

A Fast Algorithm for Optimal Wire-Sizing Under Elmore Delay Model *

Chung-Ping Chen and D. F. Wong
Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

Abstract

In this paper, we present a fast algorithm for continuous wire-sizing under the distributed Elmore delay model. Our algorithm GWSA-C is an extension of the GWSA algorithm (for discrete wire-sizing) in [CL93a]. GWSA-C is an iterative algorithm with guaranteed convergence to optimal wire-sizing solutions. When specialized to discrete wire-sizing problems, GWSA-C is an improved implementation of GWSA, both in terms of runtime and memory usage. GWSA-C is extremely fast even on very large problems. For example, we can solve a 1,000,000 wire-segment problem in less than 3 minutes runtime on an IBM RS/6000 workstation. This kind of efficiency in wire-sizing has never been reported in the literature.

1 Introduction

Since interconnect delay has become the dominant factor of circuit delay in deep sub-micron fabrication technology, it is clear that sizing the gates alone is not enough to achieve desirable circuit performance. As a result, wire-sizing for performance optimization is currently a very active research area ([CL93a] [ZDX93] [MBP95] [SP94]).

The wire-sizing optimization problem was first introduced in [CL93a] [CL93b]. The authors in [CL93a] use the monotonicity, separability, and dominance properties under the Elmore delay model to determine an optimal solution for the discrete wire-sizing problem. (By discrete wire-sizing, we mean that each wire has only a finite number of possible widths.) Most of the other wire-sizing algorithms in the literature (e.g. [ZDX93] [MBP95] [SP94]) are designed for continuous wire-sizing (i.e., each wire size can be any real number within a given range), but none of them are efficient when the number of segments is in the thousands. (Note that in practice it is possible to have "large" problems. For example, a 3000-pin clock tree has about 6,000 tree branches; if each branch has two wire segments, we have more than 10,000 wire segments!)

In this paper, we present a fast algorithm GWSA-C for continuous wire-sizing under the distributed Elmore delay model. GWSA-C is an extension of the GWSA algorithm (for discrete wire-sizing) in [CL93a]. Basically, GWSA-C is a greedy algorithm based on iteratively re-sizing the wire segments. In each iteration, the wire segments are examined one at a time; each time a wire segment is re-sized optimally

while keeping the widths of the other segments fixed. The memory requirement by GWSA-C is $O(n)$ and each iteration in re-sizing all wire segments (once per wire segment) takes $O(n)$ time, where n is the number of wire segments. GWSA-C is guaranteed to converge to optimal wire-sizing solutions. When specialized to discrete wire-sizing problems, GWSA-C is an improved implementation of GWSA, both in terms of runtime and memory usage. GWSA-C is extremely fast even on very large problems. For example, we can solve a 1,000,000 wire-segment problem in less than 3 minutes runtime on an IBM RS/6000 workstation. This kind of efficiency in wire-sizing has never been reported in the literature.

The remainder of this paper is organized as follows: In Section 2, we present the formulation of the wire-sizing problem under the Elmore delay model. In Section 3, we present our wire-sizing algorithm GWSA-C. Experimental results are presented in Section 4.

2 Problem Formulation

Given a routing tree T consisting of a driver w_0 (at the root) with resistance r_0 , a set of s sinks $\{N_1, N_2, \dots, N_s\}$ with capacitance load c_i^s , $1 \leq i \leq s$, and n wire segments $\{w_1, w_2, \dots, w_n\}$. Also given is a set of non-negative weights $\{\lambda_1, \lambda_2, \dots, \lambda_s\}$ with $\sum_{i=1}^s \lambda_i = 1$, where λ_i is a weight associated with sink N_i representing the criticality of the sink. Let l_i be the length of w_i and x_i be the wire size of w_i , $1 \leq i \leq n$. $\mathbf{x} = (x_1, x_2, \dots, x_n)$ which will be referred to as a wire-sizing solution. We are also given U_i and L_i as the respective upper bound and lower bound of the wire size of w_i , i.e. $L_i \leq x_i \leq U_i$, $1 \leq i \leq n$. Let $dec(w_i)$ be the set of descendants of w_i (excluding w_i). Let $ans(w_i)$ be the set of ancestors of w_i (excluding w_i and including w_0). The resistance and capacitance of wire segments w_i is denoted by r_i and c_i , respectively. Let $c_0 = 0$. We use μ_i to denote $\sum_{N_j \in dec(w_i)} \lambda_j$, $1 \leq i \leq n$. Let $R_i = \sum_{w_j \in ans(w_i)} \mu_j r_j$ and $C_i = \sum_{w_j \in dec(w_i)} c_j + \sum_{N_j \in dec(w_i)} c_j^s$, $1 \leq i \leq n$. R_i and C_i will be referred to as the weighted upstream resistance and the downstream capacitance, respectively, of w_i .

We use the π -model to model each wire segment w_i (see Figure 1). We have the resistance of w_i is given by $r_i = \alpha l_i / x_i$, the capacitance of w_i , is given by $c_i = \beta x_i l_i / 2$, where α is the resistance per unit length at unit width and β is the capacitance per unit square. We use the Elmore delay model [E48] to estimate the delay. The signal delay at sink N_i is

*This work was partially supported by the Texas Advanced Research Program under Grant No. 003658459.

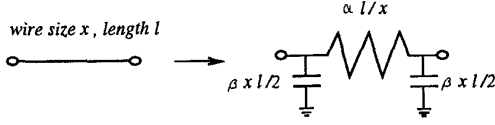


Figure 1: RC model for wire segment

computed by:

$$D_i = \sum_{w_j \in P_i} r_j (C_j + \frac{C_j}{2}), \quad (1)$$

where P_i is the set of wire segments that lie on the path from the root to sink N_i .

In this paper, we consider the following wire-sizing optimization problem:

$$\begin{aligned} \text{Minimize} \quad & F(\mathbf{x}) = \sum_{i=1}^s \lambda_i D_i \\ \text{Subject to} \quad & L_j \leq x_j \leq U_j, \quad 1 \leq j \leq n. \end{aligned}$$

3 Wire-sizing Algorithm

Lemma 1 For each x_i , $F(\mathbf{x})$ can be written in the following form

$$F(\mathbf{x}) = A_i(\mathbf{x})x_i + \frac{B_i(\mathbf{x})}{x_i} + E_i(\mathbf{x}),$$

where $A_i(\mathbf{x})$, $B_i(\mathbf{x})$, and $E_i(\mathbf{x})$ are independent of x_i , $A_i(\mathbf{x}) = \beta l_i R_i$, and $B_i(\mathbf{x}) = \alpha l_i \mu_i C_i$.

Proof:

$$\begin{aligned} F(\mathbf{x}) &= \sum_{j=1}^s \lambda_j D_j \\ &= \sum_{j=1}^s \lambda_j \sum_{w_k \in P_j} r_k (C_k + \frac{C_k}{2}) \\ &= \sum_{w_k \in T} r_k (C_k + \frac{C_k}{2}) \sum_{N_j \in \text{dec}(w_k)} \lambda_j \\ &= \sum_{w_k \in T} r_k \mu_k (C_k + \frac{C_k}{2}). \end{aligned}$$

Note that the terms that involve x_i come from $\sum_{w_k \in \text{ans}(w_i)} r_k \mu_k C_k$. In fact, only the term $\beta l_i x_i$ (the wire capacitance of w_i) in C_k contribute to the terms with x_i , hence

$$A_i(\mathbf{x}) = \beta l_i \sum_{w_j \in \text{ans}(w_i)} r_j \mu_j.$$

Since the terms that involve $\frac{1}{x_i}$ only come from $r_i \mu_i C_i = \frac{\alpha l_i}{x_i} \mu_i C_i$, we have $B_i(\mathbf{x}) = \alpha l_i \mu_i C_i$. Let $E_i(\mathbf{x}) = F(\mathbf{x}) -$

$A_i(\mathbf{x})x_i - \frac{B_i(\mathbf{x})}{x_i}$. It is clear that $A_i(\mathbf{x})$, $B_i(\mathbf{x})$, and $E_i(\mathbf{x})$ are independent of x_i \square

If we re-size w_i while keeping the wire-sizes of all the other wire segments fixed, we say that it is a local re-sizing of w_i . An optimal local re-sizing of w_i is a local re-sizing that minimize $F(\mathbf{x})$.

Lemma 2 Let $\hat{\mathbf{x}} = (x_1, x_2, \dots, x_n)$ be a wire-sizing solution. An optimal local re-sizing of w_i is given by changing the width of w_i to

$$x_i^* = \min \left(U_i, \max \left(L_i, \sqrt{\frac{B_i(\hat{\mathbf{x}})}{A_i(\hat{\mathbf{x}})}} \right) \right).$$

Proof: Suppose we change the wire-size of w_i to x_i . It follows from Lemma 1 that the new delay cost is given by

$$G(x_i) = A_i(\hat{\mathbf{x}})x_i + \frac{B_i(\hat{\mathbf{x}})}{x_i} + E_i(\hat{\mathbf{x}})$$

Differentiating G with respect to x_i , we get

$$\frac{dG}{dx_i} = A_i(\hat{\mathbf{x}}) - \frac{B_i(\hat{\mathbf{x}})}{x_i^2}$$

Let $\theta(\hat{\mathbf{x}}) = \sqrt{\frac{B_i(\hat{\mathbf{x}})}{A_i(\hat{\mathbf{x}})}}$. Note that

$$\begin{cases} \frac{dG}{dx_i} = 0, & x_i = \theta(\hat{\mathbf{x}}), \\ \frac{dG}{dx_i} < 0, & x_i < \theta(\hat{\mathbf{x}}), \\ \frac{dG}{dx_i} > 0, & x_i > \theta(\hat{\mathbf{x}}). \end{cases}$$

Hence $G(x_i)$ is decreasing when $x_i < \theta(\hat{\mathbf{x}})$, $G(x_i)$ is increasing when $x_i > \theta(\hat{\mathbf{x}})$, and $G(x_i)$ is minimum at $x_i = \theta(\hat{\mathbf{x}})$.

Let x_i^* be the wire-size of w_i in an optimal local re-sizing of the wire segments. We consider three cases.

Case 1: $\theta(\hat{\mathbf{x}}) \in [L_i, U_i]$.

In this case, we have $x_i^* = \theta(\hat{\mathbf{x}})$.

Case 2: $\theta(\hat{\mathbf{x}}) > U_i$.

Since $G(x_i)$ is decreasing on $[L_i, U_i]$, we have $x_i^* = U_i$.

Case 3: $\theta(\hat{\mathbf{x}}) < L_i$.

Since $G(x_i)$ is increasing on $[L_i, U_i]$, we have $x_i^* = L_i$.

From the above discussion, we have

$$x_i^* = \min \left(U_i, \max \left(L_i, \sqrt{\frac{B_i(\hat{\mathbf{x}})}{A_i(\hat{\mathbf{x}})}} \right) \right) \square$$

Theorem 1 $\mathbf{x}^* = (x_1^*, \dots, x_n^*)$ is an optimal wire-sizing solution if and only if

$$x_i^* = \min \left(U_i, \max \left(L_i, \sqrt{\frac{B_i(\mathbf{x}^*)}{A_i(\mathbf{x}^*)}} \right) \right), \quad 1 \leq i \leq n.$$

Proof:

(\Rightarrow) This follows directly from Lemma 2 and the fact that optimal local re-sizing of w_i would not change its width.

(\Leftarrow) If $x_i^* = \min \left(U_i, \max \left(L_i, \sqrt{\frac{B_i(\mathbf{x}^*)}{A_i(\mathbf{x}^*)}} \right) \right)$, we show that \mathbf{x}^* is a global minimum point of $F(\mathbf{x})$ over $\Omega_x = \{\mathbf{x} | L_i \leq x_i \leq U_i, 1 \leq i \leq n\}$; i.e. \mathbf{x}^* is an optimal wire-sizing solution. Note that $F(\mathbf{x})$ is a posynomial in \mathbf{x} , and that under the transformation $x_i = e^{z_i}$, $1 \leq i \leq n$, $G(\mathbf{z}) = F(e^{z_1}, \dots, e^{z_n})$ is convex over $\Omega_z = \{\mathbf{z} | L_i \leq e^{z_i} \leq U_i, 1 \leq i \leq n\}$ (see [DPZ67]). Let $\mathbf{z}^* = (z_1^*, \dots, z_n^*)$ where $x_i^* = e^{z_i^*}$, $1 \leq i \leq n$. We now consider three cases:

Case 1: $x_i^* = \sqrt{\frac{B_i(\mathbf{x}^*)}{A_i(\mathbf{x}^*)}}$.

In this case, we have $\frac{\partial F}{\partial x_i}(\mathbf{x}^*) = 0$. Thus

$$\frac{\partial G}{\partial z_i}(\mathbf{z}^*) = \frac{\partial F}{\partial x_i}(\mathbf{x}^*) \frac{\partial x_i}{\partial z_i}(\mathbf{z}^*) = \frac{\partial F}{\partial x_i}(\mathbf{x}^*) e^{z_i^*} = 0.$$

Case 2: $x_i^* = L_i$.

In this case, $L_i \geq \sqrt{\frac{B_i(\mathbf{x}^*)}{A_i(\mathbf{x}^*)}}$. We have $\frac{\partial F}{\partial x_i} \geq 0$, and $z_i - z_i^* \geq 0, \forall z \in \Omega_z$. Hence

$$\frac{\partial G}{\partial z_i}(\mathbf{z}^*)(z_i - z_i^*) = \frac{\partial F}{\partial x_i}(\mathbf{x}^*) e^{z_i^*} (z_i - z_i^*) \geq 0, \forall z \in \Omega_z.$$

Case 3: $x_i^* = U_i$.

In this case, $U_i \leq \sqrt{\frac{B_i(\mathbf{x}^*)}{A_i(\mathbf{x}^*)}}$. We have $\frac{\partial F}{\partial x_i}(\mathbf{x}^*) \leq 0$ and $z_i - z_i^* \leq 0, \forall z \in \Omega_z$. Hence

$$\frac{\partial G}{\partial z_i}(\mathbf{z}^*)(z_i - z_i^*) = \frac{\partial F}{\partial x_i}(\mathbf{x}^*) e^{z_i^*} (z_i - z_i^*) \geq 0, \forall z \in \Omega_z.$$

Combining the three cases, we get $\frac{\partial G}{\partial z_i}(\mathbf{z}^*)(z_i - z_i^*) \geq 0, \forall z \in \Omega_z$. Since G is convex (see [L84]), we have, for all $\mathbf{z} \in \Omega_z$,

$$\begin{aligned} G(\mathbf{z}) - G(\mathbf{z}^*) &\geq \nabla G(\mathbf{z}^*)(\mathbf{z} - \mathbf{z}^*) \\ &= \sum_{i=1}^n \frac{\partial G}{\partial z_i}(\mathbf{z}^*)(z_i - z_i^*) \\ &\geq 0. \end{aligned}$$

Thus $F(\mathbf{x}) - F(\mathbf{x}^*) = G(\mathbf{z}) - G(\mathbf{z}^*) \geq 0, \forall \mathbf{x} \in \Omega_x$. Therefore \mathbf{x}^* is a global minimum point. \square

We now describe our algorithm GWSA-C. Basically, GWSA-C is a greedy algorithm based on iteratively re-sizing the wire segments. In each iteration, the wire segments are examined one at a time; each time a wire segment is re-sized optimally while keeping the sizes of the other segments fixed. Lemma 2 gives the formula to compute the new size of a wire segment. Note that we can pre-compute the set of all edge weights μ_i 's in $O(n)$ time by a bottom-up traversal of the tree. The computation of each μ_i is based on previously computed μ_j 's, using the formula $\mu_i = \sum_{w_j \in \text{children}(w_i)} \mu_j$ where $\text{children}(w_i)$ is the set of the children of w_i . At the beginning of each iteration, we compute all downstream capacitances C_i 's by a bottom-up traversal of T in $O(n)$ time. The formula for computing C_i is $C_i = \sum_{w_j \in \text{children}(w_i)} (C_j + c_j)$. Each iteration of GWSA-C consists of a top-down traversal

of T ; each time we visit a wire segment w_i , we re-size w_i according to the formula in Lemma 2 where we need $A_i(\mathbf{x})$ and $B_i(\mathbf{x})$. Observe that $R_i = R_{p_i} + \mu_{p_i} r_{p_i}$, where w_{p_i} is the parent of w_i . Thus, $A_i(\mathbf{x}) = A_{p_i}(\mathbf{x}) + \beta l_i \mu_{p_i} r_{p_i}$. Since C_i is independent of the wire sizes of the ancestors of w_i , there is no need to update C_i during the top-down traversal of T . We have, $B_i(\mathbf{x}) = \alpha l_i \mu_i C_i$. Clearly, each iteration of re-sizing the wire segments takes $O(n)$ time. It is also easy to see that the memory usage by GWSA-C is also $O(n)$. We summarize the description of GWSA-C in Figure 2. Since we terminate the program when no further improvement is possible, it follows from Theorem 2 that GWSA-C converges to an optimal wire-sizing solution. We have the following theorem.

Theorem 2 *Algorithm GWSA-C converges to an optimal wire-sizing solution. It runs in $O(rn)$ time using $O(n)$ storage, where n is the number of wire segments and r is the number of iterations.*

Algorithm: GWSA-C

S1. $x_i := L_i, 1 \leq i \leq n$;

S2. Compute all μ_i 's by a bottom-up traversal of T using the following formula:

$$\mu_i := \sum_{w_j \in \text{children}(w_i)} \mu_j$$

S3. Compute all C_i 's by a bottom-up traversal of T using the following formula:

$$C_i := \sum_{w_j \in \text{children}(w_i)} (C_j + c_j)$$

S4. Perform a top-down traversal of T :
 For each w_i ,

$$A_i(\mathbf{x}) := A_{p_i}(\mathbf{x}) + \beta l_i \mu_{p_i} r_{p_i}$$

$$B_i(\mathbf{x}) := \alpha l_i \mu_i C_i$$

$$x_i = \min \left(U_i, \max \left(L_i, \sqrt{\frac{B_i(\mathbf{x})}{A_i(\mathbf{x})}} \right) \right);$$

S5. Repeat **S3**–**S4** until no improvement.

Figure 2: The GWSA-C algorithm.

Remark GWSA-C can be specialized to solve the discrete wire-sizing problem. Let $s_1 \leq s_2 \leq \dots \leq s_k$ be the set of feasible wire sizes for w_i . We modify GWSA-C as follows. In **S4**, after computing x_i , we let s_j and s_{j+1} be the two consecutive feasible wire sizes such that $s_j \leq x_i \leq s_{j+1}$. It is not hard to see that the optimal local re-sizing of w_i is obtained by checking the delay costs $G(s_j)$ and $G(s_{j+1})$ and pick the smaller one. For discrete wire-sizing problems, GWSA-C is an improved implementation of GWSA in [CL93a], both in terms of runtime and memory usage.

4 Experimental Results

We implemented and tested our algorithm on an IBM/RS6000 workstation. We first compare the runtime

of GWSA-C with GWSA in [CL93a] on a single wire with 100 to 10^6 segments. The GWSA package we used was obtained from the authors of [CL93a] in July 1995. The parameters used are shown in Table 1. Table 2 shows the runtime and memory usage comparisons between GWSA and GWSA-C. It shows that GWSA-C runs much more efficiently than GWSA, and that the number of iterations grows very slowly. Due to the requirement of large memory usage, GWSA algorithm could not get any results when the number of wires exceeded 2000. Figures 3 and 4 show that the runtime and memory usage of GWSA-C are close to linear. Table 3 shows that GWSA-C improved the delay cost in the clock tree r1-r5 [T93] by up to 89%. Table 4 shows that GWSA-C produced near-optimal results for discrete wire-sizing problems. The lower bound was obtained by running the continuous version of GWSA-C.

Unit Resistance:	0.003 $\Omega/\mu\text{m}$
Unit Capacitance:	2×10^{-17} $F/\mu\text{m}$
Minimum Wire Width:	1 μm
Maximum Wire Width:	20 μm

Table 1: RC Parameters.

# of wire segments	GWSA [CL93a]		GWSA-C		No of Iters
	Time	Memory	Time	Memory	
100	0.15	1968	0.01	32	9
200	0.76	2408	0.02	32	10
1000	22.50	17464	0.08	40	10
2000	106.68	64500	0.17	48	12
10000	-	-	1.12	108	15
20000	-	-	2.35	188	16
100000	-	-	13.52	812	18
1000000	-	-	141.42	7884	19

Table 2: Runtime and memory usage comparisons between GWSA and GWSA-C. (Unit: Runtime(second), Memory(kbytes), Step Width($1\mu\text{m}$))

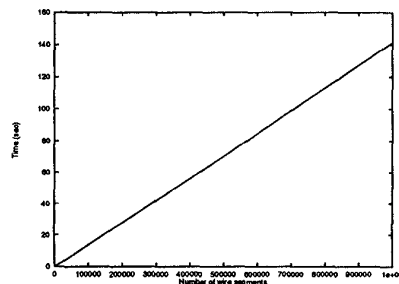


Figure 3: Runtime of GWSA-C.

Circuit Name	No of Wire Segments	Delay (ns)		Improve%
		Initial	Final	
r1	533	0.77	0.12	84.42
r2	1195	2.11	0.27	87.20
r3	1723	3.38	0.43	87.28
r4	3805	9.09	1.03	88.67
r5	6201	15.86	1.65	89.04

Table 3: Result Comparisons between with and without wire-sizing by GWSA-C.

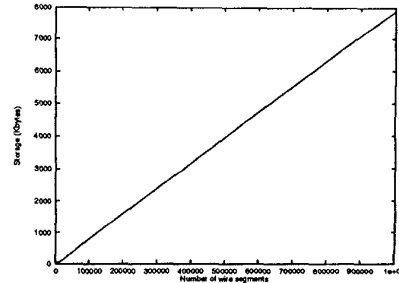


Figure 4: Memory Usage of GWSA-C.

Wire Length	Delay (ps)			Ratio
	Initial	GWSA-C	lower bound	
100	0.52	0.12	0.11	1.09
200	2.66	0.47	0.47	1.00
500	15.72	3.01	2.99	1.01
1000	63.45	12.07	11.98	1.01
10000	6051.50	1209.54	1200.73	1.01
50000	150365.34	30244.42	30024.13	1.01

Table 4: Results for GWSA-C for discrete wire sizes.

(Unit: wire length(μm))

5 Acknowledgment

We thank Jason Cong and K. L. Leung for providing the GWSA source code.

References

- [CL93a] Jason Jingsheng Cong, Kwok-Shing Leung. "Optimal wire sizing under the distributed elmore delay model," *IEEE Intl. Conf. on Computer-Aided Design*, July 1993.
- [CL93b] Jason Jingsheng Cong, Kwok-Shing Leung, Zian Zhou. "Performance-Driven Interconnect design based on distributed RC delay model." *30th ACM/IEEE Design Automation Conference*, June 1993.
- [DPZ67] R. J. Duffin, E. L. Peterson, Clarence Zener, *Geometric programming-theory and application*, John Wiley & Sons, Inc. 1967.
- [E48] W. C. Elmore "The transient response of damped linear networks with particular regard to wide band amplifiers" *J. Applied Physics*, 19(1), 1948.
- [L84] D. G. Luenberger, *Linear and Nonlinear Programming* Addison-Wesley Publishing Company. Inc. , 1984.
- [MBP95] N. Menezes, Ross Baldick and L. T. Pillage, "A sequential quadratic programming approach to concurrent gate and wire sizing" *Proc. ACM/IEEE Intl. Conf. on Computer-Aided Design*, 1995.
- [SP94] Sachin S. Sapatnekar "RC Interconnect Optimization under the Elmore Delay Model" *31st ACM/IEEE Design Automation Conference*, 1994.
- [T93] R. S. Tsay, "An exact zero-skew clock routing algorithm" *IEEE Trans. Computer-Aided Design*, vol. 12, no. 2, pp.242-249, February 1993.
- [ZDX93] Qing Zhu, Wayne W.-M. Dai, Joe G. Xi. "Optimal sizing of high-speed clock networks based on distributed rc and lossy transmission line models," *Proc. IEEE Intl. Conference on Computer-Aided Design*, pp.628-633, 1993.