

Optimal Wiresizing Under the Distributed Elmore Delay Model

Jason Cong and Kwok-Shing Leung
Department of Computer Science
University of California, Los Angeles, CA 90024

Abstract

In this paper, we study the optimal wiresizing problem under the distributed Elmore delay model. We show that the optimal wiresizing solutions satisfy a number of interesting properties, including the separability, the monotone property, and the dominance property. Based on these properties, we develop a polynomial-time optimal wiresizing algorithm for arbitrary interconnect structures under the distributed Elmore delay model. Extensive experimental results show that our wiresizing solution reduces interconnection delay by up to 51% when compared to the uniform-width solution of the same routing topology. Furthermore, compared to the wiresizing solution based on a simpler RC delay model in [7], our wiresizing solution reduces the total wiring area by up to 28% while further reducing the interconnection delays to the timing-critical sinks by up to 12%.

1 Introduction

As the VLSI fabrication technology reaches submicron device dimension and gigahertz frequency, interconnection delay has become the dominant factor in determining circuit speed [9, 14]. The analysis in [15] and [7] showed that in the conventional VLSI technology, interconnection delay is determined by the product of the driver resistance and the total wire capacitance. As a result, the minimum interconnection delay is achieved when the routing tree is an optimal Steiner tree with the minimum wire width for each segment (since it has the minimum total wire capacitance). Therefore, conventional global and detailed routers aimed at generating minimum-width Steiner routing trees using the least total wirelength. However, as we reduce the device dimension, the driver resistance becomes smaller and the wire resistance becomes larger, which results in a much larger resistance ratio (defined to be the ratio of the driver resistance versus the unit wire resistance). In this case, the *distributed* nature of the interconnect structure must be considered, and minimizing the total wire capacitance does not necessarily lead to the minimum interconnection delay.

Most existing works on performance-driven VLSI routing concentrate on optimizing the interconnect topology of the routing trees for reducing interconnection delay. In [5], a timing-driven global router was proposed to minimize both the cost (i.e. the total wirelength) and the radius (i.e. the longest path from the source to any sink) simultaneously. Another cost-radius tradeoff was achieved in [1] with further improvement in performance [3]. Both the maximum performance tree formulation in [4] and the A-tree formulation in [7] aimed at constructing a minimum-wirelength routing tree which has the shortest path connection between the source and every sink.

Experimental results showed that the algorithm in [7] can construct A-trees which are at most 4% within the optimal, and achieve interconnection delay reduction by as much as 66% when compared to the best-known Steiner routing topology. When the critical-path information is available, the critical sink routing approaches in [2] reduce the delays to specified sinks substantially.

Although steady progress has been made in optimizing interconnect topology design for delay minimization, there were very few works on wiresizing optimization for high-performance interconnect designs. Wiresizing was used by Fisher and Kung [11] in H-tree clock routing. Recently, Cong, Leung, and Zhou [7] developed an optimal wiresizing algorithm based on minimizing an upper bound of the delay in a distributed RC tree proposed by Rubinstein, Penfield and Horowitz [13], which is given by:

$$t = \sum_{\text{all nodes } k} R_k \cdot c_k \quad (1)$$

where R_k is the path resistance between the source and the node k and c_k is capacitance at the node k . This upper-bound delay model was chosen in [7] because it simplifies the wiresizing optimization. However, the simplicity of this delay model also results in several drawbacks. First, it provides only an upper bound of the worst-case RC delay in the routing tree and does not distinguish the delays at different sinks. Therefore, it is impossible to optimize the wiresizing solution to reduce the delays to the specific timing-critical sinks. Moreover, since this model tends to over-estimate the delays at many sinks in the routing tree, it often results in unnecessary over-sizing of many wire segments. Oversized wires not only occupy more routing spaces, but also increase the mutual capacitance and inductance between different signal nets. Thus, there is a strong need to develop optimal wiresizing algorithms under more accurate interconnection delay models.

In this paper, we study the optimal wiresizing problem under the distributed Elmore delay model [10, 13]. We show that the optimal wiresizing solutions satisfy a number of interesting properties, including the separability, the monotone property, and the dominance property. Based on these properties, we develop a polynomial-time optimal wiresizing algorithm for arbitrary interconnect structures under the distributed Elmore delay model.

2 Problem Formulation

Assume that we are given a routing tree T implementing a signal net which consists of a source N_+ , and a set of m sinks $\{N_1, N_2, \dots, N_m\}$. A node in T refers to the source, or a sink,

or a Steiner node, and a segment in T connects two nodes. Assume that $\{E_1, E_2, \dots, E_n\}$ is the set of segments forming the tree T , where n is the total number of segments in the tree. Notice that n is one less than the total number of nodes in the tree.

In order to model a routing tree as a distributed RC circuit accurately, a 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 as shown in Figure 1. (Adjacent grid points are unit length apart.) For each grid point u in the tree T , we use r_u and c_u to denote the resistance and capacitance, respectively, of the grid edge ended at u , and use c_u^s to denote the node capacitance at u . For simplicity, we assume that $c_u^s = c^s$ (a constant) if u is a sink, and 0 otherwise. To correctly model the driver resistance, we introduce an additional node N_0 and connect N_0 to N_+ via an additional segment with resistance R_d (the driver resistance). Since each grid point u is uniquely identified with an incoming grid edge in the routing tree, we also use u to refer to that grid edge as well in the later discussions. Given a grid point u , we use $Des(u)$ to denote the set of grid points in the subtree rooted at u (excluding u), and $Ans(u)$ to denote the set of grid points $\{v | u \in Des(v)\}$ (again, excluding u). That is, $Des(u)$ is the set of “descendant” grid points of u , and $Ans(u)$ is the set of “ancestor” grid points of u . Moreover, we use C_u to denote the total capacitance in the subtree rooted at u (including both the wire capacitances and the sink capacitances). Furthermore, we use $P(u, v)$ to denote the unique path from u to v for any grid point u, v in the routing tree.

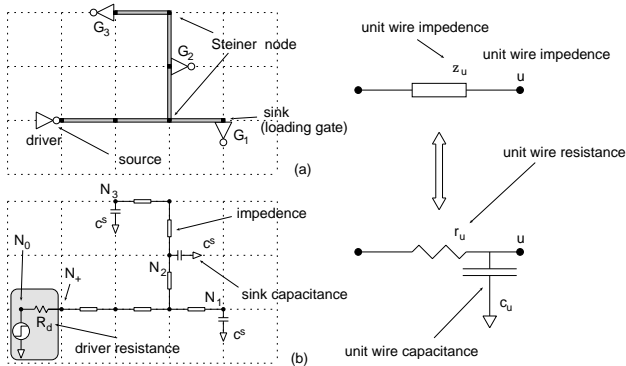


Figure 1: A grid structure for the Elmore delay model. (a) The layout of an interconnect T with 3 sinks (N_1, N_2 , and N_3). (b) The corresponding Elmore delay model of T . Each grid edge in T connects two adjacent grid points, and is modeled as a RC element containing a resistance r_u and a capacitance c_u , where u is the farther end of the grid edge from the source. Each sink has an extra loading capacitance c^s .

We assume that each wire segment has a set of discrete choices of wire widths $\{W_1, W_2, \dots, W_r\}$, and the wire width within the same segment does not change. This segment-based wiresizing model resembles more closely to the realistic design style and reflects the actual technological constraint where arbitrary width variation *within a segment* is usually undesirable.

Nevertheless, this segment-based formulation can be generalized to handle the case where variable wire width is allowed within a segment, simply by introducing artificial degree-2 Steiner nodes in the segment. Given a node u , we use w_u to denote the width of the grid edge u , and w_E and l_E to denote the width and length of the segment E , respectively. Assume that a unit-width unit-grid-length wire has wire resistance r_0 and wire capacitance c_0 , then $r_u = \frac{r_0}{w_u}$ and $c_u = c_0 \cdot w_u$ for any grid edge u .

In this paper, we use the Elmore delay model [10] as the objective function for delay optimization. Given a distributed RC circuit tree T , the signal delay at a particular node N_i , denoted as $t(N_i)$, is computed as follows:

$$t(N_i) = \sum_{u \in P(N_0, N_i)} r_u \cdot \left(\frac{c_u}{2} + C_u \right) \quad (2)$$

where the summation is taken over all the grid points on the path from the driver N_0 to the node N_i .

2.1 Single Critical Sink Formulation

We shall first study the case where there is only one critical sink N_i in the net. According to (2), the signal delay $t(N_i)$ at N_i is given by:

$$\begin{aligned} t(N_i) &= \sum_{u \in P(N_0, N_i)} r_u \cdot \left(\frac{c_u}{2} + C_u \right) \\ &= \sum_{u \in P(N_+, N_i)} \frac{r_0 \cdot c_0}{2} + R_d \cdot \sum_{u \in T} c_u^s + \\ &\quad R_d \cdot c_0 \cdot \sum_{u \in T} w_u + r_0 \cdot c_0 \cdot \sum_{u \in P(N_+, N_i)} \sum_{v \in Des(u)} \frac{w_v}{w_u} \\ &\quad + r_0 \cdot \sum_{u \in P(N_+, N_i)} \sum_{v \in Des(u)} c_v^s \cdot \frac{1}{w_u} \end{aligned} \quad (3)$$

After a sequence of transformation, we can show that minimizing the signal delay $t(N_i)$ is equivalent to minimizing the following function:

$$\begin{aligned} T_i(\mathcal{W}) &= \mathcal{K}_3 \cdot \sum_{E \in T} l_E \cdot w_E + \\ &\quad \mathcal{K}_4 \cdot \sum_{E, E' \in T, E \neq E'} l_E \cdot l_{E'} \cdot f_i(E, E') \cdot \frac{w_{E'}}{w_E} + \\ &\quad \mathcal{K}_5 \cdot \sum_{E \in T} l_E \cdot g_i(E) \cdot \frac{1}{w_E} \end{aligned} \quad (4)$$

where $\mathcal{K}_3 = R_d \cdot c_0$, $\mathcal{K}_4 = r_0 \cdot c_0$, $\mathcal{K}_5 = r_0 \cdot c^s$, and the functions $f_i(E, E')$ and $g_i(E)$ are defined as follows:

$$f_i(E, E') = \begin{cases} 1 & \text{if } E \in P(N_+, N_i) \text{ and } E' \in Des(E) \\ 0 & \text{otherwise} \end{cases}$$

$$g_i(E) = |sink(T) \cap Des(u)| \quad \text{for any } u \in E$$

where $Des(E)$ is the set of segments in the subtree “rooted” at E (excluding E), and $Ans(E)$ is the set $\{E' | E \in Des(E')\}$. Please refer to [6] for details of the transformation. Note that the first term of (4) minimizes the total wiring area of T . Moreover, A careful study of f_i ’s and g_i ’s reveals that:

$$f_i(E_1, E_2) \geq f_i(E_1, E'_2) \quad \text{if } E_2 \in Des(E'_2) \quad (5)$$

$$f_i(E_1, E_2) \geq f_i(E'_1, E_2) \quad \text{if } E_1 \in Ans(E'_1) \quad (6)$$

$$g_i(E_1) \geq g_i(E'_1) \quad \text{if } E_1 \in Ans(E'_1) \quad (7)$$

Given a routing tree and a specified critical sink N_i , all $f_i(E_1, E_2)$'s can be precomputed and stored in a two-dimensional $n \times n$ matrix before wiresizing optimization. Similarly, all $g_i(E)$'s can be precomputed and stored in a linear array.

2.2 Multiple Critical Sink Formulation

When there are several critical sinks of different priorities in the routing tree, we can generalize the previous formulation to optimize:

$$T(\mathcal{W}) = \sum_{N_i \in \text{sink}(T)} \lambda_i \cdot T_i(\mathcal{W}) \quad (8)$$

where λ_i is the weight of the delay penalty to sink N_i , and $\sum_{N_i \in \text{sink}(T)} \lambda_i = 1$. Note that (8) can be written as:

$$\begin{aligned} T(\mathcal{W}) = & \mathcal{K}_3 \cdot \sum_{E \in T} l_E \cdot w_E + \\ & \mathcal{K}_4 \cdot \sum_{E, E' \in T, E \neq E'} l_E \cdot l_{E'} \cdot F(E, E') \cdot \frac{w_{E'}}{w_E} + \\ & \mathcal{K}_5 \cdot \sum_{E \in T} l_E \cdot G(E) \cdot \frac{1}{w_E} \end{aligned} \quad (9)$$

where F and G are linear combinations of the f_i 's and g_i 's respectively, and F and G have *exactly* the same properties as the f_i and the g_i 's in the original single-critical-sink formulation as stated in (5) – (7). The readers may refer to [6] for details. In the remaining section, we shall use the multiple-critical-sink formulation in our wiresizing algorithms.

3 Properties of Optimal Wiresizing Solutions

In this section, we study several interesting properties of optimal wiresizing solutions, including the *separability*, the *monotone property*, and the *dominance property*. These properties are very useful in the development of the optimal wiresizing algorithms in the next section. The complete proofs of these results can be found in [6].

3.1 Separability

Theorem 1 *If the width assignment of the path from the source to a segment E is given, the optimal width assignment of each subtree branching from E can be carried out independently.*

We shall show later that the separability indeed plays a significant role in the development of our polynomial-time optimal wiresizing algorithm.

3.2 Monotone Property

Definition 1 *Given a routing tree T , a wiresizing solution \mathcal{W} on T is a monotone assignment if $w_E \geq w_{E'}$ for any pair of segments E, E' such that E is an ancestor of segment E' .*

Theorem 2 *For any given tree T , there exists a monotone optimal wire width assignment \mathcal{W}^* .*

According to the separability, the optimal wire width assignment \mathcal{W}^* can be represented by a set of “wavefronts” radiating outward from the source N_+ . Each wavefront defines the boundary where the segment width decreases. Wavefronts do

not intersect, but they may touch each others at the nodes in the tree, and all the segments enclosed between two “adjacent” wavefronts have the same width.

3.3 Dominance Property

Given two wiresizing assignments \mathcal{W} and \mathcal{W}' , we say that \mathcal{W} *dominates* \mathcal{W}' if the width assignment of any segment E in \mathcal{W} is greater than or equal to that in \mathcal{W}' . \mathcal{W} and \mathcal{W}' are said to have a *dominance relation* if either \mathcal{W} dominates \mathcal{W}' or \mathcal{W}' dominates \mathcal{W} .

Given a routing tree T , a wiresizing assignment \mathcal{W} on T , and any particular segment $E \in T$, a *local refinement* on E is the operation to optimize the width of segment E based on the objective function in (9), subject to the fixed assignment of \mathcal{W} on the other segments. With these definitions, we can derive the following theorem which is very useful in the development of a fast wiresizing algorithm in the next section.

Theorem 3 *If \mathcal{W}^* is an optimal wiresizing assignment, and \mathcal{W} has a dominance relation with \mathcal{W}^* , then the dominance relation is preserved after any number of local refinements on \mathcal{W}^* .*

The separability, the monotone property, and the dominance property were shown to be true for the optimal wiresizing solution under the upper-bound delay model in [7]. The results in this section show that these three properties are also true for the more complicated distributed Elmore delay model. In fact, these properties also hold in other general delay models as long as the F and G functions satisfy (5) – (7).

4 Wiresizing Algorithms

4.1 Optimal Wiresizing Algorithm

We first introduce the notion of a *single-stem tree* used in the following discussion. A single-stem tree is a tree with only one segment (called the *stem segment* of that tree) incident on its root. We use $SST(E)$ to denote the single-stem tree with stem E .

According to the monotone property, once E and every segment in $Ans(E)$ are assigned the appropriate widths, the optimal wire width assignment for the single-stem subtrees $SST(E_{c1}), SST(E_{c2}), \dots, SST(E_{cb})$ of the tree $SST(E)$ (with respect to the width assignment of E and segments in $Ans(E)$) can be *independently* determined, where the segments E_{c1}, \dots, E_{cb} are the children of E .

Assume we are given a single-stem tree with stem E , and a set of possible widths $\{W_1, W_2, \dots, W_r\}$, we can determine the optimal assignment \mathcal{W}^* on $SST(E)$ by enumerating all the possible width assignments of E . For each of the possible width assignment W_k of E ($1 \leq k \leq r$), we determine the optimal assignment for each single-stem subtree $SST(E_{ci})$ of $SST(E)$ independently by recursively applying the same procedure to each $SST(E_{ci})$ with $\{W_1, W_2, \dots, W_k\}$ as the set of possible widths (to guarantee the monotone property). The optimal assignment for E is the one which gives the smallest total delay.

If the original routing tree T is not a single-stem tree, however, we can decompose T into b single-stem trees, where b is the degree of the root of T , and apply the algorithm to each individual single-stem tree separately. This is called the Optimal Wiresizing Algorithm with the Elmore Delay Model (OWSA/ED).

Theorem 4 *Given a routing tree with n segments and r possible wire widths, the worst case time complexity of OWSA/ED is $O(n^r)$.*

The complexity of OWSA/ED indeed can grow exponentially with respect to r (which is usually a small constant in practice). This is the case when the tree is simply a chain of segments, where the total number of possible assignments evaluated by OWSA/ED equals to $\binom{n+r-1}{r-1} = \Omega(n^{r-1})$. Nevertheless, our optimal wiresizing algorithm is a significant improvement over the brute-force enumeration method which has complexity $O(r^n)$. In the next two subsections, we shall show how to further improve the runtime of the OWSA/ED algorithm.

4.2 Greedy Wiresizing Algorithm

In this subsection, we present the Greedy Wiresizing Algorithm with the Elmore Delay Model (GWSA/ED), which is based on an iterative refinement technique for efficient wire width assignment: Starting with an initial wire width assignment (say, all segments have the minimum width), we traverse the tree and perform local refinement on each segment whenever possible. This process is repeated until no improvement is achieved on any segment in the last round of traversal.

Despite its greedy nature, GWSA/ED performs very well in terms of the quality of assignments and runtime. Given a tree T with n segments, if we start with an assignment that is dominated by \mathcal{W}^* , say the minimum width assignment, each iteration will generate a better assignment (closer to the optimal) and still guarantee the dominance relation with \mathcal{W}^* . Therefore, GWSA/ED will converge after at most $n \cdot (r-1)$ traversals. During each traversal, each segment is locally refined exactly once, and each refinement takes $O(n)$ time. As a result, the worst case complexity of GWSA/ED is $O(n^3 \cdot r)$.

Moreover, the dominance property suggests a strategy of using the GWSA/ED algorithm to compute the lower and upper bounds of each segment width in the optimal assignment. If we start with the minimum-width assignment where each segment has the minimum wire width (and is therefore dominated by the optimal solution \mathcal{W}^*), the resulting assignment computed by GWSA/ED gives a lower bound of the optimal width for each segment, since each intermediate assignment computed by GWSA/ED, including the last one, is dominated by \mathcal{W}^* . Similarly, if we start with the maximum-width assignment, the resulting assignment computed by GWSA/ED gives an upper bound of the optimal width for each segment.

In most circumstances, we are able to obtain identical lower and upper bounds for all segments in the tree using the GWSA/ED algorithm, which immediately lead to an optimal assignment.

4.3 The Combined Approach to the Wiresizing Problem

We can further combine OWSA/ED and GWSA/ED into a new algorithm which guarantees the optimal assignment but runs extremely fast. The combined algorithm, called the Fast Optimal Wiresizing Algorithm with the Elmore Delay Model (FOWSA/ED), is described as follows:

First, we obtain the lower and upper bounds of each wire segment using the GWSA/ED algorithm. Then, we run a modified

version of OWSA/ED which only considers the assignments whose segment widths are consistent with the lower and upper bounds computed by the GWSA/ED algorithm. Since the lower and upper bounds obtained from the GWSA/ED algorithm are very close or even identical in most cases, the total number of candidate assignments ever generated by OWSA/ED algorithm is much smaller than that by the OWSA/ED algorithm alone. As a result, the upper and lower bounds obtained by GWSA/ED help to speedup the optimal algorithm significantly. Since in most cases the optimal wiresizing solutions are completely determined by the upper and lower wire width bounds computed by GWSA/ED, the runtime of FOWSA/ED is competitive with GWSA/ED while the optimality is guaranteed.

5 Experimental Results

We have implemented OWSA/ED, GWSA/ED, and the combined FOWSA/ED algorithm in ANSI C for the IBM-PC and Sun SPARC station environment. We have tested the wiresizing algorithms on both the MCM and the advanced IC technologies on signal nets of 4 and 8 sinks. The MCM and IC technology parameters are summarized in Table 1. The IC technology parameters are based on the $2\mu m$ CMOS process provided by the Orbit Semiconductor Foresight program, and the MCM technology parameters were obtained from [8].

5.1 Comparisons Between Different Wiresizing Solutions

We have compared our FOWSA/ED wiresizing solutions with the wiresizing solutions by OWSA based on the upper-bound RC delay model in [7], as well as the minimum-width solution (MIN) and the maximum-width solution (MAX). In this set of experiment, the set of wire widths allowed is $\{W_1, 2W_1, 3W_1, 4W_1\}$, where W_1 is the minimum width. Hence, every segment in MIN has width W_1 , and every segment in MAX has width $4W_1$. The placement of 100 4-sink nets and 100 8-sink nets were generated randomly, and the nets are routed by the batched 1-Steiner algorithm [12]. For each Steiner tree, a critical sink is chosen randomly. The delay to the critical sink(s) and the total wiring area of the different wiresizing solutions are compared. The signal delay is computed using the two-pole circuit simulator developed by Zhou et al. [16]. Extensive experimental results have shown that the two-pole simulator is comparable to SPICE in delay simulation, but runs much faster [16]. Table 2 summarizes the averages of the delays and areas for the 4-sink and 8-sink nets used in different wiresizing solutions, based on the IC and MCM parasitic parameters, respectively.

We can see from Table 2 that our wiresizing solution reduces interconnection delay by up to 51% when compared to the minimum-width solution of the same routing topology. Furthermore, compared to the wiresizing solution obtained by OWSA, our wiresizing solution reduces the total wiring area by up to 28% while further reducing the interconnection delays to the timing-critical sinks by up to 12%.

5.2 Effects of Multiple Critical Sinks

We have studied the effect of multiple critical sinks on the overall quality of the wiresizing solution by FOWSA/ED as compared to OWSA and the original routing solution (minimum width). Figure 2 shows the different wiresizing solutions

Technology:	Integrated Circuits (ICs)	Multi-Chip Modules (MCMs)
Driver Resistance (R_d):	156 Ω	25 Ω
Unit Wire Resistance (r_0):	0.112 $\Omega/\mu m$	0.008 $\Omega/\mu m$
Loading Capacitance (c^s):	1 fF	1000 fF
Unit Wire Capacitance (c_0):	0.039 $fF/\mu m$	0.060 $fF/\mu m$
Total Area:	5 mm x 5 mm	100 mm x 100 mm

Table 1: Technology parameters based on advanced IC and MCM designs.

IC # Sinks	Delay (ns)				Normalized Wiring Area	
	MIN	MAX	OWSA	FOWSA/ED	OWSA	FOWSA/ED
4	0.238	0.497 (+109.01%)	0.224 (-5.88%)	0.220 (-7.42%)	1.2745	1.2422
8	0.327	0.706 (+116.00%)	0.300 (-8.05%)	0.288 (-12.01%)	1.3599	1.2719
MCM # Sinks	Delay (ns)				Normalized Wiring Area	
	MIN	MAX	OWSA	FOWSA/ED	OWSA	FOWSA/ED
4	7.906	7.259 (-8.18%)	4.777 (-39.57%)	4.391 (-44.50%)	2.3677	1.8565
8	13.899	11.860 (-14.67%)	7.671 (-44.82%)	6.750 (-51.44%)	2.3762	1.7214

Table 2: Comparisons of the average delay (in nanoseconds) and normalized wiring area among different wiresizing algorithms. The normalized wiring areas for MIN and MAX are 1.0000 and 4.0000 respectively.

for a typical Steiner routing tree under the MCM technology. The width assignments of the trees are obtained by the following ways: (a) using the minimum width; (b) by OWSA based on the RC delay model [7]; (c) by FOWSA/ED with a single critical sink X; (d) by FOWSA/ED with a single critical sink Y; (e) by FOWSA/ED with two critical sinks X and Y; (f) by FOWSA/ED with all sinks being critical. The delay and the normalized wiring area are shown in Table 3.

This experiment shows that FOWSA/ED outperforms OWSA in both delay and wiring area reduction even in the presence of multiple critical sinks. Comparing Figure 2(c), Figure 2(d), and Figure 2(e), we can see that FOWSA/ED assigns wider widths only to segments along the critical path(s), and at the same time try to minimize widths of other segments. In general, when the number of critical sinks increases, delay to each sink (such as $t(X)$ and $t(Y)$) increases, but the average delay to the critical sinks (such as $\frac{t(X)+t(Y)}{2}$) decreases. Furthermore, comparison between Figure 2(b) and 2(f) reveals that OWSA, which uses the upper-bound RC delay model, indeed tends to over-size wires: even when every sink is critical, FOWSA/ED still gives smaller delay and wiring area than the OWSA solution.

6 Conclusions and Extensions

The results in this paper have shown convincingly that proper sizing of the wire segments in a routing tree can lead to significant reduction in the interconnection delay. Although the Elmore delay model is more complicated than the upper-bound model used in [7], our study has shown that optimal wiresizing under the distributed Elmore delay model can still be achieved in polynomial time very efficiently when the separability, the monotone property, and the dominance property are used to prune suboptimal wiresizing solutions. The use of the distributed Elmore delay model successfully avoids the over-sizing problem in [7] and leads to additional reduction of the delays to the timing-critical sinks.

In general, wiresizing will reduce the signal delay but in-

crease the wiring area. In many cases, we want to minimize both the interconnection delay and the routing area, especially when the global routing solution is very dense and mutual capacitance and inductance cannot be ignored. Therefore, we want to explore the *tradeoff* between area and delay. This tradeoff can be formulated as finding a wiresizing assignment \mathcal{W} which minimizes a weighted sum of the delay and the total wiring area, i. e. $\alpha \cdot delay + \beta \cdot area$, where α and β are constants. We have shown in [6] that this delay-area tradeoff formulation can be reduced to the general formulation in (9), and therefore can be handled by our wiresizing algorithms.

In practice, it is possible that the unit wire resistances and/or unit wire capacitances for the vertical and horizontal segments are different, because vertical and horizontal wires are usually routed on two different layers. We have proved that the *separability*, the *dominance property*, and a restricted version of the *monotone property* still hold under this model. Based on these properties, we can construct a more general polynomial-time optimal wiresizing algorithm which handles the direction-dependent wire resistances and wire capacitances efficiently using a similar methodology.

Allowing variable-width routing considerably complicates the global and detailed routing steps in VLSI layout design. The next focus of our research is to develop efficient channel routers and general area routers which can generate compact routing solutions of non-uniform wire width. Furthermore, our current wiresizing algorithm optimizes each routing tree independently. It will be of practical interest to develop efficient wiresizing algorithms and detailed routers which can take into consideration the mutual capacitance and inductance between different routing trees.

References

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

Comparison	MIN	OWSA	FOWSA/ED (X critical)	FOWSA/ED (Y critical)	FOWSA/ED (X, Y critical)	FOWSA/ED (all critical)
$t(X)$ (ns)	9.679	5.824	* 4.560	6.310	* 4.945	* 5.008
$t(Y)$ (ns)	12.117	6.773	8.116	* 6.141	* 6.651	* 6.715
$\frac{t(X)+t(Y)}{2}$ (ns)	10.898	6.299	6.338	6.226	5.798	5.862
$t(ALL)$ (ns)	8.962	5.611	5.762	5.638	5.331	5.222
Normalized Wiring Area	1.0000	2.3267	1.4554	1.7525	1.5050	1.5941

Table 3: Comparisons (delay and normalized wiring area) among wiresizing assignments obtained by different algorithms (with MCM technology). Numbers with an asterisk (*) are delays to critical sinks, $t(X)$ and $t(Y)$ are the delays at X and Y respectively, and $t(ALL)$ is the average delay to all sinks.

- [2] K. D. Boese, A. B. Kahng, and G. Robins, "High-Performance Routing Trees With Identified Critical Sinks", *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 182-187.
- [3] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, "Toward Optimal Routing Trees", *Proc. 4th ACM/SIGDA Physical Design Workshop*, 1993.
- [4] J. P. Cohoon and L. J. Randall, "Critical Net Routing", *Proc. IEEE Int'l. Conf. on Computer Design*, 1991, pp. 174-177.
- [5] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, "Provably Good Performance-Driven Global Routing", *IEEE Trans. on CAD*, 11(6), June 1992, pp. 739-752.
- [6] J. Cong and K. S. Leung, "Optimal Wiresizing Under the Distributed Elmore Delay Model", *UCLA Computer Science Tech. Report CSD-930012*, Los Angeles, CA 90024, April 1993.
- [7] J. Cong, K. S. Leung, and D. Zhou, "Performance-Driven Interconnect Design Based on Distributed RC Delay Model", *Proc. ACM/IEEE Design Automation Conf.*, 1993, pp. 606-611.
- [8] W. Dai, private communication, 1992.
- [9] W. E. Donath, R. J. Norman, B. K. Agrawal, and S. E. Bello, "Timing Driven Placement Using Complete Path Delays", *Proc. ACM/IEEE Design Automation Conf.*, 1990, pp. 84-89.
- [10] W. C. Elmore, "The Transient Response of Damped Linear Network with Particular Regard to Wideband Amplifier", *J. Applied Physics*, 19(1948), pp. 55-63.
- [11] A. L. Fisher, and H. T. Kung, "Synchronizing Large Systolic Arrays", *Proc. SPIE* 341, May 1982, pp. 44-52.
- [12] A. B. Kahng, and G. Robins, "A New Class of Iterative Steiner Tree Heuristics with Good Performance", *Proc. IEEE Int'l. Conf. on Computer-Aided Design*, July 1992, pp. 893-902.
- [13] J. Rubinstein, P. Penfield, and M. A. Horowitz, "Signal Delay in RC Tree Networks", *IEEE Trans. on CAD*, 2(3) (1983) pp. 202-211.
- [14] S. Sutanthavibul and E. Shragowitz, "An Adaptive Timing-Driven Layout for High Speed VLSI", *Proc. ACM/IEEE Design Automation Conf.*, 1990, pp. 90-95.
- [15] D. Zhou, F. P. Preparata, and S. M. Kang, "Interconnection Delay in Very High-speed VLSI", *IEEE Trans. on Circuits and Systems* 38(7), 1991.
- [16] D. Zhou, S. Su, F. Tsui, D. S. Gao, and J. Cong, "A Distributive RCL-Model for MCM Layout", *Proc. of IEEE Multichip Module Conf.*, March 1993, pp. 191-197.

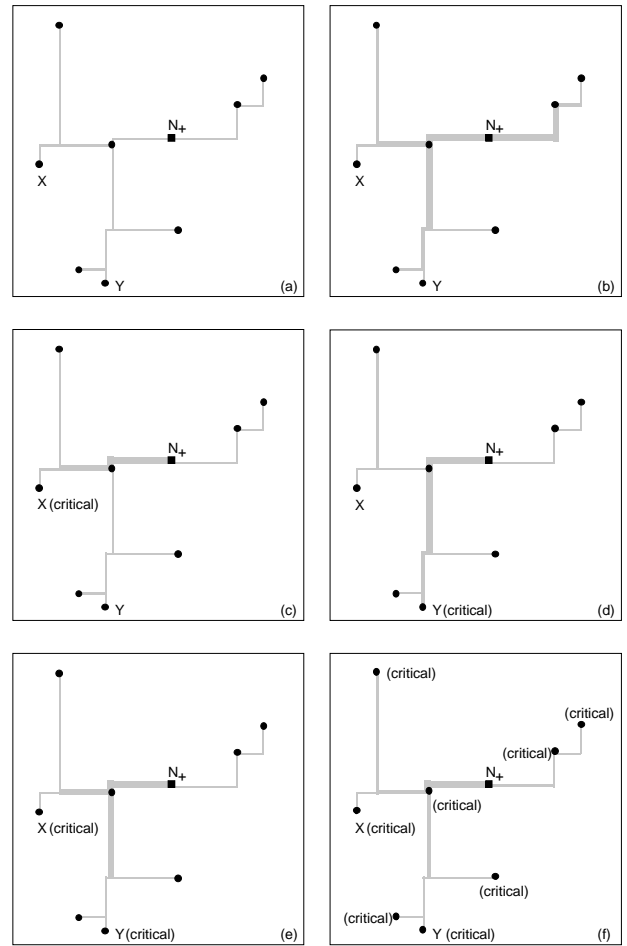


Figure 2: Different wiresizing solutions obtained using the following algorithms: (a) the minimum width; (b) by OWSA based on the RC delay model; (c) by FOWSA/ED with a single critical sink X; (d) by FOWSA/ED with a single critical sink Y; (e) by FOWSA/ED with two critical sinks X and Y; (f) by FOWSA/ED with all sinks being critical.