit

²Or equivalently, wire spacing to neighboring wire decreases since center-lines are fixed in our problem formulation.

length lumped capacitance for wire segment E in Figure 3 (a) under different wire width and pitch-spacing scenarios. It clearly shows that MLCM holds for all wire width and pitch-spacing combinations. As we shall see in Section 3, MEFCM is the capacitance condition to prove the effective-fringing property, and MLCM is the capacitance condition for the extended dominance properties for SISS and GISS.

3 Single-net Interconnect Sizing and Spacing (SISS)

In this section, we study interconnect sizing and spacing for a single critical net, i.e., the wire widths of all other nets are fixed. We first establish two important properties, namely *effective-fringing property* and *extended dominance property* for the SISS problem. Then we derive efficient algorithms for computing the optimal SISS solution.

3.1 Two Properties for SISS Problem

3.1.1 Effective Fringing Property

To tackle with the variable fringing and coupling capacitances as the width for a certain wire segment changes, we lump fringing and coupling capacitances into the effective-fringing (EF) capacitance, as in Section 2.4. Given a *fixed* effective-fringing capacitance coefficient for each edge³ of a net under optimization, we can compute the optimal wire sizing solution for this net, denoted as the OWS-EF solution⁴. The first property, so-called *effective-fringing property*, illustrates the dominance relationship between two OWS-EF solutions for the same net under two different effective fringing capacitance sets. The *dominance relation* between two OWS-EF solutions of a routing tree is defined in the same way as in [26]:

Definition 6 Dominance Relation: *Given two wire width assignments \mathcal{W} and \mathcal{W}' , let w_E be the width assignment of edge E in \mathcal{W} and w'_E be the wire width of E in \mathcal{W}' . Then, \mathcal{W} dominates \mathcal{W}' ($\mathcal{W} \geq \mathcal{W}'$) if and only if for any segment E , $w_E \geq w'_E$.*

Let \mathcal{C}_{ef} denote a set of effective-fringing capacitance coefficients $c_{ef}(E)$'s for all edges in T . We say $\mathcal{C}_{ef} \geq \mathcal{C}'_{ef}$ if $c_{ef}(E) \geq c'_{ef}(E)$ for every E in T . The effective-fringing property can be stated as follows:

³The effective-fringing capacitance coefficients for different edges may be different due to different wire widths and neighboring structures.

⁴OWS-EF problem may have variable effective-fringing capacitance coefficients for different edges in a routing tree. Thus, it is a more general case than the original OWS problem in [26] with only constant c_f and without coupling capacitance.

Theorem 1 Effective-Fringing Property (EFP): Under SRM and MEFCM, let $\hat{W}(c_a, C_{ef})$ be an OWS-EF solution for routing tree T with a set of effective-fringing capacitance coefficients $C_{ef} = \{c_{ef}(E) \mid E \in T\}$, and $\hat{W}(c_a, C'_{ef})$ be an OWS-EF solution for the same tree but with a different set of $C'_{ef} = \{c'_{ef}(E) \mid E \in T\}$. Then there exist a pair of optimal solutions $\hat{W}(c_a, C_{ef})$ and $\hat{W}(c_a, C'_{ef})$ such that $\hat{W}(c_a, C_{ef})$ dominates $\hat{W}(c_a, C'_{ef})$, if $C_{ef} \geq C'_{ef}$.

This theorem was first introduced in our ICCAD'97 technical digest [22]. Its proof is in Appendix A. Intuitively, EFP basically states that larger effective-fringing capacitance will lead to larger optimal wire sizing solution for a net.

3.1.2 Extended Dominance Property for a Single Net

The EFP reveals the relationship of two OWS-EF solutions for an *entire* net. The extended dominance property, instead, reveals the relationship for just *one wire segment* of a net. To introduce this property, we first define the local refinement (LR) operation. LR sizes a wire greedily for local optimum, introduced first in [26]. For single net optimization, the objective under LR is (3).

Definition 7 Local Refinement: Given a wire sizing solution \mathcal{W} , the local refinement of a particular wire segment E is to size E optimally while keeping other sizes in \mathcal{W} intact.

Then we are able to show the dominance relationship of two SISS solutions after performing the local refinement operations by the following theorem.

Theorem 2 Extended Dominance Property (EDP) for a Single Net: Let \mathcal{W}^* be an optimal SISS solution for a routing tree T . Then, under MRM and MLCM models, if a wire sizing solution \mathcal{W}' dominates \mathcal{W}^* , a local refinement of \mathcal{W}' will still dominate \mathcal{W}^* ; similarly, if \mathcal{W}' is dominated by \mathcal{W}^* , a local refinement of \mathcal{W}' will still be dominated \mathcal{W}^* .

This is a new theorem, not in [22]. Its proof can be found in Appendix B. The effective fringing property illustrates the dominance relationship between two optimal sizing solutions of an *entire net*, while the extended dominance property just focuses on *one wire segment* of a net. In this sense, EFP is a stronger result than EDP. However, EFP is more restrictive than EDP in terms of the resistance and capacitance conditions under which it holds. EFP is proved under the Simple Resistance Model and Monotone Effective-Fringing Capacitance

Model, while EDP is valid under the much more general Monotone Resistance Model and Monotone Lumped Capacitance Model.

3.2 Algorithm for SISS Optimization

3.2.1 Bound Computation for SISS

Both EFP and EDP suggest that one may start from an upper bound or a lower bound of the optimal SISS solution, and iteratively update these bounds toward the final optimal SISS solution.

EFP reveals the important relationship between the optimal SISS solution and a sizing solution that dominates (or is dominated by) it. Suppose \mathcal{W}^* be the optimal SISS solution to the net under optimization. Let $\mathcal{S}^{\downarrow*}$ and $\mathcal{S}^{\uparrow*}$ denote the spacings obtained based on \mathcal{W}^* , and $c_{ef}(w_E^*, s_E^{\downarrow*}, s_E^{\uparrow*})$ be the effective-fringing capacitance for each edge E based on \mathcal{W}^* , $\mathcal{S}^{\downarrow*}$, and $\mathcal{S}^{\uparrow*}$. Let \mathcal{W}^U be an upper bound of \mathcal{W}^* , then corresponding $\mathcal{S}^{\downarrow L}$ and $\mathcal{S}^{\uparrow L}$ would be lower bounds of $\mathcal{S}^{\downarrow*}$ and $\mathcal{S}^{\uparrow*}$, respectively. Under the Monotone Effective-Fringing Capacitance Model, we have $c_{ef}(w_E^U, s_E^{\downarrow L}, s_E^{\uparrow L}) \geq c_{ef}(w_E^*, s_E^{\downarrow*}, s_E^{\uparrow*})$. Now consider two OWS-EF instances with different effective-fringing capacitances. In the first instance, the effective-fringing capacitance for each edge E is $c_{ef}(w_E^U, s_E^{\downarrow L}, s_E^{\uparrow L})$. In the second instance, the effective-fringing capacitance for each edge E is $c'_{ef}(w_E^*, s_E^{\downarrow*}, s_E^{\uparrow*})$. Clearly, $\mathcal{C}_{ef} \geq \mathcal{C}'_{ef}$. From the effective-fringing property, the OWS-EF solution under \mathcal{C}_{ef} , $\hat{\mathcal{W}}(c_a, \mathcal{C}_{ef})$ would dominate the OWS-EF solution under \mathcal{C}'_{ef} , i.e., $\hat{\mathcal{W}}(c_a, \mathcal{C}'_{ef})$. Since $\hat{\mathcal{W}}(c_a, \mathcal{C}'_{ef}) = \mathcal{W}^*$, then $\hat{\mathcal{W}}(c_a, \mathcal{C}_{ef})$ would still be an upper bound of the optimal SISS solution \mathcal{W}^* . We can set $\hat{\mathcal{W}}(c_a, \mathcal{C}_{ef})$ to be the new upper bound \mathcal{W}^U . The above procedure can be applied to update wire width upper bounds iteratively until there is no improvement. Similarly, from an initial wire width lower bound \mathcal{W}^L of \mathcal{W}^* , we can also iteratively update the lower bound widths.

It shall be pointed that obtaining an OWS-EF solution by itself is not totally trivial. We can use the bottom-up dynamic programming algorithm (to be presented in Section 3.2.2) to obtain an exact OWS-EF solution. Or we may just compute the tight upper and lower bounds of the exact OWS-EF solution by local refinement operations in a similar manner of [26], since the real purpose of an OWS-EF solution is still to provide an upper or lower bound for the final optimal SISS solution.

In practice, the local refinement approach is usually much more efficient than the dynamic programming approach (see Section 5). Indeed, the LR approach can be directly justified from EDP which is valid under much more general resistance and capacitance models, i.e., MRM and MLCM. Let \mathcal{W}^* be the optimal SISS

solution, and $c(w_E^*, s_E^{\downarrow*}, s_E^{\uparrow*})$ be the lumped capacitance for an edge E based on \mathcal{W}^* , $\mathcal{S}^{\downarrow*}$, and $\mathcal{S}^{\uparrow*}$. Now consider an upper bound \mathcal{W}^U of \mathcal{W}^* . From \mathcal{W}^U , we can obtain the corresponding spacing lower bounds, $\mathcal{S}^{\downarrow L}$ and $\mathcal{S}^{\uparrow L}$ for $\mathcal{S}^{\downarrow*}$ and $\mathcal{S}^{\uparrow*}$, respectively. Under the capacitance model MLCM, we have $c(w_E^U, s_E^{\downarrow L}, s_E^{\uparrow L}) \geq c(w_E^*, s_E^{\downarrow*}, s_E^{\uparrow*})$. According to the extended dominance property, a local refinement operation to any wire segment of \mathcal{W}^U will still be an upper bound of \mathcal{W}^* . Similarly, the local refinement operation of an initial lower bound \mathcal{W}^L will still be a lower bound of \mathcal{W}^* .

Our local refinement (LR) based bound computation algorithm, named SISS-LR, is outlined in Figure 4. The LR operation can be performed analytically if closed-form capacitance formula of c_a and c_{ef} are given, in a manner similar to [26]; otherwise, we may need an enumeration of available wire widths between lower and upper bounds if capacitance tables are given, as in this work (see details in Section 5). Our experimental results (to be presented in Section 5) show that using the bound computation procedure SISS-LR alone, most wire segments already become *convergent* (i.e., with the same lower and upper bound width). For those wire segments that are not convergent, we will then use a bottom-up dynamic programming algorithm (to be presented in Section 3.2.2) to find the final optimal SISS solution within lower and upper bounds.

Note that for the symmetric case, each wire segment just has one width, while for the asymmetric case, it has two widths, one for each side of its center-line. For the asymmetric SISS, we can compute upper and lower bounds for the lumped width of each edge $w_E = w_E^{\downarrow} + w_E^{\uparrow}$ as in the symmetric SISS. Then, an upper bound for w_E^{\downarrow} can be computed by $w_E^{\downarrow U} = w_E^U - w_E^{\uparrow L}$. A lower bound for w_E^{\downarrow} can be computed by $w_E^{\downarrow L} = \max(W_{min}/2, w_E^L - w_E^{\uparrow U})$. We can obtain the lower and upper bounds for w_E^{\uparrow} in the same way. This idea is used to prove the *asymmetric effective-fringing property* in [32], and also used for the *conservative embedding* in [33]. Such a bound computation guarantees that within the lower and upper bound widths of each wire, there exists an optimal SISS solution. However, this method is often too conservative. An alternative technique is to assign lower and upper bounds for w_E^{\downarrow} and w_E^{\uparrow} directly from the two-piece wire sizing that leads to the minimal delay objective during LR bound computation. Our experience shows that it leads to tight lower and upper bounds with good performance, so we recommend it for the asymmetric SISS.

3.2.2 Dynamic Programming for SISS

The SISS problem for a single net with fixed surrounding wire segments can be solved by adapting the bottom-up dynamic programming (DP)-based algorithm as in [34, 27, 35]. This DP procedure may be called directly

Bound Computation for SISS by LR

```

i ← 0
 $\mathcal{W}^U(i) \leftarrow$  Initialize wire width upper bound
 $\mathcal{W}^L(i) \leftarrow$  Initialize wire width lower bound
do
   $\mathcal{W}^U(i+1) \leftarrow$  Local refinement upper bound for each wire
   $\mathcal{W}^L(i+1) \leftarrow$  Local refinement lower bound for each wire
while ( $\mathcal{W}^U(i) \neq \mathcal{W}^U(i-1)$  OR  $\mathcal{W}^L(i) \neq \mathcal{W}^L(i-1)$ )

```

Figure 4: LR based algorithm to compute upper and lower bound widths for SISS.

without bound computation, denoted as SISS-DP, or it may be used after the bound computation procedure SISS-LR as presented in Section 3.2.1. The outline of the DP algorithm is given as follows.

During the bottom-up dynamic programming, each edge E is associated with a set of (c, t) pairs, called *options*. Let T_E denote the subtree rooted at the upstream end-point of E . Each (c, t) pair of E corresponds to a wire sizing and spacing solution for T_E , with c being the total capacitance of T_E , and t the performance measure (i.e., the weighted sum of sink delays from the upstream end-point of E) of T_E . Similarly, each node v is also associated with a set of (c, t) pairs defined in a similar manner for the subtree rooted at v .

The algorithm works in a bottom-up fashion, starting from the sinks. At the beginning, each sink is assigned with a (c, t) pair, with c being its loading capacitance, and $t = 0$. For an edge $E = uv$ (where node v is the downstream end-point of E), let (c', t') be an option of the subtree rooted at node v . For a candidate wire width w_E of E , we can compute the corresponding option (c, t) for E as follows:

$$c = c_E + c' \tag{5}$$

$$t = \lambda(E) \cdot r_E \cdot \left(\frac{c_E}{2} + c' \right) + t' \tag{6}$$

Note that r_E is the wire resistance and c_E is the lumped wire capacitance of E . In the SISS problem, only the wire widths of the net under optimization may be changed, and the layout of other nets is fixed. Therefore, given a particular candidate width (symmetric or asymmetric) for edge E , one can calculate the spacings between edge E and its neighbors easily. The wire width and spacing information is then passed to the capacitance model to calculate the total wire capacitance. Suppose E has r candidate widths (symmetric or asymmetric), and node v has k options, we have $r \times k$ possible options for E . To merge options from two edges, suppose we have an option (c', t') for edge $E = uv$ and another option (c'', t'') for edge $E' = uw$ (they share the same

upstream node u), then the corresponding merged option (c, t) at node u can be simply calculated as follows:

$$c = c' + c''$$

$$t = t' + t''$$

Similarly, if E has I options and E' has J options, then node u will have $I \times J$ options.

As in [35], given two options (c, t) and (c', t') at a particular node or edge, if $c \leq c'$ and $t' \geq t$, then (c', t') is sub-optimal. In other words, the wire sizing and spacing solution corresponding to (c', t') is *redundant*, and can be pruned from the set of candidate options. We apply the pruning rule and only keep irredundant options during the bottom-up computation.

At the end of the bottom-up computation of (c, t) pairs, the root will have a set of irredundant options. The optimal solution is achieved by choosing the (c, t) pair with the smallest t at the root. Tracing back from the optimal (c, t) pair, we obtain the corresponding optimal wire sizing and spacing solution that leads to this optimal option for the net. Thus, compared to previous DP algorithms in [34, 27, 35], our DP approach has the following new features: (1) we consider the coupling capacitance between neighboring wires; (2) we keep two wire widths (w_E^\downarrow and w_E^\uparrow) for each edge in the asymmetric scenario while performing bottom-up accumulation and top-down pruning.

To summarize, we present two algorithms for the SISS optimization: (1) SISS-LR, which computes the lower and upper bounds by LR; (2) SISS-DP, which computes the optimal SISS solution within its lower and upper bounds. Given an initial user input of wire width choices, one may directly call SISS-DP to obtain the optimal SISS solution or use SISS-LR to get a tight lower and upper bounds first, followed by the DP if needed. As we shall see in the experimental results in Section 5, the bound computation in SISS-LR is very effective such that most wire segments will have exact lower and upper bounds, then the DP procedure do not need to be invoked. In addition, the LR procedure is usually much faster than DP procedure. So our recommended optimization flow for SISS is to perform SISS-LR first, followed by SISS-DP, if necessary.

4 Global Interconnect Sizing and Spacing for Multiple Nets

The SISS algorithm is for optimizing a single critical net. However, this optimization will largely depend on the previous layout of other neighboring nets. Since many critical nets may share limited routing resources, just

optimizing one net may indeed sacrifice the performance of other critical nets. In this section, we will study the global wire sizing and spacing optimization for multiple nets. As one can see, the main difficulty of the GISS problem arises from the interaction of multiple nets through coupling capacitance which makes the global optimality difficult to compute. Moreover, the dependency of capacitance coefficients for each wire segment on the width and spacing is complex and there is usually no simple closed-form representation in DSM designs.

Before presenting our theory and algorithm on GISS, we first give a brief review of our previous approach in ICCAD'97 [22]. The *effective-fringing property* (EFP) was first proposed in [22], with the intention to simplify the GISS optimization problem into a set of single-net OWS-EF problems. For example, to compute/update the *global* lower bound widths of a net \mathcal{N}_i , one starts with lower bound widths of all other nets and compute the lower bound effective-fringing capacitance set for \mathcal{N}_i . Then, we obtain the OWS-EF solution to \mathcal{N}_i . It was believed that such an OWS-EF solution remains to be a lower bound of the optimal wire widths of \mathcal{N}_i under GISS optimization. Similar approach was used for upper bound computation. This EFP-based bound computation process was performed iteratively for all nets. However, it was found later on that such bound computations did not consider the impact on delays of neighboring nets during each OWS-EF optimization, thus may fail to get the optimal GISS solution, especially in the case that different nets have very different weights. For example, let us take two parallel bus lines, with weight=1 for line-1 and weight=10000 for line-2. Now we want to perform symmetric GISS for them. It is obvious that line-1 shall just use minimum wire width so that the coupling capacitance between these two bus lines, and thus the delay of line-2 can be as small as possible. Meanwhile, since the delay of line-1 carries so little weight that even using minimum width, it does not affect the overall weighted delay for these two lines. Note that line-2 may be sized up to reduce its delay as it carries a much larger weight. However, we found that the EFP-based lower bound computation flow sized up line-1, and failed to get the optimal GISS solution.

In this paper, we correct the above flow and develop new theory and algorithm that can guarantee the global optimality of bound computations. We will first establish the theoretical foundation of an important property, namely the *extended dominance property for multiple nets*, which is a more general case than the *extended dominance property for a single net* in the previous section. The beauty of such property is that we are then able to compute the *global* upper and lower bounds of the optimal GISS solution *directly* using a new *bound refinement* (BR) technique, under very general resistance and capacitance models (MRM and MLCM,

respectively).

4.1 Extended Dominance Property for Multiple Nets

First of all, it shall be noted that during GISS optimization, the capacitance of a wire segment E is not only a function (explicit or implicit) of its own width w_E , but also a function of the widths of its neighboring wire segment E_n 's. Then, the wire width change of a certain wire segment E will affect the following delay terms for the *multiple-net* delay objective in Eqn. (4):

- (i) for the net that E belongs to, the delay term that consists of E 's weighted upstream resistance times E 's capacitance, as well as E 's resistance times E 's downstream capacitance.
- (ii) for any neighboring wire segment of E , denoted as E_n without loss of generality, the delay term in E_n 's net due to E_n 's capacitance change when E 's width changes, i.e., E_n 's upstream resistance times E_n 's capacitance.

Since the widths of both E and E_n may be changed during GISS optimization, it makes GISS problem much more difficult to solve than SISS. We define two new *bound refinement* (BR) operations, and prove that using BR, we can compute the *global* upper and lower bounds of each wire segment for an optimal GISS solution. The BR operation is different from the previous *local refinement* (LR) operation in the SISS optimization or in [3, 36]. BR uses *both* upper and lower bounds to obtain *either* a new lower bound *or* a new upper bound, while LR uses *only* upper bound to compute a new upper bound, and *only* lower bound to get a new lower bound.

We define the following two *bound refinement* (BR) operations for a wire segment E under consideration.

Definition 8 Upper Bound Refinement: Let \mathcal{W}^U and \mathcal{W}^L be an upper and lower bound to an optimal GISS solution. We consider the following two initial settings for all wire segments: (i) all wire widths take their upper bound from \mathcal{W}^U ; (ii) all wire widths take their upper bound from \mathcal{W}^U , except for E 's neighboring wire segment E_n 's, which take their lower bound from \mathcal{W}^L . We compute the optimal width for E (i.e., to minimize the delay objective in Eqn. (4) with other wire widths fixed) under these two scenarios, denoted as $w_1^U(E)$ and $w_2^U(E)$, respectively. Then, the upper bound refinement (UBR) width for E is $\text{MAX}(w_1^U(E), w_2^U(E))$.

Definition 9 Lower Bound Refinement: Let \mathcal{W}^U and \mathcal{W}^L be an upper and lower bound to an optimal GISS solution. We consider the following two initial settings for all wire segments: (i) all wire widths take their

lower bound from \mathcal{W}^L ; (ii) all wire widths take their lower bound from \mathcal{W}^L , except for E 's neighboring wire segment E_n 's, which take their upper bound from \mathcal{W}^U . We compute the optimal width for E under these two scenarios, denoted as $w_1^L(E)$ and $w_2^L(E)$, respectively. Then, the lower bound refinement (LBR) width for E is $\text{MIN}(w_1^L(E), w_2^L(E))$.

We have proved that UBR and LBR operations can guarantee the optimality of *global* lower and upper bound computations for symmetric GISS under general resistance and capacitance models, as described in the following theorem.

Theorem 3 Extended Dominance Property for Multiple Nets: *Under MRM and MLCM models, the upper bound refinement for any wire segment will dominate its optimal width in an optimal GISS solution; the lower bound refinement for any wire segment will be dominated by its optimal width in an optimal GISS solution.*

Again, this is a new theorem not in [22]. The proof can be found in Appendix C. The *extended dominance property for multiple nets* and two bound refinement operations (UBR and LBR) associated with it will lead to a very effective, *global* upper and lower bound computation algorithm for the GISS optimization problem.

4.2 Algorithm for GISS Optimization

Similar to that for SISS, the algorithm for GISS also has two phases. The first phase is the upper and lower bound computation using bound refinement. The second phase is the dynamic-programming algorithm to compute the final GISS solution between the lower and upper bounds. In practice, our bound computation phase is very effective such that most wire segments have identical lower and upper bound.

4.2.1 Bound Computation for GISS

The *extended dominance property for multiple nets* leads to a very effective algorithm to compute upper and lower bounds of the optimal GISS solution. The bound computation algorithm for GISS is shown in Figure 5. The BR operations (UBR for upper bound computation and LBR for lower bound computation) are used to iteratively update the width upper and lower bounds for each wire segment of each net. The algorithm starts at the iteration $i = 0$. First we initialize upper and lower bounds of all wire widths specified by the user inputs and the layout constraints. A sample initialization is shown in Figure 6. Let E_l and E_u be two parallel horizontal edges, with E_l below E_u . Let W_{min} be the minimum wire width, and S_{min} be the minimum spacing from the

layout constraint. If the distance between the center-lines of E_l and E_u is d , then the maximum width (i.e., the initial upper bound) for $w_{E_l}^\uparrow$ (the side closer to E_u) and $w_{E_u}^\downarrow$ (the side closer to E_l) is $d - W_{min}/2 - S_{min}$. The overall flow of iterative bound computation for GISS is similar to that for SISS, whereas GISS uses BR operations and SISS uses LR operation. So we do not repeat the details here. Note that for the symmetric GISS, each wire segment is associated with one width, while for the asymmetric GISS, it is associated with two widths during bound refinements.

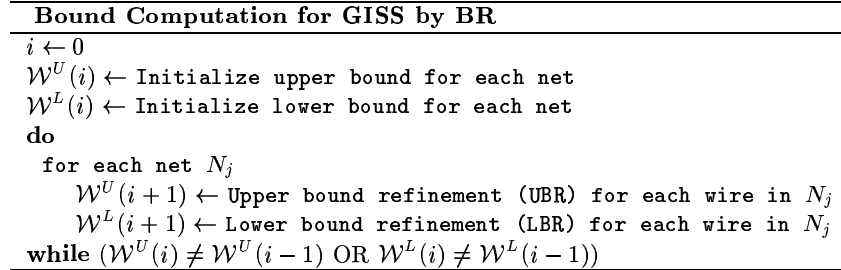


Figure 5: BR based algorithm to compute upper and lower bounds for GISS.

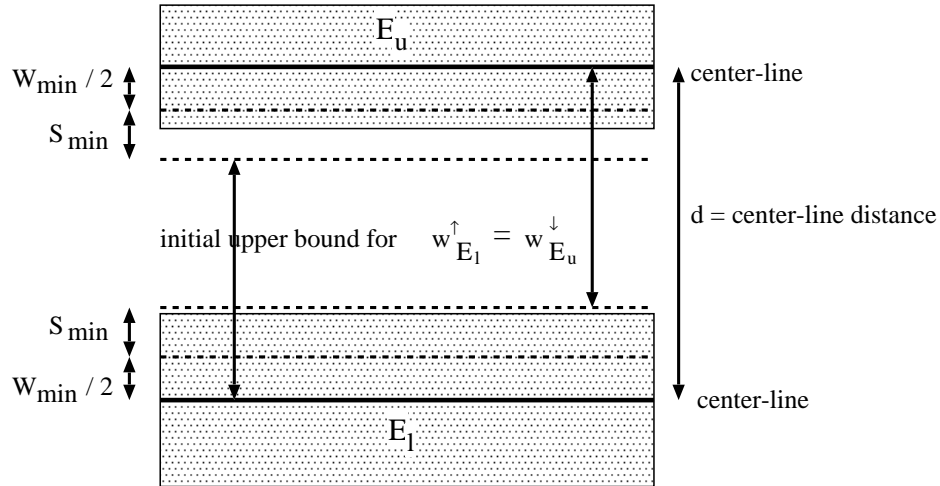


Figure 6: Initialization of upper-bound wire widths.

4.2.2 Dynamic Programming for GISS

In the case that the upper and lower bounds do not meet, we will use the following greedy DP algorithm to obtain the final wire sizing solution for each net. We first take the lower bound of each side for each wire segment as our initial layout. Then we will iteratively perform the bottom up DP algorithm similar to that presented in Section 3 to obtain the final GISS solution, in a net-by-net manner following the decreasing order of each net’s priority.

The major difference here from SISS is that in GISS, our objective is the multiple-net weighted delay. So the performance measure during the dynamic programming is now slightly modified from that in (6) to incorporate all neighboring structures for an edge E during the bottom-up accumulation, i.e.,

$$t = \lambda(E) \cdot r_E \cdot \left(\frac{c_E}{2} + c' \right) + t' + \sum_{E_n \in Neighbor(E)} R_{up}(E_n) \cdot c(w_{E_n}, w_E) \quad (7)$$

where the last term is the weighted delay summation of all E ’s neighboring nets that are related with w_E , i.e., those terms will affect the decision of bottom-up process at E . All other dynamic programming process for GISS is exactly the same as that for SISS, so we omit the details here. It shall be noted that the DP for single-net SISS guarantees the optimality, but the DP for multiple-net GISS may not guarantee it. But our optimal bound computation process for GISS usually obtains convergent or very tight lower and upper bounds, so the role of DP is pretty much optional. It mainly provides the post bound computation fine-tuning.

5 Experimental Results

We have implemented SISS and GISS algorithms using C++ on a SUN Ultra-Sparc1 with 256M-byte main memory. This section presents the experimental results. The parameters used in our experiments are based on a $0.18\mu m$ technology specified in NTRS roadmap [31]. We use a Simple Resistance Model, with the sheet resistance being $0.0638 \Omega/\square$. The minimum wire sizing W_{min} is $0.22\mu m$ and minimum edge-to-edge spacing S_{min} between neighboring wires is $0.33\mu m$. In this case, the *min-pitch*, defined as the sum of minimum spacing and minimum wire size, is equal to $0.55\mu m$. The allowable wire widths for each side of the center line are from 0.11 to $1.1 \mu m$, with the incremental step to be $0.11 \mu m$. The area, fringing and coupling capacitances are obtained by a table-look-up model simplified from the 2.5D model in [29]. The capacitance table is built by running a 3D capacitance extraction tool Fastcap [30]. The driver effective resistance for each net is set to be

	length (<i>mm</i>)	Weighted Delay (<i>ns</i>)				Maximum Delay (<i>ns</i>)			
		MIN	OWS	SISS-S	SISS-A	MIN	OWS	SISS-S	SISS-A
Net1	3.6	0.08	0.08 (-0.00%)	0.08 (-0.00%)	0.08 (-0.00%)	0.14	0.14 (-0.00%)	0.14 (-0.00%)	0.14 (-0.00%)
Net2	6.6	0.31	0.19 (-38.7%)	0.18 (-41.9%)	0.15 (-51.6%)	0.34	0.22 (-35.3%)	0.22 (-35.3%)	0.19 (-44.1%)
Net3	10.07	0.78	0.71 (-9.0%)	0.71 (-9.0%)	0.61 (-21.8%)	1.03	0.96 (-6.8%)	0.95 (-7.8%)	0.83 (-19.4%)
Net4	10.57	0.54	0.41 (-24.1%)	0.39 (-27.8%)	0.27 (-50.0%)	0.84	0.71 (-15.5%)	0.65 (-22.6%)	0.44 (-47.6%)
Net5	31.98	3.19	2.57 (-19.4%)	2.57 (-19.4%)	1.73 (-45.8%)	4.92	4.26 (-13.4%)	4.27 (-13.2%)	3.26 (-33.7%)

Table 2: Comparison of the weighted and maximum delays using different algorithms.

119 Ω . The loading capacitance at each sink is set to be 12.0fF.

5.1 Single-net Interconnect Sizing and Spacing

We perform experiments for the optimal single-net sizing and spacing algorithm on 5 nets extracted from an advanced industrial micro-processor. The routing trees are the same as those used in [4]. These trees originally have multiple sources and we randomly assign one as the unique single source. We assign equal criticality for each sink so the weighted delay becomes the average delay. Given the initial layout of these five nets, we randomly assign some surrounding wire segments with spacing from the net being 1 to $5 \times \text{min_pitch}$.

In Table 2, we summarize the weighted and maximum HSPICE delays from different algorithms: minimum wire sizing (MIN); optimal wire-sizing (OWS) algorithm without considering the coupling capacitance (but coupling capacitance through the 2D model is included in its final HSPICE simulation); symmetric SISS algorithm (SISS-S) and asymmetric SISS algorithm (SISS-A)⁵. In the parentheses under OWS, SISS-S and SISS-A, we list the percentage of improvement over MIN. From the table, we can see that SISS-A consistently outperforms all other algorithms. In terms of its weighted delay, which is our objective function, the improvement is up to 51.6% over the MIN solution (Net2), 34.1% over OWS (Net4), and 32.7% over SISS-S (Net5).

Although the weighted delay is our objective, our experimental results show that this formulation reduces the maximum delay as well. From the table, we can see that SISS-A reduces the maximum delay from MIN, OWS and SISS-S by up to 47.6%, 38.0% and 32.3% (Net4) compared with MIN, OWS and SISS-S respectively.

Although both SISS-DP and SISS-LR (possibly followed by DP if there is any unconvergent edge) lead to the same optimized delay, the latter is much more efficient in practice. This is due to the high convergence rate of the bound computation process. As shown in Table 3, our local refinement based bound computation leads to 100% convergence for both symmetric and asymmetric SISS for all 5 nets. The runtime by LR-S is only about

⁵The SISS/DP and SISS/LR lead to the same optimized delay, so we just use SISS to denote either one of them.

Net	Convergence rate		Run time (s)			
	LR-S	LR-A	DP-S	LR-S	DP-A	LR-A
net1	100%	100%	0.13	0.02	5.58	0.04
net2	100%	100%	0.71	0.03	41.1	0.06
net3	100%	100%	0.80	0.05	48.9	0.11
net4	100%	100%	0.91	0.04	39.8	0.09
net5	100%	100%	3.62	0.07	980	0.8

Table 3: The convergence rates in the LR bound computation for both symmetric and asymmetric wire sizing. Also shown are the runtimes of DP and LR based SISS algorithms,

PS	MIN	OWS	GISS-A
2 \times	1.51	1.26 (-16.6%)	0.76 (-49.7%)
3 \times	1.33	0.73 (-45.1%)	0.50 (-62.4%)
4 \times	1.28	0.46 (-64.1%)	0.40 (-68.8%)
5 \times	1.25	0.38 (-69.6%)	0.35 (-72.0%)
6 \times	1.23	0.35 (-71.5%)	0.32 (-74.0%)

Table 4: Comparison of weighted delays (in ns) for a 16-bit parallel bus structure using different algorithms under 5 different pitch-spacings (PS) from 2 to 6 \times min_pitch .

1/50 to 1/6 of that by DP-S. For asymmetric case, the speed-up of LR-A versus DP-A is even more significant (because there are now many more 2-piece wire-width choices to accumulate for DP), from 140 \times (net1) to more than 1,000 \times (net5) faster. Therefore, in practice, we recommend to use SISS-LR (followed by DP, if necessary).

5.2 Global Interconnect Sizing and Spacing for Multiple Nets

To demonstrate the effectiveness of our GISS algorithm on multiple nets, we perform experiments a 16-bit parallel bus structure of 10 mm long, with equal criticality for every bus line, and the center distance between adjacent bus line set to be $\{2, 3, 4, 5, 6\} \times min_pitch$, respectively. Each wire segment is set to be 500 μm .

Table 4 shows the weighted delays obtained from different algorithms after running HSPICE simulations. The percentage of delay reduction over MIN is still given in the parentheses. In the table, column OWS uses single-net OWS algorithm in a net by net manner, and GISS-A uses the asymmetric GISS algorithm (i.e., BR+DP). From Table 4, we can see that OWS outperforms MIN by up to 71.5% (bus 6 \times) and GISS-A further reduces the OWS delay by up to 40% (bus 2 \times). This concludes that one shall consider the coupling capacitance explicitly during the wire sizing/spacing optimization.

In Table 5, we compare different wire sizing and spacing algorithms with explicit consideration of coupling

PS	Weighted Delay (ns)				Average Width (μm)			
	EFP-A	STIS-A	GISS-A	GISS-S	EFP-A	STIS-A	GISS-A	GISS-S
$2\times$	0.81	0.80	0.76	0.82	0.68	0.69	0.66	0.56
$3\times$	0.57	0.53	0.50	0.53	1.05	1.11	0.94	0.95
$4\times$	0.46	0.45	0.40	0.41	1.34	1.49	1.21	1.24
$5\times$	0.39	0.36	0.35	0.36	1.46	1.65	1.44	1.50
$6\times$	0.36	0.32	0.32	0.33	1.48	1.64	1.63	1.66

Table 5: Comparison of weighted delays (in ns) for a 16-bit parallel bus structure using different algorithms.

capacitance. In the table, EFP stands for the *effective-fringing property* based algorithm in [22], STIS for the *extended local refinement* based algorithm in [37, 36], and GISS for our *bound refinement* based algorithm as in Section 4, respectively, for computing lower and upper bounds, then followed by the same bottom-up DP programming procedure to obtain the final solution between lower and upper bounds. We still use suffix ‘-A’ and ‘-S’ to denote the asymmetric and symmetric sizing, respectively. From Table 5, we can see that GISS-A consistently obtains less delay than both EFP-A (up to 13% for bus $4\times$) and STIS-A (up to 11% for bus $4\times$). Again, due to its flexibility to size asymmetrically around the center-lines, GISS-A outperforms GISS-S, with delay reduction by up to 7.3% (bus $2\times$). Meanwhile, it is interesting to observe that for simultaneous wire sizing and spacing, better delay does not necessarily consume larger wiring area. Our proposed algorithm GISS-A has less wiring area than STIS-A in all cases. For example of bus $4\times$, the average width by STIS-A is $1.49\mu m$, which is 23% larger than that by GISS-A ($1.21\mu m$). GISS-A also has less wire widths than EFP-A in most cases (except for bus $6\times$), with area saving of up to 10% for bus $2\times$.

Table 6 reports some statistics from lower and upper bound computations from the symmetric and asymmetric GISS algorithm, including the convergence rate (CR), the average gap between the final lower and upper bounds for each wire segment (Gap), the average number of BR operations for each edge ($\#BR$), and the runtime (CPU). We define a wire segment to be *convergent* if its lower and upper bounds are exactly the same. We can see that our BR bound computation is very effective as on average, 65% wire segments in symmetric GISS and 72% wire segments in asymmetric GISS are convergent. Even for those wire segments that are not convergent, the gaps between lower and upper bounds are very small, in most cases, just the wire width increment ($0.11\mu m$ in our experimental setting). The average gap between the lower and upper bounds for all wire segments is only from 0.011 to $0.067\mu m$. Note that since there are very few wire width choices for each wire segment, the DP

PS	GISS-S				GISS-A			
	CR	Gap (μm)	#BR	CPU(s): BR/DP	CR	Gap (μm)	#BR	CPU(s): BR/DP
2 \times	90.3%	0.011	2.95	0.41/0.10	85.2%	0.016	6.92	1.50/0.15
3 \times	61.9%	0.042	8.69	1.25/0.25	64.5%	0.040	13.8	4.35/0.62
4 \times	48.4%	0.061	9.76	1.64/0.44	66.6%	0.047	13.5	8.50/0.70
5 \times	48.4%	0.067	9.73	1.90/0.53	60.2%	0.057	14.3	11.4/2.30
6 \times	77.5%	0.025	6.96	1.56/0.17	83.4%	0.021	10.0	10.8/0.40
Avg	65.3%	0.041	7.62	1.35/0.30	72.0%	0.036	11.7	7.31/0.83

Table 6: The convergence rate (CR), average gap (Gap), average number of local refinement operations per edge (#BR) and total run times (CPU) using GISS-S and GISS-A algorithms for a 16-bit bus structure under 5 different pitch-spacings (PS) from 2 to 6 \times min_pitch .

algorithm runs very fast (see the CPU breakup between BR and DP in Table 6)⁶.

As mentioned in the beginning of Section 4, the EFP-based flow may fail to give correct lower and upper bound wire widths, especially when net weights differ drastically. To show such an example, we take two parallel bus lines with weights to be 1 and 10000, respectively. Other parameters are the same as those in the 16-bit bus example. The results by using EFP-A, STIS-A and GISS-A algorithms are listed in Table 7. It confirms that EFP-based algorithm cannot take the net weights into consideration, and may result in incorrect lower and upper bounds. In all cases, EFP-A leads to larger weighted delay (e.g., 20% for bus 2 \times) and wire width (e.g., 25% for bus 2 \times) than GISS-A. STIS-A in this example obtains delay/width comparable to GISS-A. But it uses significantly more CPU, by a factor of 40 \times of that by GISS-A. This is because the bound computation stage of STIS-A by ELR does not give good convergence. Then STIS-A has to rely heavily on our bottom-up DP procedure to pick the best solution between the lower and upper bounds. Our bound computation stage of GISS-A by BR operation, however, leads to 100% convergence rate for this weighted bus example, thus it is much more effective than STIS-A.

6 Conclusion

Our study shows convincingly that it is very important to consider coupling capacitance into VLSI interconnect optimization for deep submicron designs. We propose two wire-sizing formulations, symmetric and asymmetric, for interconnect sizing and spacing problem to a single (SISS) or multiple (GISS) nets. We reveal three important theorems, i.e., effective-fringing property and extended dominance property for a single net, and extended

⁶In fact, we also directly assigned the lower bound wire widths to be the final wire sizing solution (i.e., without the DP step) and found very little delay/area difference from the GISS results in Table 5.

<i>PS</i>	Weighted Delay (<i>ns</i>)			Average Width (μm)			CPU (s)		
	EFP-A	STIS-A	GISS-A	EFP-A	STIS-A	GISS-A	EFP-A	STIS-A	GISS-A
2×	0.463	0.395	0.394	1.309	1.172	1.155	5.93	4.27	0.12
3×	0.401	0.341	0.340	1.507	1.260	1.249	9.88	7.44	0.21
4×	0.377	0.315	0.315	1.683	1.353	1.342	14.7	11.9	0.26
5×	0.333	0.300	0.300	1.793	1.425	1.425	17.8	14.2	0.31
6×	0.306	0.295	0.295	1.793	1.425	1.425	18.0	14.4	0.32

Table 7: Comparison of weighted delay, average wire width, and CPU for two parallel bus lines with criticalities to be 1 and 10000, respectively using different algorithms.

dominance property for multiple nets to guide interconnect sizing and spacing optimization with consideration of coupling capacitance. Based on these properties, we develop efficient bound computation techniques, namely local refinement (LR) for SISS and bound refinement (BR) for GISS. When lower and upper bounds do not meet, we apply a bottom-up dynamic programming algorithm to get the final SISS or GISS solution. Our experiments show that both LR and BR can compute tight lower and upper bounds. Therefore, by bound computation alone, we can achieve optimal or near-optimal SISS and GISS solutions. In practice, it is recommended to use LR/BR for bound computation first, followed by the DP fine-tuning to obtain the final SISS/GISS solution.

The focus in this paper is on the performance side with consideration of the coupling capacitance. Another important effect of the coupling capacitance is crosstalk noise. In the future, we plan to develop efficient noise modeling and extend our LR/BR technique and DP algorithm for noise control and minimization.

Acknowledgments

The authors would like to thank Mosur K. Mohan, Krishnamurthy Soumyanath, Ganapati Srinivasa from Intel Corporation for their input and support of this project.

References

- [1] J. Cong, L. He, K.-Y. Khoo, C.-K. Koh, and D. Z. Pan, “Interconnect design for deep submicron ICs,” in *Proc. Int. Conf. on Computer Aided Design*, pp. 478–485, 1997.
- [2] J. Cong, K. S. Leung, and D. Zhou, “Performance-driven interconnect design based on distributed RC delay model,” in *Proc. Design Automation Conf.*, pp. 606–611, 1993.

- [3] J. Cong and K. S. Leung, "Optimal wiresizing under the distributed Elmore delay model," in *Proc. Int. Conf. on Computer Aided Design*, pp. 634–639, 1993.
- [4] J. Cong and L. He, "Optimal wiresizing for interconnects with multiple sources," *ACM Trans. on Design Automation of Electronics Systems*, vol. 1, pp. 478–511, Oct. 1996.
- [5] C. P. Chen, Y. W. Chang, and D. F. Wong, "Fast performance-driven optimization for buffered clock trees based on Lagrangian relaxation," in *Proc. Design Automation Conf*, pp. 405–408, 1996.
- [6] J. Lillis, C. K. Cheng, and T. T. Y. Lin, "Optimal wire sizing and buffer insertion for low power and a generalized delay model," in *Proc. Int. Conf. on Computer Aided Design*, pp. 138–143, Nov. 1995.
- [7] T. Okamoto and J. Cong, "Buffered Steiner tree construction with wire sizing for interconnect layout optimization," in *Proc. Int. Conf. on Computer Aided Design*, pp. 44–49, Nov. 1996.
- [8] J. P. Fishburn and C. A. Schevon, "Shaping a distributed-RC line to minimize Elmore delay," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 42, pp. 1020–1022, Dec. 1995.
- [9] C. P. Chen, Y. P. Chen, and D. F. Wong, "Optimal wire-sizing formula under the Elmore delay model," in *Proc. Design Automation Conf*, pp. 487–490, 1996.
- [10] C.-P. Chen and D. F. Wong, "Optimal wire sizing function with fringing capacitance consideration," in *Proc. Design Automation Conf*, pp. 604–607, 1997.
- [11] J. P. Fishburn, "Shaping a VLSI wire to minimize Elmore delay," in *Proc. European Design and Test Conf.*, 1997.
- [12] Y. Gao and D. F. Wong, "Optimal shape function for a bi-directional wire under elmore delay model," in *Proc. Int. Conf. on Computer Aided Design*, pp. 622–627, 1997.
- [13] N. Menezes, S. Pullela, F. Dartu, and L. T. Pillage, "RC interconnect synthesis — a moment fitting approach," in *Proc. Int. Conf. on Computer Aided Design*, pp. 418–425, 1994.
- [14] T. Xue, E. S. Kuh, and Q. Yu, "A sensitivity-based wiresizing approach to interconnect optimization of lossy transmission line topologies," in *Proc. IEEE Multi-Chip Module Conf.*, pp. 117–121, 1996.

- [15] J. Cong, L. He, C.-K. Koh, and P. H. Madden, "Performance optimization of VLSI interconnect layout," *Integration, the VLSI Journal*, vol. 21, pp. 1–94, 1996.
- [16] T. Gao and C. L. Liu, "Minimum crosstalk channel routing," in *Proc. Int. Conf. on Computer Aided Design*, pp. 692–696, Nov. 1993.
- [17] D. A. Kirkpatrick and A. L. Sangiovanni-Vincentelli, "Techniques for crosstalk avoidance in the physical design of high-performance digital systems," in *Proc. Int. Conf. on Computer Aided Design*, pp. 616–619, Nov. 1994.
- [18] K. L. Shepard and V. Narayanan, "Noise in deep submicron digital design," in *Proc. Int. Conf. on Computer Aided Design*, pp. 524–531, 1996.
- [19] A. Vittal and M. Marek-Sadowska, "Crosstalk reduction for VLSI," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, pp. 290–98, 1997.
- [20] C. J. Alpert, A. Devgan, and S. Quay, "Buffer insertion for noise and delay optimization," in *Proc. Design Automation Conf*, pp. 362–7, 1998.
- [21] C.-P. Chen and N. Menezes, "Noise-aware repeater insertion and wire sizing for on-chip interconnect using hierarchical moment-matching," in *Proc. Design Automation Conf*, pp. 502–506, June 1999.
- [22] J. Cong, L. He, C.-K. Koh, and D. Z. Pan, "Global interconnect sizing and spacing with consideration of coupling capacitance," in *Proc. Int. Conf. on Computer Aided Design*, pp. 628–633, 1997.
- [23] L. Vandenberghe, S. Boyd, and A. E. Gamal, "Optimal wire and transistor sizing for circuits with non-tree topology," in *Proc. Int. Conf. on Computer Aided Design*, pp. 252–259, 1997.
- [24] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *Journal of Applied Physics*, vol. 19, pp. 55–63, Jan. 1948.
- [25] J. Rubinstein, P. Penfield, Jr., and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. CAD-2, pp. 202–211, July 1983.
- [26] J. Cong and K. S. Leung, "Optimal wiresizing under the distributed Elmore delay model," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, pp. 321–336, Mar. 1995.

- [27] J. Lillis, C. K. Cheng, and T. T. Y. Lin, "Simultaneous routing and buffer insertion for high performance interconnect," in *Proc. the Sixth Great Lakes Symp. on VLSI*, 1996.
- [28] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Addison-Wesley, 1990.
- [29] J. Cong, L. He, A. B. Kahng, D. Noice, N. Shirali, and S. H.-C. Yen, "Analysis and justification of a simple, practical 2 1/2-d capacitance extraction methodology," in *Proc. ACM/IEEE Design Automation Conf.*, pp. 40.1.1–40.1.6, June, 1997.
- [30] K. Nabors and J. White, "Fastcap: A multipole accelerated 3-d capacitance extraction program," *IEEE Trans. on Computer-Aided Design*, vol. 10, pp. 1447–1459, Nov., 1991.
- [31] Semiconductor Industry Association, *National Technology Roadmap for Semiconductors*, 1994.
- [32] J. Cong, L. He, C.-K. Koh, and D. Z. Pan, "Global interconnect sizing and spacing with consideration of coupling capacitance," Tech. Rep. 970031, UCLA CS Dept, 1997. Available at <http://cadlab.cs.ucla.edu/~pan/publications/>.
- [33] J. Cong and L. He, "An efficient technique for device and interconnect optimization in deep submicron designs," in *Proc. Int. Symp. on Physical Design*, pp. 45–51, 1998.
- [34] L. P. P. van Ginneken, "Buffer placement in distributed RC-tree networks for minimal Elmore delay," in *Proc. IEEE Int. Symp. on Circuits and Systems*, pp. 865–868, 1990.
- [35] J. Cong, C.-K. Koh, and K.-S. Leung, "Simultaneous buffer and wire sizing for performance and power optimization," in *Proc. Int. Symp. on Low Power Electronics and Design*, pp. 271–276, Aug. 1996.
- [36] J. Cong and L. He, "Theory and algorithm of local-refinement based optimization with application to device and interconnect sizing," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, pp. 406–420, April 1999.
- [37] L. He, *Modeling and Optimization of VLSI Interconnects*. PhD thesis, University of California, Los Angeles, 1999.
- [38] J. Cong and C.-K. Koh, "Simultaneous driver and wire sizing for performance and power optimization," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 2, pp. 408–423, Dec. 1994.

[39] C.-K. Koh, *VLSI Interconnect Layout Optimization*. PhD thesis, University of California, Los Angeles, 1998.

Appendix

A. Proof of Effective-Fringing Property for Single Net

In the following, we prove the effective-fringing property. First, we state two previously reported results that will be used in our proof. [38, 39] showed the following relation between driver sizing and wire sizing solutions.

Lemma 1 [38, 39] DS/WS Relation: *For any tree T with one or more critical sinks, let R be the resistance of the driver driving the routing tree and \mathcal{W} be the corresponding optimal wire width assignment. Let R' be the resistance of another driver and \mathcal{W}' be the corresponding optimal wire width assignment. Then, $R < R'$ implies \mathcal{W} dominates \mathcal{W}' .* \square

The authors of [35, 39] investigated the relation between the loading capacitances and wiresizing solutions and showed the following:

Lemma 2 [35, 39] WS/CL Relation: *Let c_{s_k} be the loading capacitance of sink s_k for a routing tree T , and \mathcal{W} be the corresponding optimal wire-sizing solution. If we increase the loading capacitance from c_{s_k} to c'_{s_k} , and let \mathcal{W}' be the new optimal wire-sizing solution. Then, then there exists an optimal wire-sizing solution \mathcal{W}' that dominates \mathcal{W} .* \square

For ease of reference, we re-write the performance measure t_T for a routing tree T based on the Eqn. (3) (for compaction of notations, we drop the index i):

$$\begin{aligned}
t_T(\mathcal{W}) &= \lambda(s_0) \cdot R \cdot \sum_{s_k \in \text{sink}(T)} c_{s_k} + \lambda(s_0) \cdot R \cdot \sum_{E \in T} c_E \\
&+ \sum_{E \in T} \lambda(E) \cdot r_E \cdot \left\{ \frac{c_E}{2} + \sum_{s_k \in \text{sink}(E)} c_{s_k} + \sum_{E' \in \text{Des}(E)} c_{E'} \right\} \tag{8}
\end{aligned}$$

$$\begin{aligned}
&= \text{constant} + \lambda(s_0) \cdot R \cdot \sum_{E \in T} [c_a \cdot w_E + c_{ef}(E)] + \sum_{E \in T} \lambda(E) \cdot \frac{r}{w_E} \\
&\cdot \left\{ \frac{c_a \cdot w_E + c_{ef}(E)}{2} + \sum_{s_k \in \text{sink}(E)} c_{s_k} + \sum_{E' \in \text{Des}(E)} (c_a \cdot w_{E'} + c_{ef}(E')) \right\} \tag{9}
\end{aligned}$$

Note that for EFP, we assume a fixed area capacitance c_a , but edges may have different effective fringing capacitances. Let the set of effective fringing capacitance be \mathcal{C}_{ef} and $\hat{\mathcal{W}}(c_a, \mathcal{C}_{ef})$ be an optimal wire-sizing

solution (OWS-EF) for some constant c_a and C_{ef} (for ease of presentation, we will abbreviate the notation of $\hat{W}(c_a, C_{ef})$ into \hat{W}). Similarly, we denote \hat{W}' to be an OWS-EF solution for the same c_a , but under a different set of effective fringing capacitance set C'_{ef} . Let \hat{w}_E be the width of E in \hat{W} and \hat{w}'_E be the width of E in \hat{W}' .

First, we start from a simpler scenario that C_{ef} and C'_{ef} differ in only one wire segment. We have the following lemma.

Lemma 3 *For two different sets of effective-fringing capacitance C_{ef} and C'_{ef} with $c_{ef}(E') = c'_{ef}(E')$ for any edge $E' \neq E$, but $c_{ef}(E) > c'_{ef}(E)$, we have*

$$\frac{\lambda(E)}{2} \cdot \left(\frac{1}{\hat{w}_E} - \frac{1}{\hat{w}'_E} \right) + \sum_{E' \in Ans(E)} \lambda(E') \cdot \left(\frac{1}{\hat{w}_{E'}} - \frac{1}{\hat{w}'_{E'}} \right) \leq 0 \quad (10)$$

Proof of Lemma 3: In the following, we use $t_T(c_{ef}(E), \hat{W})$ to denote the performance measure corresponding to the case where the effective-fringing capacitance of E is $c_{ef}(E)$ with corresponding optimal wire sizing solution \hat{W} . Similarly, we can denote $t_T(c'_{ef}(E), \hat{W}')$ to be the performance measure under the $c'_{ef}(E)$ and the corresponding optimal wire sizing solution \hat{W}' . Suppose we swap the two sizing solutions. Then, we obtain new performance measures $t_T(c_{ef}(E), \hat{W}')$ and $t_T(c'_{ef}(E), \hat{W})$. Let $\Delta_1 \equiv t_T(c_{ef}(E), \hat{W}) - t_T(c'_{ef}(E), \hat{W}')$. Then, from Eqn. (9) and according to the definition of local refinement:

$$\begin{aligned} \Delta_1 &= \lambda(s_0) \cdot R \cdot \sum_{E' \in T} c_a \cdot (\hat{w}_{E'} - \hat{w}'_{E'}) \\ &+ \sum_{E' \in T, E' \neq E} \lambda(E') \cdot \frac{r}{\hat{w}_{E'}} \cdot \left\{ \frac{c_a \cdot \hat{w}_{E'} + c_{ef}(E')}{2} + \sum_{s_k \in sink(E')} c_{s_k} + \sum_{E'' \in Des(E'), E'' \neq E} (c_a \cdot \hat{w}_{E''} + c_{ef}(E'')) \right\} \\ &- \sum_{E' \in T, E' \neq E} \lambda(E') \cdot \frac{r}{\hat{w}'_{E'}} \cdot \left\{ \frac{c_a \cdot \hat{w}'_{E'} + c_{ef}(E')}{2} + \sum_{s_k \in sink(E')} c_{s_k} + \sum_{E'' \in Des(E'), E'' \neq E} (c_a \cdot \hat{w}'_{E''} + c_{ef}(E'')) \right\} \\ &+ \sum_{E' \in Ans(E)} \lambda(E') \cdot \frac{r}{\hat{w}_{E'}} \cdot (c_a \cdot \hat{w}_E + c_{ef}(E)) - \sum_{E' \in Ans(E)} \lambda(E') \cdot \frac{r}{\hat{w}'_{E'}} \cdot (c_a \cdot \hat{w}'_E + c_{ef}(E)) \\ &+ \lambda(E) \cdot \frac{r}{\hat{w}_E} \cdot \left\{ \frac{c_a \cdot \hat{w}_E + c_{ef}(E)}{2} + \sum_{s_k \in sink(E)} c_{s_k} + \sum_{E' \in Des(E)} (c_a \cdot \hat{w}_{E'} + c_{ef}(E')) \right\} \\ &- \lambda(E) \cdot \frac{r}{\hat{w}'_E} \cdot \left\{ \frac{c_a \cdot \hat{w}'_E + c_{ef}(E)}{2} + \sum_{s_k \in sink(E)} c_{s_k} + \sum_{E' \in Des(E)} (c_a \cdot \hat{w}'_{E'} + c_{ef}(E')) \right\} \\ &\leq 0 \end{aligned}$$

where the first line corresponds to the terms related with the driver resistance, the second and the third lines correspond to those edge terms not related with E , the fourth line corresponds to E and its ancestor edges, and

the fifth and the sixth lines correspond to E and its downstream subtrees.

Similarly, let $\Delta_2 \equiv t_T(c'_{ef}(E), \hat{\mathcal{W}}') - t_T(c'_{ef}(E), \hat{\mathcal{W}})$, and we have $\Delta_2 \leq 0$. Summing Δ_1 and Δ_2 , we can easily cross the common terms out and obtain

$$\lambda(E) \cdot r \cdot \left(\frac{1}{\hat{w}_E} - \frac{1}{\hat{w}'_E} \right) \cdot \left(\frac{c_{ef}(E) - c'_{ef}(E)}{2} \right) + \sum_{E' \in \text{Ans}(E)} \lambda(E') \cdot r \cdot \left(\frac{1}{\hat{w}_{E'}} - \frac{1}{\hat{w}'_{E'}} \right) \cdot (c_{ef}(E) - c'_{ef}(E)) \leq 0$$

Since $c_{ef}(E) > c'_{ef}(E)$, Lemma 3 follows. \square

Let T_E be the single stem sub-tree rooted at E , $\hat{\mathcal{W}}(T_E)$ and $\hat{\mathcal{W}}'(T_E)$ be those wire sizing solutions of edges in T_E only. Then we have the following lemma for $\hat{\mathcal{W}}(T_E)$ and $\hat{\mathcal{W}}'(T_E)$.

Lemma 4 *Given two different sets of effective-fringing capacitance C_{ef} and C'_{ef} with $c_{ef}(E') = c'_{ef}(E')$ for any edge $E' \neq E$, and $c_{ef}(E) > c'_{ef}(E)$, we have $\hat{\mathcal{W}}(T_E)$ dominates $\hat{\mathcal{W}}'(T_E)$.*

Proof of Lemma 4: Denote the two nodes of E being u and v with u being the upstream node and v being the downstream one. Let T_v be the downstream subtree rooted at node v . We first argue that $\hat{\mathcal{W}}(T_v)$ dominates $\hat{\mathcal{W}}'(T_v)$ for all edges in T_v . The total weighted upstream resistance of T_v with sizing solution $\hat{\mathcal{W}}$ is

$$R(T_v) = \lambda(s_0) \cdot R + \lambda(E) \cdot r \cdot \frac{1}{2\hat{w}_E} + \sum_{E' \in \text{Ans}(E)} \lambda(E') \cdot r \cdot \frac{1}{\hat{w}_{E'}}$$

and the total weighted upstream resistance of T_v with sizing solution $\hat{\mathcal{W}}'$ is

$$R'(T_v) = \lambda(s_0) \cdot R + \lambda(E) \cdot r \cdot \frac{1}{2\hat{w}'_E} + \sum_{E' \in \text{Ans}(E)} \lambda(E') \cdot r \cdot \frac{1}{\hat{w}'_{E'}}$$

Since from Lemma 3, we have $R(T_v) \leq R'(T_v)$, then applying Lemma 1, we conclude that $\hat{\mathcal{W}}(T_v)$ dominates $\hat{\mathcal{W}}'(T_v)$.

Now, we consider the wire segment E and prove that $\hat{w}_E \geq \hat{w}'_E$. We keep the wire width assignment of all other edges, and swap only the wire width assignment of edge E in the two wire sizing solutions $\hat{\mathcal{W}}$ and $\hat{\mathcal{W}}'$, i.e., $\hat{\mathcal{W}}_{\text{swap}} = \hat{\mathcal{W}}$ except $\hat{w}_E \leftarrow \hat{w}'_E$, and $\hat{\mathcal{W}}'_{\text{swap}} = \hat{\mathcal{W}}'$ except $\hat{w}'_E \leftarrow \hat{w}_E$. Let Δ_1 be $t_T(c_{ef}(E), \hat{\mathcal{W}}) - t_T(c_{ef}(E), \hat{\mathcal{W}}_{\text{swap}})$ and Δ_2 be $t_T(c'_{ef}(E), \hat{\mathcal{W}}') - t_T(c'_{ef}(E), \hat{\mathcal{W}}'_{\text{swap}})$. Then,

$$\begin{aligned} \Delta_1 &= \lambda(s_0) \cdot R \cdot c_a \cdot (\hat{w}_E - \hat{w}'_E) + \\ &\quad \sum_{E' \in \text{Ans}(E)} \lambda(E') \cdot \frac{r}{\hat{w}_{E'}} \cdot c_a \cdot (\hat{w}_E - \hat{w}'_E) + \end{aligned}$$

$$\begin{aligned} & \lambda(E) \cdot \left(\frac{r}{\hat{w}_E} - \frac{r}{\hat{w}'_E} \right) \cdot \left(\frac{c_{ef}(E)}{2} + \sum_{s_k \in \text{sink}(E)} c_{s_k} + \sum_{E' \in \text{Des}(E)} (c_a \cdot \hat{w}_{E'} + c_{ef}(E')) \right) \\ & \leq 0 \end{aligned}$$

Obtaining Δ_2 in a similar fashion, then summing Δ_1 and Δ_2 together, we have the following expression:

$$\begin{aligned} & c_a \cdot (\hat{w}_E - \hat{w}'_E) \cdot \sum_{E' \in \text{Ans}(E)} \lambda(E') \cdot \left(\frac{r}{\hat{w}_{E'}} - \frac{r}{\hat{w}'_{E'}} \right) + \\ & \lambda(E) \cdot \left(\frac{r}{\hat{w}_E} - \frac{r}{\hat{w}'_E} \right) \cdot \left(\frac{c_{ef}(E) - c'_{ef}(E)}{2} + \sum_{E' \in \text{Des}(E)} c_a \cdot (\hat{w}_{E'} - \hat{w}'_{E'}) \right) \leq 0 \end{aligned}$$

Since $\mathcal{W}(T_v)$ dominates $\mathcal{W}'(T_v)$ and $c_{ef}(E) > c'_{ef}(E)$, if we assume that $\hat{w}_E < \hat{w}'_E$, then

$$\sum_{E' \in \text{Ans}(E)} \lambda(E') \cdot \left(\frac{1}{\hat{w}_{E'}} - \frac{1}{\hat{w}'_{E'}} \right) \geq 0$$

Combining it with Lemma 3, we will have $\frac{1}{\hat{w}_E} - \frac{1}{\hat{w}'_E} \leq 0$, which is a contradiction to the assumption that $\hat{w}_E < \hat{w}'_E$. Therefore, $\hat{w}_E \geq \hat{w}'_E$, and $\hat{\mathcal{W}}(T_E)$ dominates $\hat{\mathcal{W}}'(T_E)$. \square

Lemma 4 reveals the dominance relationship for E and its downstream wire segments, the following lemma extends this dominance relationship to the entire routing tree.

Lemma 5 *Consider any edge E in T . If $c_{ef}(E') = c'_{ef}(E')$ for any edge $E' \neq E$, but $c_{ef}(E) > c'_{ef}(E)$. Then, $\hat{\mathcal{W}}$ dominates $\hat{\mathcal{W}}'$.*

Proof of Lemma 5: Now consider node u , the upstream node of E . Create a sink s_u such that the loading capacitance of s_u is equivalent to the total capacitance of T_E with wire-sizing $\hat{\mathcal{W}}$. Create another tree $T' = T - T_E$, and that at node u , it has sink s_u . Note that the combination of the optimal wire-sizing solution of T' with $\hat{\mathcal{W}}(T_E)$ is exactly $\hat{\mathcal{W}}$. Similarly, let $T'' = T - T_E$, but at node u , it has sink s'_u with loading capacitance equivalent to the total capacitance of T_E with wire-sizing $\hat{\mathcal{W}}'$. From Lemma 4, $c_{s_u} > c_{s'_u}$. By applying Lemma 2, the wiresizing solution of T' dominates T'' . Therefore, the lemma holds. \square

So far, the proof is focused on the scenario that C_{ef} and C'_{ef} differ in only one wire segment. If there are more than one edges with $c_{ef} > c'_{ef}$, we only need a simple mathematical induction on all edges E with $c_{ef}(E) > c'_{ef}(E)$ and use Lemma 5 to prove Theorem 1.

B. Proof of Extended Dominance Property for a Single Net

For a wire segment E with width of w_E , we denote $r(w_E)$ and $c(w_E)$ to be its resistance and lumped capacitance⁷.

Then the performance measure t_T for the routing tree T under optimization in (3) can be rewritten as follows.

$$\begin{aligned}
t_T(\mathcal{W}) &= \lambda(s_0) \cdot R \cdot \sum_{E' \in T} c(w_{E'}) + \lambda(E) \cdot r(w_E) \cdot \frac{c(w_E)}{2} \\
&\quad + \lambda(E) \cdot r(w_E) \cdot C_{down}(E) + c(w_E) \cdot R_{up}(E) \\
&\quad + \sum_{E' \in T, E' \neq E} \lambda(E') \cdot r(w_{E'}) \cdot \left\{ \frac{c(w_{E'})}{2} + \sum_{s \in sink(E')} c_s + \sum_{E'' \in Des(E'), E'' \neq E} c(w_{E''}) \right\} \quad (11)
\end{aligned}$$

where $R_{up}(E) = \sum_{E' \in Ans(E)} \lambda(E') \cdot r(w_{E'})$.

Let \mathcal{W}^* be an optimal SISS solution for the routing tree T and \mathcal{W}' dominates \mathcal{W}^* . We perform local refinement on a wire segment E for \mathcal{W}' , and still denote the newly refined width of E to be w'_E . Let $t_T(\mathcal{W}^*)$ and $t_T(\mathcal{W}')$ be the performance measure corresponding to \mathcal{W}^* and \mathcal{W}' , respectively. We swap the wire width assignment of E in \mathcal{W}^* and \mathcal{W}' , i.e., $\mathcal{W}'_{swap} = \mathcal{W}^*$ except $w_E^* \leftarrow w'_E$, and $\mathcal{W}^*_{swap} = \mathcal{W}'$ except $w'_E \leftarrow w_E^*$. It shall be noted that when we prove the effective-fringing property, we swap the wire-sizing of the entire net; whereas for EDP, we only swap the wire-sizing of a wire segment. Let Δ_1 be $t_T(\mathcal{W}^*) - t_T(\mathcal{W}'_{swap})$ and Δ_2 be $t_T(\mathcal{W}') - t_T(\mathcal{W}'_{swap})$. Then similar to Appendix A, we have

$$\begin{aligned}
\Delta_1 &= \lambda(s_0) \cdot R \cdot [c(w_E^*) - c(w'_E)] + \lambda(E) \cdot [r(w_E^*) \cdot c(w_E^*) - r(w'_E) \cdot c(w'_E)] / 2 \\
&\quad + \lambda(E) \cdot [r(w_E^*) - r(w'_E)] \cdot \left[\sum_{s \in sink(E)} c_s + \sum_{E' \in Des(E)} c(w_{E'}^*) \right] \\
&\quad + [c(w_E^*) - c(w'_E)] \cdot \sum_{E' \in Ans(E)} \lambda(E') \cdot r(w_{E'}^*) \\
&\leq 0
\end{aligned}$$

$$\begin{aligned}
\Delta_2 &= \lambda(s_0) \cdot R \cdot [c(w'_E) - c(w_E^*)] + \lambda(E) \cdot [r(w'_E) \cdot c(w'_E) - r(w_E^*) \cdot c(w_E^*)] / 2 \\
&\quad + \lambda(E) \cdot [r(w'_E) - r(w_E^*)] \cdot \left[\sum_{s \in sink(E)} c_s + \sum_{E' \in Des(E)} c(w_{E'}^*) \right] \\
&\quad + [c(w'_E) - c(w_E^*)] \cdot \sum_{E' \in Ans(E)} \lambda(E') \cdot r(w_{E'}^*) \\
&\leq 0
\end{aligned}$$

⁷For simplifying notations, we drop the spacings, i.e., s_E^\downarrow and s_E^\uparrow , in the wire capacitance notation since they are determined by w_E in SISS formulation.

Summing Δ_1 and Δ_2 up, we have

$$\begin{aligned}
\Delta_1 + \Delta_2 &= \lambda(E) \cdot [r(w_E^*) - r(w_E')] \cdot \sum_{E' \in Des(E)} [c(w_{E'}^*) - c(w_{E'}')] \\
&\quad + [c(w_E^*) - c(w_E')] \cdot \sum_{E' \in Ans(E)} \lambda(E') \cdot [r(w_{E'}^*) - r(w_{E'}')] \\
&\leq 0
\end{aligned} \tag{12}$$

We only need to consider the sinks with the positive criticalities. Zero criticality does not contribute to the weighted delay. Therefore, only those edges with $\lambda(E) > 0$ are of interest. Since \mathcal{W}' dominates \mathcal{W}^* , we have $\sum_{E' \in Des(E)} [c(w_{E'}^*) - c(w_{E'}')] < 0$ or $\sum_{E' \in Ans(E)} \lambda(E') \cdot [r(w_{E'}^*) - r(w_{E'}')] > 0$.

Now if the dominance property does not hold, i.e., $w_E' < w_E^*$, we would have $r(w_E^*) - r(w_E') < 0$ since the wire resistance monotonically decreases as wire width increases, and $c(w_E^*) - c(w_E') > 0$ since the wire capacitance monotonically increases as wire width increases. As a result, we would have $\Delta_1 + \Delta_2 > 0$, which is a contradiction to (12). So $w_E' \geq w_E^*$, i.e., a local refinement of \mathcal{W}' still dominates the optimal sizing solution $\hat{\mathcal{W}}$, forming an upper bound for $\hat{\mathcal{W}}$.

Similarly, we can prove that if \mathcal{W}' is dominated by the optimal sizing solution \mathcal{W}^* , a local refinement of \mathcal{W}' will still be dominated by \mathcal{W}^* , providing a lower bound for \mathcal{W}^* .

C. Proof of Extended Dominance Property for Multiple Nets

To make the presentation easy, we first consider one neighboring wire. It is trivial to extend to more neighboring wires as we shall see during the proof. For a wire segment E , we denote $r(w_E)$ to be the resistance of E with width w_E , and $c(w_E, w_{E_n})$ to be the lumped capacitance of E with width w_E and neighboring wire width w_{E_n} . Denote E_n 's weighted upstream resistance to be $R_{up}(E_n)$, i.e., $R_{up}(E_n) = \sum_{E' \in Ans(E_n)} \lambda(E') \cdot r(w_{E'})$. We only need to write down the parts of Eqn. (4) that is related with E (cf. Eqn. (11)).

$$\begin{aligned}
t(\mathcal{W}) &= R_{up}(E_n) \cdot c(w_{E_n}, w_E) + \left[\lambda(s_0) \cdot R + \sum_{E' \in Ans(E)} \lambda(E') \cdot r(w_{E'}) + \lambda(E) \cdot \frac{r(w_E)}{2} \right] \cdot c(w_E, w_{E_n}) \\
&\quad + \lambda(E) \cdot r(w_E) \cdot \left\{ \sum_{s \in sink(E)} c_s + \sum_{E' \in Des(E)} c(w_{E'}, w_{E_n}') \right\}
\end{aligned} \tag{13}$$

We first prove the upper bound case. The lower bound case can be proved symmetrically. Let \mathcal{W}^* be an optimal GISS solution, and $\mathcal{W}^U/\mathcal{W}^L$ be an upper/lower bound for \mathcal{W}^* . We perform upper bound refinement (UBR) to a wire segment E using the following two initial settings of \mathcal{W}' , which is either (i) all wire widths

in \mathcal{W}' take their upper bound from \mathcal{W}^U ; or (ii) all wire width in \mathcal{W}' take their upper bound from \mathcal{W}^U , except for E 's neighboring wire segment E_n 's, which take their lower bound from \mathcal{W}^L . Without introducing more notations, we still denote the bound refinement of wire segment E for \mathcal{W}' to be w'_E .

Let $t(\mathcal{W}^*)$ correspond to the delay under optimal solution \mathcal{W}^* and $t(\mathcal{W}')$ correspond to the delay under \mathcal{W}' . Similar to Appendix B, we swap the wire width assignment of E in the two wire sizing solutions, \mathcal{W}^* and \mathcal{W}' , i.e., $\mathcal{W}_{s_{wap}}^* = \mathcal{W}^*$ except $w_E^* \leftarrow w'_E$, and $\mathcal{W}'_{s_{wap}} = \mathcal{W}'$ except $w'_E \leftarrow w_E^*$. Let $\Delta_1 \equiv t(\mathcal{W}^*) - t(\mathcal{W}_{s_{wap}}^*)$ and $\Delta_2 \equiv t(\mathcal{W}') - t(\mathcal{W}'_{s_{wap}})$. According to the definition of BR, we have

$$\begin{aligned}
\Delta_1 &= R_{up}^*(E_n) \cdot [c(w_{E_n}^*, w_E^*) - c(w_{E_n}^*, w'_E)] \\
&\quad + \lambda(s_o) \cdot R \cdot [c(w_E^*, w_{E_n}^*) - c(w'_E, w_{E_n}^*)] + \lambda(E) \cdot [r(w_E^*) \cdot c(w_E^*, w_{E_n}^*) - r(w'_E) \cdot c(w'_E, w_{E_n}^*)] / 2 \\
&\quad + [c(w_E^*, w_{E_n}^*) - c(w'_E, w_{E_n}^*)] \cdot \sum_{E' \in Ans(E)} \lambda(E') \cdot r(w_{E'}^*) \\
&\quad + \lambda(E) \cdot [r(w_E^*) - r(w'_E)] \cdot \left[\sum_{s \in sink(E)} c_s + \sum_{E' \in Des(E)} c(w_{E'}^*, w_{E_n}^*) \right] \\
&\leq 0
\end{aligned}$$

$$\begin{aligned}
\Delta_2 &= R'_{up}(E_n) \cdot [c(w'_{E_n}, w'_E) - c(w'_{E_n}, w_E^*)] \\
&\quad + \lambda(s_o) \cdot R \cdot [c(w'_E, w'_{E_n}) - c(w_E^*, w'_{E_n})] + \lambda(E) \cdot [r(w'_E) \cdot c(w'_E, w'_{E_n}) - r(w_E^*) \cdot c(w_E^*, w'_{E_n})] / 2 \\
&\quad + [c(w'_E, w'_{E_n}) - c(w_E^*, w'_{E_n})] \cdot \sum_{E' \in Ans(E)} \lambda(E') \cdot r(w_{E'}^*) \\
&\quad + \lambda(E) \cdot [r(w'_E) - r(w_E^*)] \cdot \left[\sum_{s \in sink(E)} c_s + \sum_{E' \in Des(E)} c(w_{E'}^*, w'_{E_n}) \right] \\
&\leq 0
\end{aligned}$$

Note that the width of E_n does not change during BR of E , but its spacing to E changes as E 's width changes. Summing Δ_1 and Δ_2 up, we have

$$\begin{aligned}
\Delta_1 + \Delta_2 &= R_{up}^*(E_n) \cdot [c(w_{E_n}^*, w_E^*) - c(w_{E_n}^*, w'_E)] + R'_{up}(E_n) \cdot [c(w'_{E_n}, w'_E) - c(w'_{E_n}, w_E^*)] \\
&\quad + \lambda(s_o) R \cdot [c(w_E^*, w_{E_n}^*) - c(w'_E, w_{E_n}^*) + c(w'_E, w'_{E_n}) - c(w_E^*, w'_{E_n})] \\
&\quad + \lambda(E) \cdot [r(w_E^*) \cdot c(w_E^*, w_{E_n}^*) - r(w'_E) \cdot c(w'_E, w_{E_n}^*) + r(w'_E) \cdot c(w'_E, w'_{E_n}) - r(w_E^*) \cdot c(w_E^*, w'_{E_n})] / 2 \\
&\quad + [c(w_E^*, w_{E_n}^*) - c(w'_E, w_{E_n}^*)] \cdot \sum_{E' \in Ans(E)} \lambda(E') \cdot r(w_{E'}^*)
\end{aligned}$$

$$\begin{aligned}
& + [c(w'_E, w'_{E_n}) - c(w^*_E, w'_{E_n})] \cdot \sum_{E' \in \text{Ans}(E)} \lambda(E') \cdot r(w'_{E'}) \\
& + \lambda(E) \cdot [r(w'_E) - r(w^*_E)] \cdot \sum_{E' \in \text{Des}(E)} [c(w'_{E'}, w'_{E'_n}) - c(w^*_{E'}, w^*_{E'_n})] \\
& \leq 0
\end{aligned}$$

We only need to consider the sinks with the positive criticalities. Zero criticality does not contribute to the weighted delay. Therefore, only those edges with $\lambda(E) > 0$ are of interest. Since \mathcal{W}' dominates \mathcal{W}^* for any wire segment $E' \neq E_n$, regardless of which initial setting in (i) and (ii) used to locally optimize wire width of E , one can easily obtain the following inequalities.

$$\begin{aligned}
R_{up}^*(E_n) - R'_{up}(E_n) & > 0 \\
\sum_{E' \in \text{Des}(E)} [c(w'_{E'}, w'_{E'_n}) - c(w^*_{E'}, w^*_{E'_n})] & > 0 \\
\sum_{E' \in \text{Ans}(E)} \lambda(E') \cdot [r(w^*_{E'}) - r(w'_{E'})] & > 0
\end{aligned}$$

To simplify the capacitance terms, we define for two wire-width pairs of E and E_n , the following two capacitance modes.

Definition 10 Convex Mode: For a wire segment E and its neighboring wire segment E_n with a fixed pitch-spacing between them, suppose we have two different widths for E such that $w^*_E > w'_E$, and two different width for E_n such that $w^*_{E_n} > w'_{E_n}$. Then the capacitance model is in the convex mode if

$$c(w^*_E, w^*_{E_n}) + c(w'_E, w'_{E_n}) > c(w^*_E, w'_{E_n}) + c(w'_E, w^*_{E_n}). \quad (14)$$

The convex capacitance mode is defined in an analogous way as a convex function, which essentially states that the sum of two “end-point” values will be larger than the sum of two “intermediate-point” values. (14) can also be written as

$$c(w^*_E, w^*_{E_n}) - c(w^*_E, w'_{E_n}) > c(w'_E, w^*_{E_n}) - c(w'_E, w'_{E_n}),$$

which implies that given a fixed pitch-spacing, for the same increase of E_n 's wire width, the resulting capacitance increase for E is larger if E 's width is larger. Our study concludes that the convex capacitance mode usually holds for DSM designs. Table 8 shows the capacitance values under some typical pitch-spacing and wire width combinations. To simplify the table dimension, we only include the cases where $w^*_E = w^*_{E_n}$ and $w'_E = w'_{E_n}$. We

can see that in all table entries, $c(w_1, w_1) + c(w_2, w_2) > c(w_1, w_2) + c(w_2, w_1)$, i.e., the capacitance model is in the convex mode.

We can also define the counterpart of the convex mode, i.e., the concave mode, for the sake of completeness.

Definition 11 Concave Mode: *For a wire segment E and its neighboring wire segment E_n with a fixed pitch-spacing between them, suppose we have two different widths for E such that $w_E^* > w'_E$, and two different width for E_n such that $w_{E_n}^* > w'_{E_n}$. Then the capacitance model is in the concave mode if*

$$c(w_E^*, w_{E_n}^*) + c(w'_E, w'_{E_n}) < c(w_E^*, w'_{E_n}) + c(w'_E, w_{E_n}^*). \quad (15)$$

Now we prove by contradiction. If the EDP does not hold, i.e., $w_E^* > w'_E$, then for the two capacitance modes:

- If the capacitance is in the concave mode at wire segment E , we have $w_{E_n}^* < w'_{E_n}$ (i.e., the starting point for UBR should use initial setting (i)), and $w_E^* > w'_E$ (the contradictory assumption). Then from the definition of the concave mode, we have

$$\begin{aligned} c(w_{E_n}^*, w_E^*) - c(w_{E_n}^*, w'_E) &> c(w'_{E_n}, w_E^*) - c(w'_{E_n}, w'_E) \\ c(w'_E, w'_{E_n}) - c(w'_E, w_{E_n}^*) &> c(w_E^*, w'_{E_n}) - c(w_E^*, w_{E_n}^*) \end{aligned}$$

Together with the three inequalities, we would conclude that $\Delta_1 + \Delta_2 > 0$, which is a contradiction.

- If the capacitance is in the convex mode at wire segment E , we have $w_{E_n}^* > w'_{E_n}$ (i.e., the starting point for UBR should use initial setting (ii)), and $w_E^* > w'_E$ (the contradictory assumption). Then from the definition of the convex mode, we have

$$\begin{aligned} c(w_{E_n}^*, w_E^*) - c(w_{E_n}^*, w'_E) &> c(w'_{E_n}, w_E^*) - c(w'_{E_n}, w'_E) \\ c(w_E^*, w_{E_n}^*) - c(w_E^*, w'_{E_n}) &> c(w'_E, w_{E_n}^*) - c(w'_E, w'_{E_n}) \end{aligned}$$

And again, it leads to the contradiction of $\Delta_1 + \Delta_2 > 0$.

Note that the optimal GISS solution \mathcal{W}^* is unknown during the bound refinement phase, so we will check both capacitance modes, and compute the corresponding locally optimal width for E under the two corresponding initial settings of \mathcal{W}' in the upper bound refinement (UBR) operation, i.e., $w_1^U(E)$ and $w_2^U(E)$. We then take

pitch-sp	w_1	w_2	$c(w_1, w_2) + c(w_2, w_1)$	$c(w_1, w_1) + c(w_2, w_2)$	Cap. Mode
110	22	44	0.2806	0.2839	Convex
110	22	66	0.3088	0.3291	Convex
110	44	66	0.3452	0.3530	Convex
165	22	44	0.2333	0.2350	Convex
165	22	66	0.2458	0.2507	Convex
165	22	88	0.2604	0.2732	Convex
165	22	110	0.2769	0.3076	Convex
165	22	132	0.2968	0.3767	Convex
165	44	66	0.2575	0.2586	Convex
165	44	88	0.2745	0.2811	Convex
165	44	110	0.2948	0.3155	Convex
165	44	132	0.3192	0.3846	Convex
165	66	88	0.2948	0.2968	Convex
165	66	110	0.3192	0.3312	Convex
165	66	132	0.3498	0.4003	Convex
165	88	110	0.3498	0.3536	Convex
165	88	132	0.3881	0.4227	Convex
165	110	132	0.4418	0.4571	Convex
220	22	44	0.2217	0.2220	Convex
220	22	66	0.2296	0.2306	Convex
220	22	88	0.2385	0.2410	Convex
220	22	110	0.2478	0.2560	Convex
220	22	132	0.2577	0.2758	Convex
220	22	154	0.2683	0.3046	Convex
220	22	176	0.2820	0.3516	Convex
220	44	66	0.2344	0.2347	Convex
220	44	88	0.2438	0.2451	Convex
220	44	110	0.2539	0.2601	Convex
220	44	132	0.2646	0.2798	Convex
220	44	154	0.2787	0.3087	Convex
220	44	176	0.2944	0.3557	Convex
220	154	176	0.4305	0.4384	Convex
275	44	198	0.2783	0.3089	Convex
330	66	132	0.2495	0.2504	Convex
330	110	198	0.2922	0.2963	Convex

Table 8: The convex capacitance mode is usually held. Parameters are based on a $0.18\mu m$ technology. The unit for pitch-spacing, w_1 and w_2 is in 1/100 micron. The unit for capacitance is in fF .

the larger one of them, i.e., $MAX(w_1^U(E), w_2^U(E))$ and it will guarantee to be an upper bound width of the optimal wire width of E in the optimal GISS solution \mathcal{W}^* . Similarly, we can prove the lower bound case, using the lower bound refinement (LBR) operation. If there are more than one neighboring wire segments to E , we just need to sum up those E_n terms into Δ_1 and Δ_2 .