[15] 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.

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

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

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

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

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

References

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

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

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

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

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

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

[8] 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.

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

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

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

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

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

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

if E is right to E', $F(E_1, E') = F(E_2, E') = \cdots = F(E, E') = F_r(S)$ and $F(E', E_1) = F(E', E_2) = \cdots = F(E', E) = F_l(S)$.

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

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

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

Again assuming edge E is in segment S, according to Lemma 3

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

According to Lemma 7 and the equation (11) for the local refinement, Lemma 8 holds.

Lemma 8 *When given the wiresizing solution $\mathcal{W}$ and any particular edge E, the local refinement result for E with respect to $\mathcal{W}$ is independent of the edge-division for wires other than E.*

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

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

For a wiresizing solution $\mathcal{W}$ which dominates the optimal solution $\mathcal{W}^*$, let $\mathcal{W}^b$ be the wiresizing solution after performing an BRU operation of $\mathcal{W}$ on E under $\mathcal{E}$. Assume again $F_l(S) \geq F_r(S)$, we have $w_l^b = w_{F2}^b = w_{F3}^b = ... = w_r^b$ according to the definition of the BRU operation. Meanwhile, let $\mathcal{W}^*$ be the optimal solution. We have $w_l^* \geq w_{F2}^* \geq w_{F3}^* \geq ... \geq w_r^*$ according to the *local monotone property*.

According to Lemma 8, w_l^b is also the local refinement result for edge E_l under $\mathcal{E}_F$ such that $w_l^b \geq w_l^*$. In conclusion, $w_l^b = w_{F2}^b = w_{F3}^b = ... = w_r^b \geq w_l^* \geq w_{F2}^* \geq w_{F3}^* \geq ... \geq w_r^*$. Recall that the bundled refinement of edge E does not change the wire width for any edge E' other than $E_l, E_{F2}, E_{F3}, ..., E_r$, so $\mathcal{W}^b$ still dominates $\mathcal{W}^*$.

The BRL case can be shown true in a similar way. □

F. Correctness of OWBR Algorithm

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

Proof: If $\mathcal{E}$ is the wiresizing edge-division after OWBR, for any edge E under $\mathcal{E}$, l_E is longer than δ if and only if E is an *convergent* edge, whose bound can not be tightened any more.

If E is of length δ, the lower and upper bounds on it are same as those given by a local refinement operation under $\mathcal{E}_F$, as discussed in the proof of Theorem 5. Therefore, Theorem 6 holds. □

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

$$\Psi(\mathcal{W},\overline{E}) \ + \ \Phi(\mathcal{W},\overline{E})\cdot\tilde{w}_E \ + \ \Theta(\mathcal{W},\overline{E})\cdot\frac{1}{\tilde{w}_E} \ \leq \ \Psi(\mathcal{W},\overline{E}) \ + \ \Phi(\mathcal{W},\overline{E})\cdot w_E^* \ + \ \Theta(\mathcal{W},\overline{E})\cdot\frac{1}{w_E^*} \qquad (12)$$

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

$$\Psi(\mathcal{W}^*,\overline{E}) \ + \ \Phi(\mathcal{W}^*,\overline{E})\cdot w_E^* \ + \ \Theta(\mathcal{W}^*,\overline{E})\cdot\frac{1}{w_E^*} \ \leq \ \Psi(\mathcal{W}^*,\overline{E}) \ + \ \Phi(\mathcal{W}^*,\overline{E})\cdot\tilde{w_E} \ + \ \Theta(\mathcal{W}^*,\overline{E})\cdot\frac{1}{\tilde{w}_E} \quad (13)$$

Summing up (12) and (13), we obtain:

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

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

E Bundled Refinement Property

Lemma 7 *Given an MSIT, an edge-division $\mathcal{E}$ and a wiresizing solution $\mathcal{W}$, for any particular edge E under $\mathcal{E}$, if we divide E into a sequence of edges $E_1, E_2, \cdots$, and E_k, each inherits the wire width assignment of E, and call the resulting edge-division and wiresizing solution $\mathcal{E}'$ and $\mathcal{W}'$, respectively, then we have the following relationships for any edge E' other than E with respect to Functions (8)-(10):*

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

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

$\Theta(MSIT,\mathcal{E},E',\mathcal{W})=\Theta(MSIT,\mathcal{E}',E',\mathcal{W}')$ [6]

Proof: Lemma 7 is true if the follows are true:

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

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

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

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

Assuming edge E is in segment S, There are two cases for edge E':

Case 1: E' is in S, according to Lemma 1

if E is left to E', $F(E_1,E')=F(E_2,E')=\cdots=F(E,E')=F_l(S)$ and $F(E',E_1)=F(E',E_2)=\cdots=F(E',E)=F_r(S)$.

[6] In general, under the modeling method used in [10, 6, 4, 16], for the elmore delay t^{ij} between source N_i and sink N_j, we have $t^{ij}(\mathcal{E},\mathcal{W})=t^{ij}(\mathcal{E}',\mathcal{W}')$. I.e., the Elmore delay is independent of the edge-division when given the wiresizing solution.

$F(E_r, E_l)$ if $F_l(S) > F_r(S)$, and $w^*_{E_l} \leq w^*_{E_r}$ if $F(E_l, E_r) < F(E_r, E_l)$ if $F_l(S) < F_r(S)$. Thus, the width assignment for edges within S is decreased rightward if $F_l(S) > F_r(S)$, and it is increased rightward if $F_l(S) < F_r(S)$. □

We assume that there exists an optimal wiresizing solution where the width assignment within the segment S is uniform when $F_l(S) = F_r(S)$. Then, we have the *SST local monotone property*:

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

D. Dominant Property

Theorem 4 *If a wire width assignment $\mathcal{W}$ dominates $\mathcal{W}^*$ for an MSIT, then any* local refinement *of $\mathcal{W}$ still dominates $\mathcal{W}^*$. Similarly, if a wire width assignment $\mathcal{W}$ is dominated by $\mathcal{W}^*$, then any* local refinement *of $\mathcal{W}$ is dominated by $\mathcal{W}^*$.*

Proof: Similar to the proof of the dominance property in [10], we define the following equations for any particular edge E:

$$\begin{aligned}
\Psi(MSIT, \mathcal{E}, E, \mathcal{W}) &= \mathcal{K}_1 \cdot \sum_{E' \in MSIT - \{E\}} w_{E'} + \mathcal{K}_2 \cdot \sum_{E', E'' \in MSIT - \{E\}, E' \neq E''} F(E', E'') \cdot \frac{w_{E''}}{w_{E'}} + \\
&\quad \mathcal{K}_3 \cdot \sum_{E', E'' \in MSIT - \{E\}, E' \neq E''} F(E', E'') \cdot \frac{1}{w_{E'}} + \\
&\quad \mathcal{K}_4 \cdot \sum_{E' \in MSIT - \{E\}} G(E') \cdot \frac{1}{w_{E'}} + \mathcal{K}_5 \cdot \sum_{E' \in MSIT - \{E\}} H(E') \cdot \frac{1}{w_{E'}} \qquad (8) \\
\Phi(MSIT, \mathcal{E}, E, \mathcal{W}) &= \mathcal{K}_1 + \mathcal{K}_2 \cdot \sum_{E' \in MSIT - \{E\}} F(E', E) \cdot \frac{1}{w_{E'}} \qquad (9) \\
\Theta(MSIT, \mathcal{E}, E, \mathcal{W}) &= \mathcal{K}_2 \cdot \sum_{E' \in MSIT - \{E\}} F(E, E') \cdot w_{E'} + \mathcal{K}_3 \cdot \sum_{E' \in MSIT - \{E\}} F(E, E') \\
&\quad + \mathcal{K}_4 \cdot G(E) + \mathcal{K}_5 \cdot H(E) \qquad (10)
\end{aligned}$$

We can then rewrite the objective function (3) as follows:

$$t(MSIT, \mathcal{E}, \mathcal{W}) = \Psi(MSIT, \mathcal{E}, E, \mathcal{W}) + \Phi(MSIT, \mathcal{E}, E, \mathcal{W}) \cdot w_E + \Theta(MSIT, \mathcal{E}, E, \mathcal{W}) \cdot \frac{1}{w_E} \qquad (11)$$

Since wiresizing solution $\mathcal{W}$ is the only variable factor in the local refinement operation and equation (8)–(10) depend only on the wire width assignment for edges other than E, for simplicity, we use $\Psi(\mathcal{W}, \overline{E})$, $\Phi(\mathcal{W}, \overline{E})$ and $\Theta(\mathcal{W}, \overline{E})$ instead of $\Psi(MSIT, \mathcal{E}, E, \mathcal{W})$, $\Phi(MSIT, \mathcal{E}, E, \mathcal{W})$ and $\Theta(MSIT, \mathcal{E}, E, \mathcal{W})$, respectively.

Because $\Psi(\mathcal{W}, \overline{E})$, $\Phi(\mathcal{W}, \overline{E})$, and $\Theta(\mathcal{W}, \overline{E})$ are independent of w_E, they are considered as constant for local refinement of E. Minimizing (11) gives the local refinement $\tilde{w}_E$ of segment E. Notice that for any pair of wiresizing solutions $\mathcal{W}$ and $\mathcal{W}'$ such that $\mathcal{W}'$ dominates $\mathcal{W}$. We have $\Phi(\mathcal{W}, \overline{E}) \geq \Phi(\mathcal{W}', \overline{E})$ and $\Theta(\mathcal{W}, \overline{E}) \leq \Theta(\mathcal{W}', \overline{E})$.

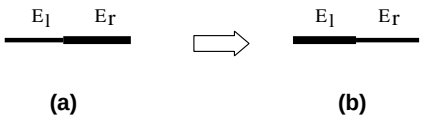


Figure 6: (a) The original width assignments for E_l and E_r. (b) Swap the width assignments for E_l and E_r.

Proof: If we denote $\overline{\mathcal{W}} = \mathcal{W} - \{w_{E_l}, w_{E_r}\}$, the objective function (3) can be written as:

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

We use $t(\mathcal{W}^*)$ to denote this objective function under the optimal wiresizing solution $\mathcal{W}^*$. After swapping the width assignments for E_l and E_r with respect to the optimal solution $\mathcal{W}^*$ (see Fig. 6), we denote the resulted wiresizing solution and correspondent objective function as $\mathcal{W}'$ and $t(\mathcal{W}')$, respectively.

According to Lemma 3 and 4

$$G(E_l) = G(E_r) = G(S) \tag{4}$$

$$H(E_l) = H(E_r) = H(S) \tag{5}$$

$$F(E_l, E) = F(E_r, E) \quad if\ E \neq E_r \neq E_l \tag{6}$$

$$F(E, E_l) = F(E, E_r) \quad if\ E \neq E_r \neq E_l \tag{7}$$

thus,

$$\begin{aligned} t(\mathcal{W}') - t(\mathcal{W}^*) &= \mathcal{K}_2 \cdot F(E_l, E_r) \cdot (\frac{w^*_{E_l}}{w^*_{E_r}} - \frac{w^*_{E_r}}{w^*_{E_l}}) + \mathcal{K}_2 \cdot F(E_r, E_l) \cdot (\frac{w^*_{E_r}}{w^*_{E_l}} - \frac{w^*_{E_l}}{w^*_{E_r}}) + \\ & \mathcal{K}_3 \cdot F(E_l, E_r) \cdot (\frac{1}{w^*_{E_r}} - \frac{1}{w^*_{E_l}}) + \mathcal{K}_3 \cdot F(E_r, E_l) \cdot (\frac{1}{w^*_{E_l}} - \frac{1}{w^*_{E_r}}) \\ &= \{F(E_l, E_r) - F(E_r, E_l)\} \cdot \{w^*_{E_l} - w^*_{E_r}\} \cdot \{\mathcal{K}_2 \cdot \frac{(w^*_{E_l} + w^*_{E_r}) + \mathcal{K}_3)}{w^*_{E_l} \cdot w^*_{E_r}}\} \end{aligned}$$

Because $\frac{(w^*_{E_l} + w^*_{E_r}) + \mathcal{K}_3)}{w^*_{E_l} \cdot w^*_{E_r}} > 0$ and $t(\mathcal{W}') - t(\mathcal{W}^*) \geq 0$, we have $\{F(E_l, E_r) - F(E_r, E_l)\} \cdot \{w^*_{E_l} - w^*_{E_r}\} \geq 0$.

□

Lemma 6 *If $F_l(S) > F_r(S)$, then there exists an optimal wiresizing solution where the width assignment for edges within the segment S is increased rightward. Similarly, it is decreased rightward if $F_l(S) < F_r(S)$.*

Proof: Assume E_l and E_r are two arbitrary adjacent edges under the finest edge-division in S and E_l is just left to E_r, $w^*_{E_l}$ and $w^*_{E_r}$ are their optimal width assignment, respectively. Clearly, $w^*_{E_l} \geq w^*_{E_r}$ if $F(E_l, E_r) >$

B. LST Monotone Property

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

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

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

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

In this case, the left-right relationship in an LST is the same as the ancestor-descendant relationship in an $SSIT$. If we replace the ancestor-descendant relationship in the proof of the $SSIT$ monotone property in [6] by the left-right relationship, the proof is applicable to the LST monotone property. □

C. Local Monotone Property

We shall prove the *SST local monotone property* based on a series of lemmas. For simplicity, we assume the finest edge-division for this proof. Once the *local monotone property* holds in the finest edge-division, it holds in any valid edge-division.

Lemma 4 *If E_l and E_r are two adjacent edges within a segment S and E_l is just left to E_r, then*

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

There are two cases for Lemma 4.

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

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

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

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

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

Lemma 5 *If E_l and E_r are two edges in a segment S and E_l is just left to E_r, then*

$\{F(E_l, E_r) - F(E_r, E_l)\} \cdot \{w^*_{E_l} - w^*_{E_r}\} \geq 0.$

$$\left[\sum_{E\in SST}\nu(E)\cdot\frac{1}{w_E}+\sum_{LST\in\mathcal{L}}\left\{\sum_{E\in LST}\nu(E)\cdot\frac{1}{w_E}\right\}\right]+$$
$$\left[\sum_{E\in SST}\phi(E)\cdot\frac{1}{w_E}+\sum_{LST\in\mathcal{L}}\left\{\sum_{E\in LST}\phi(E)\cdot\frac{1}{w_E}\right\}\right]$$

We define:

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

We then have:

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

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

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

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

We then have:

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

Because the wire widths in both SST and P are given, $\mathcal{H}_{P,1}$ is a constant which depends only on the wire widths in SST and P. For any $b\in\mathcal{B}$, $\mathcal{H}_{P,2}(b)$ is a function whose variables are the widths of the edges in SST, P and b. Therefore, the contribution of b to the objective function (3) is independent of any other subtree in $\mathcal{B}$. Clearly, the separability within an LST holds. □

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

Lemma 1–3 can be verified based on the definitions of the coefficient functions F, G and H. Note that given an $MSIT$ with m segments, f^{ij}, g^{ij} and h^{ij} can be computed in $O(m^2)$ time, thus $F(S, S')$ can be computed in $O(m^4)$ time and stored in an $m \times m$ matrix, and $F_l(S), F_r(S), G(S)$ and $H(S)$ can also be computed in $O(m^4)$ time but stored in linear arrays of size m. Such computation is carried out before the sizing procedure.

A. LST Separability

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

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

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

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

We then rewrite Eqn. (3) as a function of the loading subtrees in $\mathcal{L}$ and the source subtree SST:

$$\begin{aligned} t(MSIT, R_d, \mathcal{W}) &= \sum_{E \in MSIT} \mathcal{K}_1 \cdot w_E + \sum_{E, E' \in MSIT, E \neq E'} \mu(E, E') \cdot \frac{w_{E'}}{w_E} + \sum_{E, E' \in MSIT, E \neq E'} \omega(E, E') \cdot \frac{1}{w_E} + \\ &\quad \sum_{E \in MSIT} \nu(E) \cdot \frac{1}{w_E} + \sum_{E \in MSIT} \phi(E) \cdot \frac{1}{w_E} \\ &= \left[\sum_{E \in SST} \mathcal{K}_1 \cdot w_E + \sum_{LST \in \mathcal{L}} \left\{ \sum_{E \in LST} \mathcal{K}_1 \cdot w_E \right\} \right] + \\ &\quad \left[\sum_{E, E' \in SST, E \neq E'} \mu(E, E') \cdot \frac{w_{E'}}{w_E} + \sum_{LST \in \mathcal{L}} \left\{ \sum_{E \in SST, E' \in LST} \mu(E, E') \cdot \frac{w_{E'}}{w_E} \right\} + \right. \\ &\quad \left. \sum_{LST \in \mathcal{L}} \left\{ \sum_{E, E' \in LST, E \neq E'} \mu(E, E') \cdot \frac{w_{E'}}{w_E} \right\} \right] + \\ &\quad \left[\sum_{E, E' \in SST, E \neq E'} \omega(E, E') \cdot \frac{1}{w_E} + \sum_{LST \in \mathcal{L}} \left\{ \sum_{E \in SST, E' \in LST} \omega(E, E') \cdot \frac{1}{w_E} \right\} + \right. \\ &\quad \left. \sum_{LST \in \mathcal{L}} \left\{ \sum_{E, E' \in LST, E \neq E'} \omega(E, E') \cdot \frac{1}{w_E} \right\} \right] + \end{aligned}$$

lower and upper bounds, one reason is that the current implementation builds the data structure for the finest edge-division even if the *bundled refinement* does not need it at all. Thus, the total running time still can be reduced in further implementation without building the data structure for the finest edge-division.

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

Table 8: Performance comparison between $GWSA$ and $BWSA$

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

Table 9: Total running time comparison between $GWSA$-based and $BWSA$-based algorithms

7 Conclusions and Future Work

The results in this paper have shown convincingly that proper sizing of the wire segments in multi-source nets can lead to significant reduction in the interconnect delay. We have also developed an efficient wiresizing algorithm using *(coarse) variable* edge-division, which achieves the same optimal wiresizing solutions as the optimal wiresizing algorithms based on the finest edge-division, but uses much less computational time.

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

Acknowledgements

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

Appendix: Proof of Theorem

We shall first present the following properties for the coefficient functions F, G and H, which are of help to prove Theorem 1 to 6.

Lemma 1 Given an *MSIT*, for any edges E_1 and E_2 within a segment S, $F(E_1, E_2)$ is an invariant (denoted $F_l(S)$) if E_1 is left to E_2, and $F(E_1, E_2)$ is another invariant (denoted $F_r(S)$) if E_1 is right to E_2.

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

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

Table 6: Multi-source wiresizing results on random nets

	net length (μm)	Normalized Area opt_msws	Maximum Delay (ns) min_width	Maximum Delay (ns) opt_msws	Average Delay (ns) min_width	Average Delay (ns) opt_msws
net1	3600	1.000	0.1435	0.1435 (0.0%)	0.1260	0.1260(0.0%)
net2	6600	1.044	0.2572	0.2567 (-0.2%)	0.2004	0.1994(-0.50%)
net3	10070	1.475	0.5230	0.4025 (-23.0%)	0.3504	0.3241(-7.5%)
net4	10570	2.000	0.5853	0.4873 (-16.7%)	0.5007	0.3846(-23.2%)
net5	16980	1.775	0.9496	0.7635 (-19.6%)	0.6375	0.5711(-10.4%)
net6	31980	2.706	3.4505	2.1979 (-36.3%)	1.8968	1.4512(-23.5%)

Table 7: Multi-source wiresizing results on several nets in an Intel microprocessor layout

by $GWSA$ under the finest edge-division and by $BWSA$ under the refined edge-division, respectively. Since $BWSA$ can be used for single-source wiresizing as well, we test both algorithms again by assuming there is only one source together with eight timing critical sinks in each net. Similar speed-up factors were obtained under different experiment conditions.[5] Results for multi-source wiresizing in the CMOS technology are reported in Table 8 with the speed-up ranging from a factor of two orders of magnitude to a factor of three orders of magnitude. In all these experiments, both $BWSA$ and $GWSA$ provide the convergent lower/upper bounds, i.e., the optimal solutions are achieved just by lower/upper bound computations.

B. Results on Industrial Nets

Furthermore, we applied both $BWSA$ and $GWSA$ algorithms to the set of Intel nets. Because the time to compute the lower/upper bounds for most nets in this set is too small to measure, we compared the total running time. In Table 9, the *BWSA-based* algorithm is just $OWBR$, i.e., $BWSA$ to compute lower and upper bounds, followed by enumerating for the SST and $OWSA$ for LSTs. The *GWSA-based* algorithm is just to replace $BWSA$ by $GWSA$ in the $OWBR$ scheme. $OWBR$ is observed to run more than 100x faster than the $GWSA$-based algorithm.

It is interesting to note that $BWSA$ gives identical lower and upper bounds for net3 while $GWSA$ *does not*, which is also an example of multiple $\mathcal{E}_F$-tight bounds for the optimal solution as mentioned in Section 5.1.C. Also note that, in case of $OWBR$, the total running time is *not* dominated by the time to compute

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

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

Net Length	3000 μm			30000 μm		
MCM Wiresizing	min_width	opt_ssws	opt_msws	min_width	opt_ssws	opt_msws
Maximum Delay (ns)	0.417	0.425(+1.9%)	0.406(-2.63%)	4.76	4.94(+4.3%)	3.23(-32.1%)
Average Delay (ns)	0.361	0.359(-0.6%)	0.321(-11.1%)	3.70	3.46(-6.5%)	2.71(-26.7%)
Normalized Area	1	2.39	1.83	1	2.49	3.44

Table 5: Comparison of *min_width*, *opt_ssws* and *opt_msws* solutions on a four-source tree with H-topology

may have larger maximum delay and consume larger routing area, while our *opt_msws* solutions reduce the maximum delay and the average delay by up to 32.7% and 25.5% for the CMOS technology, respectively, and up to 32.1% and 26.7% for the MCM technology, respectively. The single-source wiresizing method is clearly not applicable to the *MSWS* problem.

B. Results on Random Nets

Ten eight-pin nets with pins randomly distributing in a 1000×1000 grid and routed again by the 1-Steiner algorithm are used in our experiments. For each eight-pin random net, eight timing-critical source-sink pairs are chosen randomly. Wiresizing is carried out under both the CMOS technology and the MCM technology by treating each grid edge as a wire of unit length. Table 6 shows the comparison of the *min_width* and *opt_msws* wiresizing solutions for four of such nets. Clearly, *opt_msws* solutions consistently outperform the min_width solutions. On average, we observe 12.8% and 7.9% reduction of the maximum delay and the average delay among eight critical source-sink pairs in each net for the submicron IC technology, respectively, and 15.6% and 10.6% for the MCM technology, respectively.

C. Results on Industrial Nets

We also tested our algorithms on these real multi-source nets provided by Intel. As shown in Table 7, the *opt_msws* solutions consistently outperform the *min_width* solutions with as much as 36.3% and 23.5% reduction on the maximum delay and the average delay, respectively.

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

6.3 Speed-up Using Variable Edge-Division

A. Results on Random Nets

The ten eight-pin nets used in the above subsection were test for both CMOS and MCM technologies. We compared the CPU times to obtain the lower and upper bounds of the optimal wiresizing solution, used

which are between 0.55% and 0.03%. As an estimation, the suboptimality of the Elmore-delay optimal solution against the SPICE-delay optimal solution shall be of this range, i.e., 0.55% to 0.03%.

Furthermore, these two measurements are computed for the best 100 and 10 wiresizing solutions in terms of the Elmore delay model of each random solution set, respectively. It turns out that the better the wiresizing solutions according to the Elmore delay model, the less the deviations of the SPICE-delay for the correspondent ranking difference. For example, among the 1,000 random solutions for net1, this type of deviation is 0.0195% for the best 100 and only 0.0022% for the best 10. If we assume that these random solutions are uniformly distributed in the overall solution space, it is implied that, in general, in the area near the optimum in the solution space, the Elmore delay model has a even higher fidelity.

Based on data for the best 10 of these random solution sets, the suboptimality of the Elmore delay model for wiresizing optimization is less than 0.015%.[3] Thus, we believe that the Elmore delay model has really high fidelity for wiresizing optimization. Furthermore, this conclusion is applicable to all other wiresizing works [10, 6, 16] where the Elmore delay model is used as well.

In our SPICE simulation, the inductance is not taken into consideration, which is justified by the fact that the inductive effect is negligible under current CMOS technology. The fidelity will be studied in the feature for the MCM technology where the inductive effect tends to be no longer negligible.[4]

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

6.2 Comparison between Different Wiresizing Solutions

Let *min_width* be the wiresizing solution with minimum wire width W_1 everywhere, *opt_ssws* be the wiresizing solution given by the optimal *SSWS* algorithms[10, 6], and *opt_msws* be the wiresizing solution given by our optimal *MSWS* algorithms. Also, let *normalized area* denote the area ratio of wiresizing solution versus the *min_width* wiresizing solution, which is equivalent to the average wire with if the minimum wire width W_1 is scaled to 1. Both *maximum delay* and *average delay* are only in terms of critical source-sink pairs. In the following experiments, we assign $\lambda^{ij} = 1$ for a critical source-sink pair and $\lambda^{ij} = 0$ otherwise, thus, the objective in equation (3) is equivalent to the *average delay* among critical source-sink pairs.

A. Results on Simple Artificial Nets

Wiresizing results on the four-pin H-topology interconnect tree in Figure 1 are shown in Table 5, for both the CMOS technology and the MCM technology. Compared to the *min_width* solutions, the *opt_ssws* solutions

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

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

6.1 Fidelity of the Elmore Delay Model

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

We measured the *fidelity* on several real multi-source nets provided by Intel. These nets were extracted from the top-level floorplan of a high-performance microprocessor. Most pins of these nets can serve as both inputs and outputs, and all pairs between sources and sinks (excluding feedthrough pins) are considered to be timing critical. We use 1-Steiner tree algorithm [15] to route these nets. Table 3 summarizes the routing trees for these nets.

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

Table 3: Routing trees for multi-source nets extracted from the layout for a high performance Intel microprocessor

We computed both the weighted average Elmore delay and the weighted average SPICE-computed 50% delay. The ranking technique similar to [2] was used: first rank wiresizing solutions by their Elmore delay, then rank them by their SPICE-computed delay, and find the average of the absolute value of the difference between the two rankings. Because the number of wiresizing solutions is prohibitively large to enumerate, 1,000 wiresizing solutions are generated with random wire width for each $10\mu m$-long wire and both SPICE and Elmore delays are computed and then ranked.

Table 4 depicts the fidelity of the Elmore delay model against the SPICE-computed 50% delay for wiresizing optimization. The average differences in ranking are between 2.068 and 56.95 of 1,000. Besides, we compute the average deviations in term of the SPICE-computed delay according to the average ranking differences,

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

Table 4: Average difference in ranking and the corresponding suboptimality for Intel nets based on $0.5\mu m$ CMOS technology

immediately. Otherwise, the optimal solution shall be found between the lower and upper bounds where them do not meet.

Because the *separability* and the *monotone property* hold in *LST*s, the optimal single-source wiresizing algorithm, named *OWSA* [10], can be used for *LST*s. The *OWSA* algorithm is a polynomial-time dynamic programming method. Without the *separability* in *SST*, the wire width assignment for non-convergent edges in the *SST* need to be enumerated subject to the *local monotone* property.

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

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

6 Experimental Results

We have implemented the *OWBR* algorithm in ANSI C for the Sun SPARC station environment and tested our algorithm on a large number of *MSIT*s for both the MCM and the advanced IC technologies, including several multi-source nets extracted from the layout of an Intel high performance microprocessor.

The parameters used in our experiments are summarized in Table 2. The IC technology parameters are based on the $0.5\mu m$ CMOS process provided by North Carolina Microelectronic Center (MCNC)[17], and the MCM technology parameters were obtained from [11]. The loading capacitance in IC technology is the total gate capacitance of a minimize size transistor in the MCNC $0.5\mu m$ CMOS, while the minimum loading capacitance in MCM technology is the pad capacitance of $1000fF$ on MCM substrate.

The wire width choices are $\{ W_1, 2W_1, 3W_1, 4W_1, 5W_1\}$, where W_1 is the minimum wire width ($0.95\mu m$ in the IC technology and $10\mu m$ in the MCM technology). Note that our algorithms are still valid if the wire widths are not integral multiples of the minimum width. The wire length units are 10 μm for the IC technology and 100 μm for the MCM technology, respectively.

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

Table 2: Technology parameters based on advanced IC and MCM designs.

B. Complexity

It is proven in [10] that the worst-case complexity of GWSA for the *SSWS* problem is $O(r \cdot n^3)$, where r is the number of wire-width choices, n is the total number of edges of a fixed edge-division. The proof is still true for our formulation of the *MSWS* problem, except the complexity of computing these coefficients function is slightly higher in the *MSWS* case.

When the *finest* edge-division $\mathcal{E}_F$ is used, n is also the total wire length of an *MSIT* in terms of the wire length unit δ. Let n_0 be the wire length of the longest segment in the *MSIT* and m the number of segments in the *MSIT*, we shall show that the average-case complexity of *BWSA* is only $O(r \cdot m^3 \cdot \ln^4 n_0)$, much below the worst-case complexity $O(r \cdot n^3)$ for *GWSA*.

In *BWSA*, each non-convergent edge is divided into two edges during edge refinement, then a *pass* of bundled refinement operations will be carried out. We assume that one of the two edges for an original non-convergent edge shall be convergent after such a pass of *bundled refinement*. This assumption holds very likely after the first several *passes*.

Under this assumption, each segment can be refined into at most $ln n_0 + 1$ edges. In essence, each pass of *BWSA* is a single execution of *GWSA* under the current edge division. The total number of edges is $O(m \cdot \ln n_0)$, thus the complexity of a pass in *BWSA* is $O(r \cdot (m \cdot \ln n_0)^3) = O(r \cdot m^3 \cdot \ln^3 n_0)$. Note that the total number of passes in *BWSA* can not be more than $\ln n_0$, the average-case complexity of *BWSA* is $O(r \cdot m^3 \cdot \ln^3 n_0) \cdot O(\ln n_0) = O(r \cdot m^3 \cdot \ln^4 n_0)$.

Let k be the number of pins in an *MSIT*, then $m \leq 2(k-1)$. Since k is a constant bounded by the size of the largest multi-source net (normally no larger than 32), m is also a constant, plus $\ln n_0 \ll n$, *BWSA* is much more efficient than *GWSA*. This has been further confirmed by experimental results.

C. Optimality

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

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

5.2 Optimal Wiresizing Algorithm Using Bundled Refinement

An optimal wiresizing algorithm using bundled refinement (*OWBR* algorithm) is presented. Given an *MSIT*, OWBR first computes a lower bound and an upper bound of the optimal wiresizing solution using BWSA. If the lower and upper bounds meet, which is very likely in practice, we get the optimal wiresizing solution

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

Function gBWSA_L/U($\delta, \mathcal{E}, \mathcal{W}_{lower}/\mathcal{W}_{upper}$)
/*
* Given the wire length unit δ, an edge-division $\mathcal{E}$, a lower/upper bound $\mathcal{W}_{lower}/\mathcal{W}_{upper}$ of the optimal wiresizing solution, and a set of possible wire widths $\{W_1, W_2, \cdots, W_r\}$, return a tighter lower/upper bound.
*/
$\mathcal{W} \leftarrow \mathcal{W}_{lower}/\mathcal{W}_{upper}$;
do
$progress \leftarrow$ `false`;
for each edge E of $\mathcal{E}$ **do**
$w \leftarrow$ BRL($\delta, \mathcal{E}, \mathcal{W}_{lower}, E$) or BRU($\delta, \mathcal{E}, \mathcal{W}_{upper}, E$);
if $w \neq \mathcal{W}(E)$ **then**
$progress \leftarrow$ `true`; $\mathcal{W}(E) \leftarrow w$;
end if
end for;
while $progress =$ `false`;
return $\mathcal{W}$;
end Function;

Function NER($\delta, \mathcal{E}, \mathcal{W}_{lower}, \mathcal{W}_{upper}$)
/*
* Given the wire length unit δ, an edge-division $\mathcal{E}$, a lower bound and an upper bound of the optimal wiresizing solution, for each non-convergent edge, if it is longer than δ, divide it into two edges, each inherits the bounds(Need-driven Edge Refinement). Once the edge-division has been refined, set the flag not_done true.
*/
$not_done \leftarrow false$;
for each edge E of $\mathcal{E}$ **do**
if $\mathcal{W}_{lower}(E) \neq \mathcal{W}_{upper}(E)$, and E is longer than δ
divide E into two edges with (nearly) equal lengths;
$not_done \leftarrow true$;
return $\mathcal{E}$ and not_done;
end Function;

Function BWSA($MSIT, \delta$)
/*
* Given an $MSIT$, the wire length unit δ and a set of possible wire widths $\{W_1, W_2, \cdots, W_r\}$, return the lower and upper bounds of the optimal wiresizing solution of the $MSIT$ by applying gBWSA_L() and gBWSA_U() with a recursively refined edge-division.
*/
$\mathcal{E} \leftarrow MSIT$; $\mathcal{W}_{lower} \leftarrow W_1$; $\mathcal{W}_{upper} \leftarrow W_r$;
do
$\mathcal{W}_{lower} \leftarrow$ gBWSA_L($\delta, \mathcal{E}, \mathcal{W}_{lower}$);
$\mathcal{W}_{upper} \leftarrow$ gBWSA_U($\delta, \mathcal{E}, \mathcal{W}_{upper}$);
$\mathcal{E} \leftarrow NER(\delta, \mathcal{E}, \mathcal{W}_{lower}, \mathcal{W}_{upper})$;
while not_done
return $\mathcal{W}_{lower}$ and $\mathcal{W}_{upper}$;
end Function;

Table 1: Bundled Wiresizing Algorithm (BWSA)

This kind of pass iteration is repeated until we either have the identical lower and upper bound for all edges under current division, or each non-convergent edge are δ long.

E_r based on the objective function (3) while keeping the assignment of $\mathcal{W}$ on $\overline{E_r}$ and any edge E' other than E. Then, $\tilde{w}_{E_r}$ is regarded as a refined lower bound of the *entire* edge E.

We can show the following:

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

5 Optimal Wiresizing Algorithm

5.1 Bundled Wiresizing Algorithm

The greedy wiresizing algorithm GWSA was given in [10]. Working on a *fixed* edge-division, GWSA can use *local refinement* operations to compute the lower/upper bounds of the optimal wiresizing solution starting with the minimum and maximum wire width assignments, respectively. Our *bundled wiresizing algorithm (BWSA)* (Table 1) computes the lower/upper bounds of the optimal wiresizing solution for an *MSIT*, also starting with the minimum and maximum wire width assignments, respectively, but using *bundled refinement* operations rather than *local refinement* operations, and a gradually refined edge-division rather than a fixed one. BWSA achieves the same optimal lower and upper bounds for much less computation costs when compared with GWSA.

A. Overview

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

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

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

Definition 2 (Bundled Edge) *Given an MSIT, a segment S and the finest edge-division $\mathcal{E}_F$, let $E_1, \cdots, E_p$ be a maximal sequence of successive edges in S under $\mathcal{E}_F$, with the same wire width in the optimal wiresizing solution under $\mathcal{E}_F$, we say that they form a bundled edge (see Figure 5).*

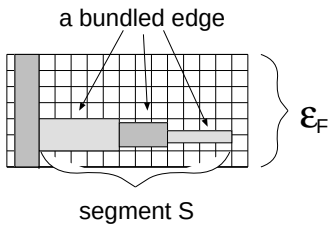


Figure 5: The optimal wiresizing solution for segment S under the finest edge-division $\mathcal{E}_F$. There are three bundled edges in segment S.

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

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

4.2 Bundled Refinement Operations

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

The rational for the *BRU* operation is as follows: if $F_l(S) \geq F_r(S)$, in the optimal solution $\mathcal{W}^*$, E_l is always wider than all edges under $\mathcal{E}_F$ in $\overline{E_l}$ (the *local monotone* property). The refinement of an upper bound of $w^*_{E_l}$ is still an upper bound of it, thus also gives an (possibly refined) upper bound of the optimal wire width assignments for any edge under $\mathcal{E}_F$ in $\overline{E_l}$. Note that E will not be divided into E_l and $\overline{E_l}$ when performing the *BRU* operation on edges other than E.

Similarly, the *bundled refinement operation for the lower bound (BRL)* can be defined for a wiresizing solution $\mathcal{W}$ dominated by $\mathcal{W}^*$. Again, assuming $F_l(S) \geq F_r(S)$, we treat E as two edges E_r and $\overline{E_r}$. E_r is the right end of E and with length δ, $\overline{E_r}$ is the remaining part of E. Let $\tilde{w}_{E_r}$ be the locally optimized width for

Although the *dominance property* was proven based on the ancestor-descendant relation in [10] for the *SSWS* problem, we showed that it is a general property neither dependent on the ancestor-descendent relation, nor on the left-right relation.

Theorem 4 enables efficient computations of lower and upper bounds of the optimal wiresizing solution by the *GWSA* algorithm in [10]. It uses the *local refinement* operation iteratively to tighten the lower bound or the upper bound of each edge in an *MSIT*. A much more powerful refinement operation, called the *bundled refinement* operation, which may tighten the lower bound or the upper bound for a number of edges in a single operation, will be introduced in the next section.

4 Properties of Optimal MSWS/E Solutions

Up to now, both the *MSWS* problem defined in this paper and the *SSWS* problem studied in [10, 6, 16, 18] are only studied in the context of a *fixed* edge-division. The edge-division controls how often the wire width is allowed to change. However, it is difficult to choose a proper edge-division. For best accuracy, a very fine, uniform edge-division needs to be chosen, which results in much high memory usage and computation time due to the large number of edges. We now investigate methods to obtain the optimal wiresizing results using a *non-uniform* and *coarser* edge-division .

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

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

4.1 Edge-division Refinement and Bundled Edges

Given an *MSIT*, let $\mathcal{E}_0$ be the edge-division with each segment in the *MSIT* as an edge, $\mathcal{E}_F$ the uniform edge-division where each edge is δ long, δ is the unit wire length and is determined by the design rules. If each edge in edge-division $\mathcal{E}$ corresponds to one or several edges in edge-division $\mathcal{E}'$, $\mathcal{E}'$ is a refinement of $\mathcal{E}$. An edge-division $\mathcal{E}$ is *valid* only if $\mathcal{E}$ is a refinement of $\mathcal{E}_0$ and the length of every edge is a multiple of δ. Then, among all valid edge-divisions, $\mathcal{E}_0$ is coarsest and $\mathcal{E}_F$ is finest.

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

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

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

This *LST monotone* property can be proven by considering an *LST* as an *SSIT* with its driver located at the branching node from the *SST*, and replacing the left-right relation in the *LST* with the ancestor-descendent relation in an *SSIT*.

Note that the *monotone* property just holds within each *LST*. The root edge in an *LST* may be wider than the edge from which the *LST* branches off (see Figure 4). This example also shows that the monotone property like that in the *SSWS* problem does not hold for any particular source on an *MSIT*.

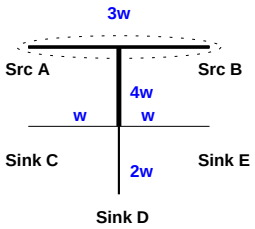


Figure 4: The optimal wire width assignments for a two-source net with w being the minimum wire width. The *SST* is surrounded by the dashed curve. Segments outside the curve belong to an *LST*. The wire width assignment is *monotone* in the *LST*. However, the root edge of the *LST* is wider than edges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

C. SST Local Monotone Property

Although the signal direction is changeable in the edges of the *SST* when different sources are active, surprisingly, our study shows that optimal *MSWS* solutions still satisfy a local monotone property.

Lemma 1 *Given an* MSIT, *for any edges E_1 and E_2 within a segment S, $F(E_1, E_2)$ is an invariant (denoted $F_l(S)$) if E_1 is left to E_2, and $F(E_1, E_2)$ is another invariant (denoted $F_r(S)$) if E_1 is right to E_2.*

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

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

D. Dominance Property

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

3.2 Decomposition of an MSIT

In order to reduce the complexity with the *MSWS* problem, we decompose an *MSIT* into a *source subtree (SST)* and a set of *loading subtrees (LSTs)* (see Figure 3). The *SST* [1] is the subtree spanned by all source nodes in the *MSIT*. After we remove the *SST* from the *MSIT*, the remaining segments form a set of subtrees, each of them is called an *LST*. When every pin of an *MSIT* can be a source at different times, the *SST* is the entire *MSIT* and there is no *LST*.

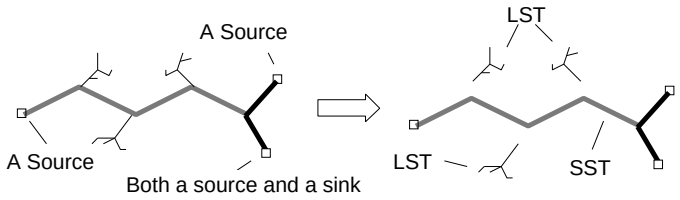


Figure 3: An *MSIT* can be decomposed into a source subtree *SST*, and a set of subtrees (three *LST*s here) branching off from the *SST*. The dark edges belong to the *SST*.

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

3.3 Properties of Optimal MSWS Solutions

A. LST Separability

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

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

B. LST Monotone Property

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

[1] Note that *SST* defined in this paper is different from that defined in [10], where *SST* is used to denote a single stem tree.

$$\mathcal{K}_3 \cdot \sum_{E,E' \in MSIT} F(E,E') \cdot \frac{l_E \cdot l_{E'}}{w_E} \; + \; \mathcal{K}_4 \cdot \sum_{E \in MSIT} G(E) \cdot \frac{l_E}{w_E} \; + \; \mathcal{K}_5 \cdot \sum_{E \in MSIT} H(E) \cdot \frac{l_E^2}{w_E} \qquad (3)$$

where

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

Although this weighted delay formulation for multiple sources and sinks is very similar to that for the single source and multiple sinks in [6], the coefficient functions F, G and H have very different properties, which lead to much higher complexity and very different properties for the *MSWS* problem when compared to the *SSWS* problem. These properties will be discussed in Section 3.

3 Properties of Optimal MSWS Solutions

3.1 Review of Optimal SSWS Solutions

When there is only one source in the routing tree, each edge has a unique signal flow direction. We can define the "ancestors" and "descendants" with respect to the signal flow in the tree. The following properties of optimal *SSWS* solutions were given in [10].

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

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

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

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

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

The presence of multiple sources greatly complicates the wiresizing problem. For example, with multiple sources, even a monotone wiresizing is not well defined. Nevertheless, our research have revealed a number of interesting properties of the optimal wiresizing solutions for *MSIT*s, which will be presented in the next a few subsections. Some of these properties generalize the results on the *SSWS* problem, and others are unique for the *MSWS* problem.

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

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

where $t^{ij}(MSIT, \mathcal{E}, \mathcal{W})$ is the Elmore delay [12] between the source N_i and the sink N_j. It is a function of the edge-division $\mathcal{E}$ and the wiresizing solution $\mathcal{W}$, with the penalty weight λ^{ij} to indicate its priority.

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

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

When there is only one source in an interconnect tree, the $MSWS$ problem degenerates into the single source wiresizing ($SSWS$) problem. Also, a slightly more general wiresizing problem, the multi-source wiresizing problem without a given edge-division (*the $MSWS/E$ problem)* will be formulated in Section 4.

2.2 Weighted Delay Formulation

Similar to [10, 6], the Elmore delay t^{ij} between source N^i and sink N^j can be formulated as follows:

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

where w_E and l_E are respectively the (wire) width and length of the edge E, respectively. When $\mathcal{E}$ is uniform and every edge has the same length, the length can be omitted from this equation, just as given in [6]. Besides, $\mathcal{K}_0^{ij}, \mathcal{K}_2, \cdots, \mathcal{K}_5$ are constants depending only on the IC or MCM fabrication technology (details concerning them can be found in [10, 6]), and $f^{ij}(E, E'), g^{ij}(E)$ and $H^{ij}(E)$ are defined below.

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

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

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

where $P(N_i+, N_j)$ is the unique path from the driver resistor N_{i+} (of source N_i) to sink N_j, $Des^i(E)$ is the edge set of all downstream edges of E with respect to the signal flow from source N_i, $sink(E)$ is the node set of all downstream sinks of E, again with respect to the signal flow from source N_i and c_v^s is the capacitance at sink v.

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

$$t(MSIT, \mathcal{E}, \mathcal{W}) = \mathcal{K}_0 + \mathcal{K}_1 \cdot \sum_{E \in MSIT} l_E \cdot w_E + \mathcal{K}_2 \cdot \sum_{E, E' \in MSIT} F(E, E') \cdot \frac{l_E \cdot l_{E'} \cdot w_{E'}}{w_E} +$$

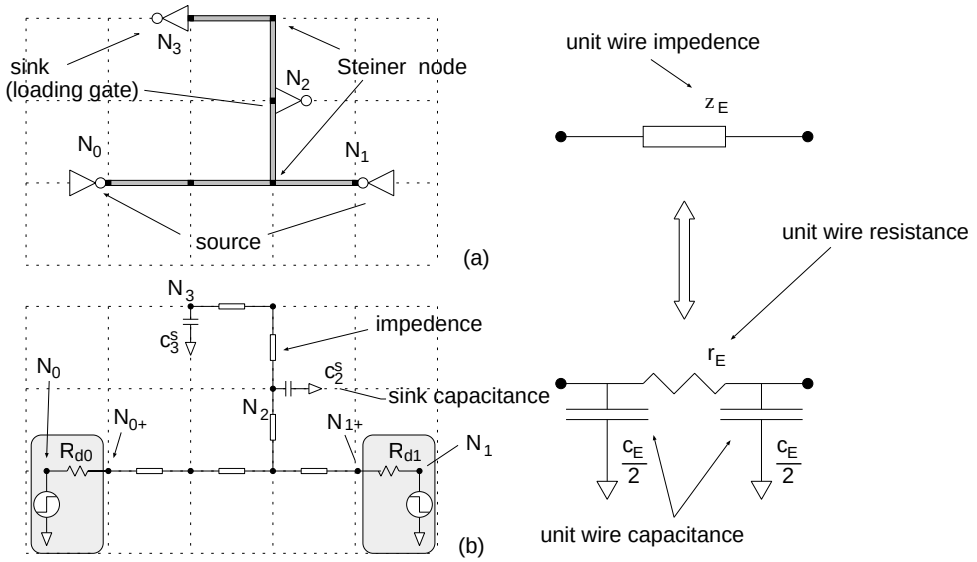


Figure 2: A two-source and two-sink interconnect tree with its distributed RC model under a uniform edge-division. (a) The layout of an $MSIT$ with two sources N_0 and N_1, and two sinks N_2 and N_3. (b) The distributed RC interconnect model of this $MSIT$. Each edge e under the uniform edge-division is modeled as a π-type RC circuit containing a resistor of r_E and two capacitors of $\frac{c_E}{2}$ each. Each sink v has an extra loading capacitance C_v^s. Each source is modeled by a driver resistance connected to an ideal voltage source.

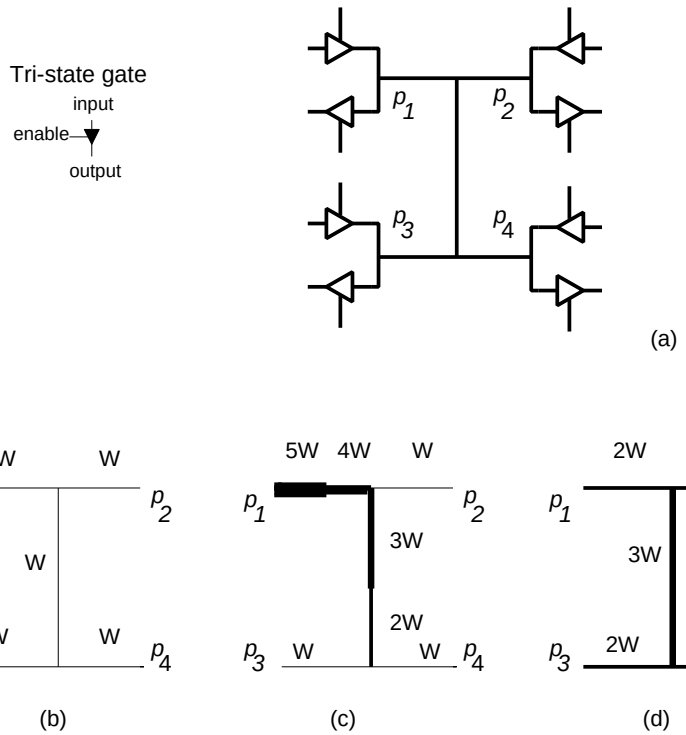
2 Problem Formulation

2.1 Wiresizing for $MSIT$

Given an *MSIT*, each pin can be a source (driver), or a sink (receiver), or both at different times. Let *src(MSIT)* be the set of sources, and *sink(MSIT)* the set of sinks. A pin belongs to both *src(MSIT)* and *sink(MSIT)* if it can be a source and a sink at different times. We assume that no two sources are active at the same time.

A *node* refers to either a pin or a Steiner node in the *MSIT*. A *segment* connects two *node*s. Let $\{S_1, S_2, \cdots, S_m\}$ be the set of segments in the *MSIT*. In practice, each segment is divided into a sequence of *edges*. Let $\{E_1, E_2, \cdots, E_n\}$ be the set of all *edges* in the *MSIT*. The wiresizing problem is to find a wire width from a set of given choices $\{W_1, W_2, \cdots, W_r\}$ ($W_1 < W_2 < \cdots < W_r$) for each edge. We assume that the wire width within an edge does not change, thus the *edge-division* controls how often we allow wire widths to change.

We follow the same modeling technique used in [10, 6]. Each edge is treated using a π-type RC circuit to model the wire resistance, wire area capacitance, and wire fringing capacitance. The sink is modeled by a loading capacitance, a source (driver) is modeled by a fixed-value driver resistor connected to a voltage source. So a given interconnect including its sources and sinks are modeled by a distributed RC circuit (see Figure 2 for the modeling of a two-source two-sink net). The reader may refer to [10, 6] for more details.



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

Figure 1: A multi-source interconnect tree with three wiresizing solutions (w is the minimum wire width)

(a) A signal net with four pins (P_1, P_2, P_3 and P_4). Each of them is controlled by a tri-state gate and may become a source. (b) The minimum width wiresizing solution (min_width). (c) The optimal single-source wiresizing solution (opt_ssws), assuming that pin P_1 is the unique source, and pins $\{P_2, P_3, P_4\}$ are the critical sinks (d) The optimal multi-source wiresizing solution (opt_msws).

wiresizing methods. To the best of our knowledge, it is the first work which presents an in-depth study of both the optimal wiresizing problem for multi-source interconnect tree and the optimal wiresizing problem using a *variable* edge-division.

The remainder of this paper is organized as follows: In Section 2, we present the formulation of the *MSIT* wiresizing problem. In Section 3 and 4, we study the properties of the optimal wiresizing solutions for *MSIT* designs, respectively under fixed and variable edge-divisions. These properties lead to efficient algorithms given in Section 5. Section 6 shows experimental results. Section 7 concludes the paper with discussions of future work. All proofs will be given in the Appendix. An extended abstract of this paper was presented in ICCAD'95 [4].

a polynomial-time optimal algorithm was developed. Later on, a sensitivity-based wiresizing heuristic was proposed in [16] to minimize the maximum interconnect delay.

Further reduction of both delay and power dissipation was presented in [6] by simultaneous driver and wire sizing (*SDWS*). It revealed a relationship between the optimal driver sizing and the optimal wire sizing under the dominance relation, and developed an efficient SDWS algorithm. Recently, a sensitivity-based, simultaneous gate and interconnect sizing was presented in [18], which aims at reducing the critical path delay under a more accurate circuit-level delay model. Moreover, wiresizing heuristics to minimize clock skew was used in [19, 20, 3].

All these interconnect optimization methods assume that there is a unique source in each interconnect tree and minimize the delay between the source and a set of critical sinks. Thus, they are only applicable to *single source interconnect trees (SSITs)*. However, there exist many interconnect trees with multiple potential sources, each driving the interconnect tree at a different time. For example, in an interconnect tree designed for a bus, each pin is controlled by a tri-state gate and may become the source of the interconnect tree (see Figure 1(a), reproduced from [7]) . None of the existing interconnect optimization methods consider such *multi-source interconnect trees (MSITs)*, except a very recent work by Cong and Madden [7], where an $MSIT$ topology optimization method based on the construction of min-cost min-diameter A-trees was developed. Their method can reduce the maximum delay along an $MSIT$ by an average of 6.1% for submicron CMOS and 19.9% for MCM, respectively, when compared with conventional routing methods.

It is not difficult to see that the single-source wiresizing ($SSWS$) solution may perform poorly for wiresizing in $MSIT$ designs. For the multi-source net in Figure 1, compared with the minimum-width wiresizing solution min_width (Figure 1(b)), the optimal single-source wiresizing solution *opt_ssws* (Figure 1(c)) in fact increases the maximum delay by 0.6%, while the multi-source optimal wiresizing solution *opt_msws* (Figure 1(d)) reduces the maximum delay by 16.8%. Thus, multi-source wiresizing optimization method is necessary for $MSIT$ designs.

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

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

Optimal Wiresizing for Interconnects with Multiple Sources *

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

Abstract

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

1 Introduction

As the VLSI technology reaches submicron device dimensions, gigahertz clock frequencies, and highly integrated multi-chip module design, interconnect delay has become the dominating factor to determine system performance. Therefore, interconnect optimization for delay minimization has drawn much attention recently. Techniques for interconnect delay minimization falls into two categories. One is topology optimization, such as the constructions of bounded-radius bounded-cost trees[5], AHHK trees[1], A-trees[8], low-delay trees[2], and IDW/CFD trees[14]. In essence, these methods construct an interconnect tree to minimize both the total tree length and the paths between the input pin (also called the source) and a set of timing-critical output pins (also called critical sinks) , while the conventional Steiner tree algorithms only minimize the total tree length.

The other type of interconnect optimization methods is wiresizing of interconnect trees, which was first introduced in [8, 10] to minimize a weighted average interconnect delay from the source to critical sinks and

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