Wire Sizing as a Convex Optimization Problem: Exploring the Area-Delay Tradeoff

Sachin S. Sapatnekar
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011.

1 Introduction

It is rapidly becoming obvious that with the current trends in technology, interconnect delays are becoming an increasingly dominant factor in determining circuit speed. Until recently, interconnect resistance was often insignificant, while its capacitance was not, and hence optimal interconnect design frequently involved ensuring that all wire sizes were minimal. However, with advancement in technology, reduction in circuit geometries, increases in circuit speeds, and the advent of MCM's, the *wire sizing* problem for interconnect optimization has become significant.

The problem of wire sizing has not received very much attention until recently. Cong *et al.* presented some work in the area in [1, 2]. The approach in [1] used a delay model based on an upper bound [3] on the Elmore delay, and minimized the delay of the interconnect under minimum and maximum wire width constraints. This was extended in [2], where the Elmore delay was directly used to perform the timing optimization. The form of the Elmore delay model in this work makes the assumption that the critical leaf nodes of the interconnect tree are provided by the user (this information may, however, not be easily available in all design situations). A weighted sum of the Elmore delays to these leaf nodes is then minimized, where the weights are apparently user-defined.

In this work, we first use a form of the Elmore delay that does not require the critical leaf nodes to be specified. Like [1, 2], this work assumes that the interconnect network to be optimized is a tree structure. The objective here is to minimize the maximum of all Elmore delays at leaf nodes of the interconnect tree. Under this model, the separability property of the models in [1, 2] does not hold, and hence those algorithms will not provide the solution to this problem. Under this different delay model, we first prove some properties of the wire sizing problem, and then show a counterexample to show the invalidity of separability.

The contributions of this work are as follows. Firstly, this work presents, for the first time, a methodology for wire sizing under delay constraints that subsumes the case of sizing for minimum delay that has been studied before. Secondly, a smooth area-delay trade-off is shown, and it

is experimentally proved that achieving the minimum delay is not a good engineering solution; rather, a delay goal of 10-15 % above the minimum is a better engineering goal. Thirdly, this work uses the Elmore delay model where a distinct delay expression is used for the delay to each leafnode. The algorithms that are used are shown to give good results by comparing the solutions with lower bounds on the exact solutions.

The problem is formulated and its properties studied in Section 2. Two meaningful statements of the wire sizing problem are presented in Section 3. One statement minimizes the overall delay of the tree, while the other minimizes the wiring area under delay constraints at leaf nodes of the tree. Two efficient continuous optimization algorithms are presented in Section 4, of which one is a heuristic, and the other solves the underlying continuous optimization problem *exactly*. In Section 5, a mapping heuristic that transforms the continuous solution to the desired discrete solution is described. Finally, we present experimental results in Section 6, and conclude the paper in Section 7.

2 Formulation of the Problem

2.1 Modeling Interconnect and Interconnect Delay

This work models a wire as a succession of RC segments, shown in Figure 1, connected in series. The resistance, R_i, and capacitance, C_i, of the i^{th} segment are given by the formulæ

$$\begin{aligned} R_i &= \rho l_i / w_i \\ C_i &= \beta l_i \cdot w_i, \end{aligned} \tag{1}$$

where w_i and l_i are, respectively, the width and length of the i^{th} segment. Under the above model, any interconnect tree can be modeled using an equivalent RC tree.

In this work, we will use the words *width* and *size* interchangeably.

Definition 1: A node j is a *descendant* of node i in a tree T if the path from the root node of T to j contains node i. The node i is called an *ancestor* of node j. Similarly, wire S_p is an ancestor (descendant) of wire S_q if the path from the root node to S_q (S_p) contains wire S_p (S_q).

The delay $T_{d,i}$ of an RC tree is given by the well-known Elmore delay formula [3]. If P_i is the unique path from the root of the RC tree to node i, and $desc(j)$ represents all nodes that are descendants of node j in the tree, then according to this formula, the delay to node i is given by

$$T_{d,i} = \sum_{j \in P_i} R_j \sum_{k \in desc(j)} C_k \tag{2}$$

In an actual circuit, the root node is connected to a driver with equivalent resistance R_d, as shown in Figure 2. Moreover, in addition to wire capacitances, there may be several loading capacitances along the length of the wire. The Elmore delay to any node of the corresponding RC tree may easily be calculated using Eq. (2).

We take the Elmore delay of a tree as the maximum of the Elmore delays to any leaf node. An advantage of this definition is that the delay value for the tree is a physical quantity that a circuit designer can relate to immediately. Moreover, as will be shown later, this provides a natural extension into the problem of wire sizing under delay constraints. Note that our definition of the Elmore delay of a tree differs from the model in [2], where the user is required to identify the critical leaf nodes (we require no such user input), and a weighted sum of the Elmore delays to these leaf nodes is minimized.

The problem of minimizing the delay of an interconnect tree is a multiple objective optimization, since one must consider the delay at every leafnode as an objective. For such a problem, one is interested in finding Pareto critical points [4], which are the rough equivalent of minima in multicriterion optimization. As shown in [4], a weighted sums method such as that used in [2] cannot be used to characterize all Pareto critical points. The objective function that we consider for minimizing the tree delay is the maximum delay to a leafnode; this is a case of a minmax problem with all weights set to 1. Therefore, this can easily be extended to the general minmax problem with weights that can, unlike weighted sums, be used to characterize all Pareto points [4].

2.2 Properties of the General Wire Sizing Problem

We begin by proving a few results on the optimal wire sizes. Some of these results have been proved in [1, 2] for their delay model. We show here that some (but not all) of those results are also valid under the Elmore delay model that we have used. At this point, we make minimal assumptions, so that Theorem 1 below is true for any reasonable definition of optimality.

Definition 2 A wire width assignment f for a tree T is a n-tuple of numbers $[w_1, \cdots, w_n]$, where n is the number of wires in the interconnect tree, and w_i is the width of wire i.

Definition 3 Given a routing tree T, a wire width assignment f on T is a monotonic assignment if $w_p \geq w_c$ whenever wire S_p is an ancestor of wire S_c.

Definition 4 Given two wire width assignments f and f' on the same tree T, f dominates (is dominated by) f' if and only if $w_i(f) \geq w_i(f')$ ($w_i(f) \leq w_i(f')$) for all wires $i \in$ T.

Definition 5 A wire assignment f for a tree T is *suboptimal* if there exists another wire assignment f' for T, different from f, such that f dominates f', and the Elmore delay to *every* leaf node in T under assignment f' is no greater than that under assignment f.

Note that the definition of an optimal assignment here is open to interpretation under any formulation that uses the Elmore delay model, and that we have not restricted ourselves to a strict definition of optimality at this point. However, under any reasonable definition of optimality, Definition 5 must hold. The result in Theorem 1 below is, therefore, similar to, but more general than the analogous results presented in [1, 2] due to the more general definition of optimality that we use. It is specifically targeted to wire segments of equal length. Related statements for the case of wire segments of unequal length are made in Section 3.3.

Theorem 1 (The monotonicity property) Any nonmonotonic wire width assignment f^*, on wire segments of equal length, is suboptimal.

Proof Assume, for purposes of contradiction, that in the nonsuboptimal assignment f^*, there is a pair of edges (i.e., wires) $e_p = (v_{p1}, v_{q1})$ and $e_q = (v_{q1}, v_{q2})$. Thus, v_{p1} is the immediate ancestor of v_{q1}, which in turn, is the immediate ancestor of v_{q2} in the tree; if the width, w_p, of e_p is less than the width, w_q, of e_q, then f^* is nonmonotonic.

We now present three possible cases. In the first two, we show that an assignment f in which all wires have the same width as in f^*, except that the w_q is set to w_p, has a smaller delay than f^*. Note that if so, then f^* dominates f, and by Definition 5, f^* must be suboptimal. The above change alters only the resistance of branch e_q, and the capacitance at node v_{q2} in the RC network. In the third case, we present an argument that leads to a contradiction.

Let P_{q2} be the unique path from the root of the tree to v_{q2}. For any leaf node i of the tree, with path P_i from the root, we have three possibilities:

Case 1. (Figure 3(a)) If $P_i \cap P_{q2} = \emptyset$, the change does not affect the Elmore delay to node i. Therefore, we have shown the existence of a wire assignment, f, with the same delay to node i as f^*, that is dominated by f^*. Hence, f^* is suboptimal with respect to the delays to all such nodes i.

Case 2. (Figure 3(b) and (c)) If not, if $P_i \cap P_{q2} \neq P_{q2}$ (i.e., i is not a descendant of v_{q2}), then the only contribution of wire e_q to the delay to node i is as a capacitance. Since the capacitance of e_q in assignment f is smaller than that in assignment f^*, the Elmore delay to node i is now made smaller. Thus, f^* is suboptimal with regard to all nodes that fall under Case 2.

Case 3. (Figure 3(d)) If not, then $P_i \cap P_{q2} = P_{q2}$ (i.e., i is a descendant of v_{q2}), and wire e_q contributes to the delay to node i is as a resistance as well as as a capacitance. Let $e_1 = (s, v_1), e_2 = (v_1, v_2), \cdots, e_l = (v_{l1}, v_{p1}), e_p = (v_{p1}, v_{q1}), e_q = (v_{q1}, v_{q2}), e_r = (v_{q2}, v_{r1}), \cdots, e_x = (v_{m-1}, v_m = v_i)$ be the unique path from the root s to node i in the tree. If R_i is the resistance

of edge i, and C_j is the capacitance at node v_j, then the Elmore delay to node i is

$$R_1 \cdot (C_1 + C_2 + \cdots + C_m + C_{\text{offpath},1}) + R_2 \cdot (C_2 + C_3 + \cdots + C_m + C_{\text{offpath},2}) + \cdots + R_q \cdot (C_{q2} + C_{r1} + \cdots + C_m + C_{\text{offpath},q}) + \cdots + R_m \cdot (C_m + C_{\text{offpath},m}) \quad (3)$$

where $C_{\text{offpath},i}$ is the sum of all off-path capacitances driven by resistance R_i. Note that if the size of branch q is changed from w_q to w_p in assignment f_1, since all wire segments are of equal length, we may say that

$$\begin{aligned} C_{q2}(f_1) &= C_{q2}(f^*) - c \cdot (w_q - w_p) \text{ (where } c \text{ is a positive constant)}, \\ R_q(f_1) &= R_q(f^*) \cdot w_q / w_p, \\ \text{and } w_q &> w_p \end{aligned}$$

Also, note that the term $R_q \cdot C_{q2}$ is a independent of the width of wire q due to the relationships in Equation (1).

The change in the delay to node i caused by this is

$$\Delta_1 = -(R_1 + \cdots + R_l + R_p) \cdot c \cdot (w_q - w_p) + R_q \cdot (\frac{w_q}{w_p} - 1) \cdot (C_{r1} + \cdots + C_m + C_{offpath,q}) \quad (4)$$

Similarly, if we were to change the size of branch p from w_p to w_q, the change in the delay would be

$$\begin{aligned} \Delta_2 &= -(R_1 + \cdots + R_l) \cdot c \cdot (w_p - w_q) + R_p \cdot (\frac{w_p}{w_q} - 1) \cdot (C_{q2} + C_{r1} + \cdots + C_m + C_{offpath,p}) \\ &= -(R_1 + \cdots + R_l) \cdot c \cdot (w_p - w_q) + R_q \cdot (1 - \frac{w_q}{w_p}) \cdot (C_{q2} + C_{r1} + \cdots + C_m + C_{offpath,p}) \quad (5) \end{aligned}$$

Note that the value of Δ_1 is the same for *any* leafnode that is downstream of the branches p and q (since the sum of resistances in Equation (4) corresponds to the total resistance upstream of p, and the sum of capacitances corresponds to the total capacitance downstream of p). An identical statement can be made for Δ_2.

Since f^* is assumed not to be suboptimal, it must be true that $\Delta_1 \geq 0$ for some leafnode downstream of p and q, and and $\Delta_2 \geq 0$ for some leafnode downstream of p and q. However, we have already seen that all such leafnodes have identical values of Δ_1 (Δ_2). Consequently, we must have

$$\begin{aligned} \Delta_1 + \Delta_2 &\geq 0 \\ i.e., \quad -R_p \cdot c \cdot (w_q - w_p) + R_q \cdot (1 - \frac{w_q}{w_p}) \cdot (C_{q2} + C_{offpath,p} + C_{offpath,q}) &\geq 0 \quad (6) \end{aligned}$$

which is clearly a contradiction since $w_q > w_p$. This implies that f^* must be suboptimal for all nodes under Case 3.

Therefore, in all cases, we have shown that the nonmonotonic assignment f^* is a suboptimal solution. □

Although the above result has been proved for the single layer metal case, an analogous result may also be proved for multilevel interconnect, using the same proof technique.

Theorem 2 Let i be a leafnode, and let P_i be the path from the root node to i. Then the delay from the root to node i cannot be decreased by increasing any wire size that does not lie on P_i.

Proof The size of any wire that does not lie on P_i may either

(a) never appear in the Elmore delay expression for node i, in which case it does not affect the delay to i, or

(b) appear in the Elmore delay expression to i as a capacitive load, in which case increasing its size would cause the Elmore delay to i to increase. □

2.3 Limitations of Monotonicity

The result in Theorem 1, and the work in [1, 2] implicitly assume that the maximum allowable size for each wire is the same. This may not be so in all situations. For example, in congested routing regions, one may prefer to limit the maximum wire size, and hence monotonicity fails. However, the monotonicity property is not critical to the correctness of the work presented here. As will be shown in Section 3.3, monotonicity holds for the continuous sizing problem even under nonuniform wire lengths when the objective is to minimize the delay. Under delay constraints and nonuniform wire lengths, however, monotonicity does not hold, as can be shown through counterexamples.

2.4 Does Separability Hold for this Delay Model?

Definition 6[1]: A *single-stem subtree* at a node N is defined as a subtree rooted at N, with exactly one edge, called the *stem*, incident on N. Figure 4 illustrates this definition pictorially.

Under the delay models used in [1, 2], it is shown that the width of each wire depends only on the widths of its ancestors and descendants. As a result, if $T_{SS1}, T_{SS2} \cdots T_{SSk}$ are the single-stem subtrees rooted at node N, it has been proven under their delay models that the optimal wire width assignments for T_{SSi} can be determined independently of $T_{SSj}, j = 1 \cdots k, j \neq i$. This has been referred to as *separability*. By using this property, for a tree with n wires and r possible wire widths, algorithms of worst-case complexity $O(n^{r-1})$ have been proposed.

Under our delay model, however, we can show that separability does not hold; this is shown by the following counterexample. As a result, the algorithms in [1, 2] cannot be applied to solve this problem.

Example: Consider the simple example shown in Figure 5. Assume, for simplicity, the following:

- Each branch resistance is related to the branch width by the relation, $R_i \propto 1/w_i$.
- Each branch capacitance is related to the branch width by the relation, $C_i \propto w_i$.
- The capacitive load at each branch is as shown in the figure.
- The maximum allowable wire size is 15 units.
- The driver has a resistance of 1 unit.

The delays to the two leaf nodes are given by the expressions:

$$D_1 = K \cdot (1 + \frac{1}{x_1})(x_1 + x_2 + x_3 + C_1 + C_2) + \frac{1}{x_3}(x_3 + C_1)$$

$$D_2 = K \cdot (1 + \frac{1}{x_1})(x_1 + x_2 + x_3 + C_1 + C_2) + \frac{1}{x_2}(x_2 + C_2)$$

where K corresponds to a proportionality constant. By enumeration, it was found that the minimum delay to leaf node 1 occurs when $x_1 = 10, x_2 = 1, x_3 = 7$, the minimum delay to leaf node 2 corresponds to the situation where $x_1 = 10, x_2 = 6, x_3 = 1$, while the maximum of the two delays was minimized at $x_1 = 10, x_2 = 4, x_3 = 5$, which shows that the single-stem subtrees cannot be optimized independently of each other.

An alternative view is as follows: if we apply separability and set the width x_1 and compute widths x_2 and x_3 independently, and repeat this procedure for all allowable values of x_1, it is found that the obtained solution is $x_1 = 10, x_2 = 6, x_3 = 7$, which is not the optimum solution.

The reason for this is easy to see. The delay to node 2 depends on the widths x_1 and x_2, which act as both resistors and capacitors *and* the width x_3, which acts as a capacitive load. The optimal delay to node 2 implies that x_3 must be minimal; however, this could cause the delay to node 1 to be too large. At the optimum, there is a "balance" between the resistance of x_3 that causes a small delay to node 1, and the capacitance of x_3 that causes a small delay to node 2 as well. Thus, the sizing along the path from the root to node 2 is dependent on the sizes of branches that are off this path, and hence separability does not work. □

3 The Wire Sizing Problem

3.1 Statement of the Wire Sizing Problem

As mentioned earlier, several viable definitions of optimality are possible. We now address two problems:

- Wire sizing for minimum delay under maximum width constraints.
- Wire sizing under maximum delay and maximum width constraints.

The optimization problems associated with the above two definitions are:

Problem P1 minimize $(\max_{i \in leafnode(\mathrm{T})} d_i)$

subject to $w_j < w_{j,spec} \ \forall \ j = 1 \cdots n$.

Problem P2 minimize $\sum_{i \in \mathrm{T}} w_i$

subject to $d_i < D_{spec} \ \forall \ i \in leafnode(\mathrm{T})$ and $w_j < w_{j,spec} \ \forall \ j = 1 \cdots n$.

In the remainder of this work, we will address the two problems above.

For Problem **P1**, clearly, by Theorem 1, any nonmonotonic solution is suboptimal. The same property also holds for the Problem **P2**, since by Theorem 1, corresponding to any nonmonotonic feasible solution, there exists a monotonic feasible solution with a smaller objective function value.

3.2 Properties of the Continuous Wire Sizing Problem

Definition 7: The *continuous wire sizing problem* is the problem of finding optimal wire widths to solve the wire sizing problem, such that wire widths may take on any real value. This is in contrast to the *(discrete) wire sizing problem* where the wire widths are constrained to be integers.

Property 1: *The delay along any path of an RC tree is a posynomial [5] function of the sizes of wires in the tree.*

Property 2: *The continuous wire sizing problems* **P1** *and* **P2**, *stated in Section 3.1, are unimodal, i.e., any local minimum of these optimization problems is a global minimum.*

To observe this, note that the simple transformation,

$$(w_i) = (e^{x_i}), \tag{7}$$

transforms any posynomial function of the w_i's to a convex function of the x_i's [5]. Hence, under this transformation, for both problems, the objective function as well as the constraints are convex. As a consequence of the fact that the mapping function is one-to-one, it is easy to see that the optimization problems **P1** and **P2** are unimodal.

It may be worthwhile to caution the reader here that it is only the *continuous* wire sizing problem that is unimodal; the (discrete) wire sizing problem is combinatorial, and no such statements can be made about it. However, a solution to the continuous wire sizing problem gives a lower bound on the solution to the discrete problem.

3.3 Monotonicity and Nonuniform Wire Segment Lengths

The following is an extension of Theorem 1 for wire segments of nonuniform length; however it is not a generalization of Theorem 1 in two respects: firstly, we have only been able to prove its applicability to the continuous wire sizing problem, and secondly, this result is not valid for the problem of minimizing area under a delay constraint.

Although the proof is not applicable to the discrete wire sizing problem, based on our experience on the relation between the continuous and discrete wire sizing problems, we believe that a wire sizing solution restricted to monotonic solutions only would give close to optimal, if not optimal solutions.

Theorem 3: For the continuous sizing problem, any nonmonotonic wire width assignment f^* is suboptimal.

Proof The proof proceeds as in the proof of Theorem 1. Cases 1 and 2 are dealt with in a similar manner. The treatment for Case 3 is different, and is dealt with here. Unless specifically mentioned, all of the terminology is the same as that in Theorem 1.

Let the resistance and capacitance, respectively, of a wire of length l and width w be given by $\rho\frac{l}{w}$ and $\alpha \cdot l \cdot w$, where ρ and α are technology-dependent constants.

Referring back to Figure 3(d), if i is a descendant of v_{q2}, then we can write the delay to node i as a function of w_p and w_q as

$$D_i(w_p, w_q) = R_{prev} \cdot (\alpha \cdot l_p w_p + \alpha \cdot l_q w_q + C_{rest}) + \rho\frac{l_p}{w_p} \cdot (\alpha \cdot l_p w_p + \alpha \cdot l_q w_q + C_{rest}) + \rho\frac{l_q}{w_q} \cdot (\alpha \cdot l_q w_q + C_{rest}) + K \qquad (8)$$

where R_{prev} is the sum of all resistances on the path from the root node to node v_{p1}, C_{rest} is the

sum of all downstream capacitances beyond node v_{q2}, and K corresponds to the delay terms that are independent of w_p and w_q.

For any downstream node covered by Case 3, the sensitivity of D_i to w_p and w_q is equal and is given by:

$$\frac{\partial D_i}{\partial w_p} = R_{prev}\alpha l_p - \rho\alpha l_p l_q \frac{w_q}{w_p^2} - \rho C_{rest}\frac{l_p}{w_p^2} \tag{9}$$

$$\frac{\partial D_i}{\partial w_q} = R_{prev}\alpha l_q + \rho\alpha l_p l_q \frac{1}{w_p} - \rho C_{rest}\frac{l_q}{w_q^2} \tag{10}$$

At the minimum, $\frac{\partial D_i}{\partial w_p} = -\lambda_p$ and $\frac{\partial D_i}{\partial w_p} = -\lambda_q$, where λ_p and λ_q are the Lagrange multipliers associated with the interval constraints for w_p and w_q, respectively. Note that the maximum allowable wire width for each segment must be uniform, as stated in Section 2.3. At the point where the delay is minimum, the values of λ_p and λ_q are nonnegative for the upper bound constraint and nonpositive for the lower bound constraint [6]. We have the following relation:

$$-l_q\frac{\partial D_i}{\partial w_p} + l_p\frac{\partial D_i}{\partial w_q} = l_q\lambda_p - l_p\lambda_q,$$

$$\text{i.e.} \quad \rho\alpha l_p l_q\left(\frac{l_p}{w_p} + \frac{l_q w_q}{w_p^2}\right) + \rho C_{rest} l_p l_q\left(\frac{1}{w_p^2} - \frac{1}{w_q^2}\right) = l_q\lambda_p - l_p\lambda_q. \tag{11}$$

Notice that on the left hand side, the first term is always strictly positive, and the second term is strictly positive since $w_p < w_q$ by assumption. Therefore, the left hand side is always strictly positive.

We now consider the following possibilities:

Case a. When the maximum width constraint on segment q is inactive, i.e., $w_q \neq W_{max}$, then $\lambda_q = 0$. Note that w_q cannot equal W_{min} since $w_q > w_p \geq W_{min}$. Since $w_p < w_q$, we must have either (i) $w_p \neq W_{min}$, in which case $\lambda_p = 0$ and the right hand side of Equation (11) is zero, or (ii) $w_p = W_{min}$, in which case $\lambda_p \leq 0$ and the right hand side is nonpositive. Since the left hand side of Equation (11) is positive, this leads to a contradiction in either case.

Case b. When the maximum width constraint on segment q is active, then $\lambda_q > 0, \lambda_p = 0$, and $w_q = W_{max} \geq w_p$. Therefore, if (i) $w_p \neq W_{min}$, in which case $\lambda_p = 0$ and the right hand side of Equation (11) is negative, or (ii) $w_p = W_{min}$, in which case $\lambda_p \leq 0$ and the right hand side is nonpositive. Thus, in either case, the right hand side of Equation (11) is negative, leading to a contradiction.

Therefore, the assumption of nonmonotonicity at the minimum is incorrect. □

4 Solving the Continuous Wire Sizing Problem

We now present two alternatives to solving the continuous wire sizing problem. The first method is a sensitivity-based heuristic that has quick runtimes, but is not guaranteed to be optimal. However, as will be seen from our experimental results the quality of the solution is reasonably good. The second method is a convex optimization technique that finds the *exact* solution to the continuous optimization problem, at the expense of larger runtimes.

4.1 A Sensitivity-based Algorithm for Wire Sizing

Since the enumerative solution to the wire sizing problem with n wires and r permissible sizes is of complexity $O(r^n)$, we propose an efficient heuristic for solving the problem.

The heuristic presented here is efficient and sensitivity-based; such heuristics have been used successfully in finding solutions to posynomial programming problems, for example, in the transistor sizing algorithm, TILOS [7]. The heuristic first finds a solution to the continuous wire sizing problem, and then finds the discrete solution by using a sensitivity-based mapping algorithm to round off wire sizes to the next higher or lower integer. As shown in Section 6, this causes an insignificant degradation in the quality of the solution.

Rather than using a heuristic to solve the continuous problem, one could use an exact optimization algorithm, such as the convex programming algorithm used in [8], the method of Lagrange multipliers, etc. However, since the number of variables, which equals the number of RC sections in the wire, may be very large, the employment of any such algorithm would be computationally expensive.

The pseudo-code representing the algorithm WIMIN is shown below:

```
BEGIN ALGORITHM WIMIN()
 F = bumping factor;
 while (stopping criterion not met)
   current_leaf_node = leaf node with the
         largest delay violation;
   maxsensitivity = 0;
   maxsensitivity_wire = -1;
   for each wire i that is an ancestor
      of current_leaf_node
     if F * width(i) > width(predecessor[i])
       continue;
```

```
    if sensitivity S_i < maxsensitivity
      maxsensitivity = S_i;
      maxsensitivity_wire = i;
   if (maxsensitivity_wire == -1)
    /* minimum delay has been found */
    exit;
   width(maxsensitivity_wire) *= F;
 MAP();
END ALGORITHM WIMIN()
```

In each iteration, the leafnode with the largest violation is identified; this will be referred to as the current leaf node. We define the sensitivity, S_i of wire i as

$$S_i = \frac{Delay(\text{wire } i \text{ size } = F \cdot w_i) - Delay(\text{wire } i \text{ size } = w_i)}{(F-1) \cdot w_i} \tag{12}$$

where $Delay$ is the delay from the root node to the current leaf node, and F is a number just larger than 1. (Although the exact sensitivity of the delay function could have been computed here, since we will be taking steps of discrete sizes, it is more beneficial to compute the sensitivity as a finite difference.) By Theorem 2, the delay of the current leafnode can only be decreased by increasing the sizes of wires that lie on the path between the root node and that leafnode, i.e., the sensitivities of all other wires is positive. The sensitivity of each such wire is identified, and the size of the single wire with the minimum negative sensitivity (i.e., the negative sensitivity that has the largest magnitude) is bumped up by multiplying it by the same constant factor, $F > 1$, as in Equation (12) (typical values of F are 1.2 or 1.5). This ensures that the delay to the current leafnode is reduced in every iteration.

Note that when the monotonicity property holds, it is unnecessary to compute the sensitivity for any wire for which the bumping operation violates monotonicity.

The process continues until no wire has a negative sensitivity, which gives the solution to the unconstrained Problem **P1**, or until the delay specifications at all leaf nodes are met, which provides the solution to the constrained Problem **P2**. This is the stopping criterion alluded to in the pseudo-code.

4.2 A Formal Convex Optimization Algorithm for Wire Sizing

As mentioned in Section 3.2, the transformation in Equation 7 maps the continuous wire sizing problem onto an equivalent convex optimization problem. Here, we employ a rigorous mathematical programming algorithm for convex optimization that was proposed in [9]; implementational details are provided in [10]. The algorithm works in an n-dimensional space, where n is the number of variables. In this paper, we use this approach primarily as a form of validating our sizing heuristic; as mentioned earlier, the solution to the continuous problem provides a lower bound on the optimal discrete solution.

Initially, a polytope $\mathcal{P} \in \mathbf{R}^n$ that contains the optimal solution, $\mathbf{x}_{opt}$, is chosen. The objective of the algorithm is to start with a large polytope, and in each iteration, to shrink its volume while keeping the optimal solution, $\mathbf{x}_{opt}$, within the polytope, until the polytope becomes sufficiently small. The initial polytope $\mathcal{P}$ may, for example, be selected to be an n-dimensional box described by the set

$$\{\mathbf{x} \mid \log_e(w_{i,min}) \leq x_i \leq \log_e(w_{i,max})\} \tag{13}$$

where $w_{i,min}$ and $w_{i,max}$ are the minimum and maximum wire sizes, respectively, of the i^{th} wire. The algorithm proceeds iteratively as follows.

Step 1 A *center* $\mathbf{x}_c$ deep in the interior of the current polytope $\mathcal{P}$ is found.

Step 2 An *oracle* is invoked to determine whether or not the center $\mathbf{x}_c$ lies within the feasible region $\mathcal{F}$. This may be done by verifying that all of the constraints of the optimization problem are met at the point $\mathbf{x}_c$.

The oracle, i.e., the feasibility check, may be performed by first applying Theorem 1 to check for monotonicity (when it holds); a nonmonotonic wire assignment is automatically infeasible. If the wire assignment is monotonic, then a full delay calculation must be carried out to check whether the wire assignment satisfies the delay constraints. Note that for the unconstrained problem, any assignment of wire sizes lies in the feasible region.

If the point $\mathbf{x}_c$ lies outside $\mathcal{F}$, it is possible to find a *separating hyperplane* passing through

$\mathbf{x}_c$ that divides $\mathcal{P}$ into two parts, such that $\mathcal{F}$ lies entirely in the part satisfying the constraint

$$\mathbf{c}^T\mathbf{x} \geq \beta \tag{14}$$

$$\text{where } \mathbf{c} = -[\nabla g_p(\mathbf{x})]^T \tag{15}$$

is the negative of the gradient of a violated constraint, g_p, and

$$\beta = \mathbf{c}^T\mathbf{x}_c. \tag{16}$$

The separating hyperplane above corresponds to the tangent plane to the violated constraint.

If the point $\mathbf{x}_c$ lies within the feasible region $\mathcal{F}$, then there exists a hyperplane (14) that divides the polytope into two parts such that $\mathbf{x}_{opt}$ is contained in one of them, with

$$\mathbf{c} = -[\nabla f(\mathbf{x})]^T \tag{17}$$

being the negative of the gradient of the objective function, and β being defined by (16) once again.

Step 3 In either case, the constraint (14) is added to the current polytope to give a new polytope that has roughly half the original volume.

Step 4 The process is repeated until the polytope is sufficiently small.

Further implementational details of the algorithm are provided in [10]. The computational complexity of this algorithm is $O(n^{2.5})$ where n is the number of design variables.

5 Mapping the Continuous Solution to Discrete Sizes

The mapping algorithm is shown below:

```
BEGIN ALGORITHM MAP()
 Mark all wires as unprocessed;
 Mark all leafnodes as unprocessed;
 while (all leafnodes not processed)
   current_leaf_node = unprocessed leaf node
         with the largest delay;
   for each unprocessed wire segment i that is an
```

```
        ancestor of current_leaf_node
    if (width(i) is an integer) continue;
    w_{i+} = ⌈width(i)⌉
    w_{i-} = ⌊width(i)⌋
    if (| delay(w_{i+}) − delay(width(i)) |
        <| delay(w_{i-}) − delay(width(i)) |)
      width(i) = w_{i+};
    else
      width(i) = w_{i-};
END ALGORITHM MAP()
```

It starts from the leafnode, L, with the largest delay, and processes each wire on the path between node L and the root node. If the size of the current wire is an integer, its size remains unchanged. If not, the change in the delay to L caused by changing the wire size to the closest higher (lower) integer, w_{i+} (w_{i-}) is computed, and one that creates a smaller delay fluctuation is selected. L is now marked as "processed" and the algorithm proceeds iteratively with the unprocessed leafnode that has the largest delay. Note that in the mapping phase, once a wire has been processed, its size remains unchanged, and that each wire is considered only once.

6 Experimental Results

6.1 Results on Several Examples

The heuristic algorithm and the convex programming algorithm are implemented as C programs named WIMIN (WIre-size MINimizer) and COSI (Convex Optimization for Sizing Interconnect), respectively, on a DECstation 5000/133. Both WIMIN and COSI were run on twelve test networks, which are described in Table 1. The technology parameters used are those used in advanced MCM designs [1, 2], and are described in Table 2.

The results of COSI are not separately shown, since there was virtually no noticeable difference between the quality of the results of the two algorithms. The CPU times for COSI were of the order of 15s for problems Intct1-Intct5, 1 minute for problems Intct6-Intct9, and 3 minutes for problems Intct10-Intct12. Hence, unless otherwise specified, the results presented here correspond

Table 1: Description of the circuit examples.

Circuit	Unit Grid Size	Number of Wires	Number of Leaf Nodes
Intct1	1000	100	17
Intct2	1000	100	13
Intct3	1000	100	16
Intct4	1000	100	20
Intct5	1000	100	4
Intct6	200	500	16
Intct7	200	500	16
Intct8	200	500	16
Intct9	100	1000	9
Intct10	100	1000	16
Intct11	100	1000	16
Intct12	100	1000	16

Table 2: Technology parameters based on advanced MCM designs.

Technology:	Multichip Modules
Driver Resistance:	25 Ω
Unit Wire Resistance:	0.008 Ω/μm
Loading capacitance:	1000 fF
Unit Wire Capacitance:	0.060 fF/μm
Total area:	100 mm $\times$ 100 mm

to the results from WIMIN. It was seen that unlike WIMIN, the runtimes of COSI do not scale very well with large problem sizes.

Experimental results for Problem **P1**, in which the wire sizes that correspond to the minimum interconnect delay are presented for each of the test circuits, are shown in Table 3. For WIMIN, the value of the multiplicative factor, F, is set to 1.2. In our implementation, an additive factor was tried instead of a multiplicative factor; however, this was found to give poorer results. This may be attributed to the fact that wires near the source need to be sized more than those near the leaf nodes, and the general profile of the correctly sized wires resembles a geometric progression, rather than an arithmetic progression.

Table 3: Results of Minimizing Interconnect Delay.

Circuit	Unsized		Maxsize =2			Maxsize = 6				
			Minimum delay			Minimum delay			$D_{spec} = 1.15 \times D_{min}$	
	Cost	Delay (ns)	Cost	Delay	CPU	Cost	Delay	CPU	Cost	CPU
Intct1		1.622	118	1.161	1.1s	161	0.931	2.3s	128 (26%)	0.9s
Intct2		2.526	128	1.652	0.8s	189	1.182	1.3s	143 (32%)	0.6s
Intct3	99	2.710	120	1.787	0.8s	182	1.186	1.7s	144 (26%)	0.5s
Intct4		1.759	120	1.288	1.2s	180	1.087	2.4s	123 (46%)	0.9s
Intct5		2.231	115	1.650	0.4s	223	1.214	0.6s	163 (37%)	0.4s
Intct6		0.872	551	0.715	5.0s	672	0.633	13.1s	527 (28%)	1.4s
Intct7	499	1.002	565	0.774	5.2s	739	0.664	12.0s	552 (34%)	2.1s
Intct8		1.297	609	0.935	6.0s	864	0.740	13.1s	643 (35%)	3.1s
Intct9		1.236	700	0.865	4.0s	1072	0.689	4.8s	732 (46%)	3.1s
Intct10		1.540	1108	1.132	11.1s	1376	0.903	29.2s	1168 (18%)	5.8s
Intct11	999	2.387	1226	1.601	15.9s	1712	1.123	34.1s	1385 (24%)	7.4s
Intct12		3.102	1178	2.012	14.7s	2033	1.369	27.4s	1529 (33%)	7.4s

For each circuit, we show the cost and delay of the unsized circuit, i.e., the circuit in which all wires have unit width. As mentioned earlier, the cost is taken as the sum of wire sizes. The next two three-column sets show the cost, RC delay, and the execution time for the optimization, when the maximum allowable wire size is 2 and 6, respectively. Note that the computation time of the algorithm is very reasonable. With some increase in wire sizes, it can be seen that the interconnect delay can be improved significantly.

The bulk of the CPU time is incurred by the continuous optimization problem, and only a small fraction (under 10%) is attributable to the mapping phase. The run times are reasonable even for large circuits.

In the last two columns of Table 3, for the case when the maximum allowable wire size is 6, the delay constraint is relaxed to 15% over the minimum delay, and problem **P2** is solved. We apply a uniform timing constraint on each leaf node of the tree. Note that the nature of the algorithm is such that there may be different delay specifications at each of the leaf nodes for Problem **P2**, and not a uniform specification. For no reason in particular, however, we restrict ourselves to a

uniform timing constraint for all leaf nodes in this section. It must be stressed, however, that the algorithm is general enough to handle nonuniform timing constraints too. The corresponding cost and run times are shown. The figures in brackets under the "Cost" column represent the % cost reduction compared to the minimum delay case. Improvements of as much as 46% are seen. Due to the paucity of routing resources on a chip, this area improvement is very significant.

Next, we present results on Problem **P2**, i.e., on minimizing interconnect delay under timing constraints, graphically on three specific circuits in Figure 6. This picture serves to illustrate the area-delay tradeoff made during wire sizing. As before, the value of the factor F is set to 1.2.

The results plotted in Figure 6 show the true utility of using the problem formulation **P2**. It is observed that the interconnect area overhead required to achieve the minimum possible delay is extremely high, for the last fraction of delay reduction. While some of this is attributable to suboptimality of the sensitivity-based algorithm, the same characteristics were found to hold when the factor F was very close to 1, when the solution is close to optimal. This explains why, in Table 3, substantial improvements in the cost functions are achieved when the constraints are relaxed by a small amount.

The graphs in Figure 6 show a comparison between the discrete solution provided by WIMIN and the continuous solution provided by COSI. Note that COSI's continuous solution is the exact solution to the continuous optimization problem, and is a lower bound for the optimal discrete solution (the slight discrepancies where the COSI solution is apparently more costly than the WIMIN solution in some cases are insignificant and are caused by the convergence criterion for COSI). As can be seen from the Figure, the solution provided by WIMIN is nearly optimal. In fact, it is worth noting here that the continuous solution may not be achievable if one is restricted to discrete sizes, and hence the solution from WIMIN may well be *the* optimal solution.

A comment about the accuracy of this optimization is in order. The continuous sizing solution is, by the construction of the algorithm, less than the specification. However, the discrete solution delay is not always so, and may provide a solution that has slightly larger delay than the specification. This is not critical, since the Elmore delay model is known to be accurate only up

to 10 or 20 %, whereas the discrepancy between the discrete solution delay and the specification is less, and some is attributable to discretization noise.

6.2 Comparison with the Results of the Approach in [2]

A comparison of this approach with the method in [2] is shown in Figure 7. The algorithm is applied here to a line of length 100 mm, divided into twenty segments. An dynamic programming procedure was carried out on the sizes of the twenty segments for maximum wire segment widths of 1,2,3,4,5 and 6 μm, respectively[1], to find the wire assignment that gives the minimum delay. These points are plotted on the dotted line in Figure 7(a). The lower darkened line corresponds to the area delay tradeoff generated by our approach using a maximum wire segment width of 6 μm.

Although the curves seem to be close to each other in Figure 7(a), the true story is told in Figure 7(b). Several observations can be made about the comparison:

1. The approach in [2] gives one point for every maximum width. This number of points is substantially less than the number achievable using our approach, and our approach gives a much smoother area-delay tradeoff curve with many allowable selections on the curve. Notably, the number of points in the flatter area of the curve (where the solutions make good engineering sense unlike those in the steep part) is significantly smaller for [2]. This is significant since the points for [2] are distant from each other in this region.

2. The area savings using our approach, shown in Figure 7(b), is significantly large in all cases in the region of interest.

3. The cost/benefit ratio increases rapidly as one moves towards lower delays, and hence it is the part of the curve to the right of Delay = 2.3ns that corresponds to viable and sensible solutions that a designer would use. Note that the solutions to the left are probably even more expensive than the cost function reflects. Our cost function takes the area as the sum of

[1] Note that the method in [2] performs an enumeration guided by monotonicity, separability, and the use of upper and lower bounds, for computational efficiency.

the wire segment areas; however, large disparities in wire segment sizes makes routing more difficult, and this is not reflected in the cost.

To verify the accuracy of our approach, we tried to achieve the minimum delay using our approach, for different values of the maximum wire width. Since our method continues to reduce the delay as long as possible, and discontinues its efforts when no further delay reduction is possible, this may be achieved by setting the delay target to zero. It was seen that in every case, varying the maximum width from 1μm to 6μm, the solution given by our approach was either exactly the same, or insignificantly different from the enumerated solution.

7 Concluding Remarks

A new algorithm for interconnect sizing has been described in this paper. The contributions of this work are as follows:

- Wire sizing is performed under a delay model that does not require the user to specify the critical leaf nodes, and that will work even in the general case when the monotonicity property does not hold.
- The problem of obtaining the optimal wire sizes under delay constraints is addressed for the first time. Previous work in the literature has only addressed the problem of wire delay minimization. A smooth area-delay tradeoff in the sizing operation is shown.
- It is shown experimentally that achieving the absolute minimum delay for a net involves a wasteful use of resources; instead, a delay target of even 15% over the minimum delay provides a good engineering solution with a substantial reduction in the net delay with only a small area overhead.

Although we have presented our results on the single level interconnect case, the sizing algorithm is trivially extendable to the multilevel case; the proportionality constants for the resistances and

capacitances will change in the multilevel interconnect case, but the posynomial nature of the functions will be maintained.

The algorithm is also easily extendable to sizing buses, where the problem can be stated as minimizing the wire area, subject to bidirectional constraints between the leafnodes of a routing net. The convex programming properties continue to hold, and the same solution method can be extended.

Acknowledgements

The author would like to thank Prof. Jason Cong and the anonymous reviewers for their constructive comments on the manuscript. Thanks are also due to Piyush Sancheti for helping implement portions of the convex programming-based algorithm.

References

[1] J. Cong, K.-S. Leung, and D. Zhou, "Performance-driven interconnect design based on distributed RC model," Tech. Rep. CSD-920043, UCLA Computer Science Department, Oct. 1992. (Abbreviated version appeared in the *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 606–611, 1993).

[2] J. J. Cong and K.-S. Leung, "Optimal wiresizing under Elmore delay model," *IEEE Transactions on Computer-Aided Design*, vol. 14, pp. 321–337, Mar. 1995.

[3] J. Rubenstein, P. Penfield, and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Transactions on Computer-Aided Design*, vol. CAD-2, pp. 202–211, July 1983.

[4] R. K. Brayton, G. D. Hachtel, and A. L. Sangiovanni-Vincentelli, "A survey of optimization techniques for integrated-circuit design," *Proceedings of the IEEE*, vol. 69, pp. 1334–1362, Oct. 1981.

[5] J. Ecker, "Geometric programming: methods, computations and applications," *SIAM Review*, vol. 22, pp. 338–362, July 1980.

[6] D. G. Luenberger, *Linear and Nonlinear Programming*. Reading, Massachusetts: Addison-Wesley, 2nd ed., 1984.

[7] J. Fishburn and A. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," in *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 326–328, 1985.

[8] S. S. Sapatnekar, V. B. Rao, and P. M. Vaidya, "A convex optimization approach to transistor sizing for CMOS circuits," in *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 482–485, 1991.

[9] P. M. Vaidya, "A new algorithm for minimizing convex functions over convex sets," *Proc. IEEE Foundations of Computer Science*, pp. 332–337, Oct. 1989.

[10] S. S. Sapatnekar, V. B. Rao, P. M. Vaidya, and S. M. Kang, "An exact solution to the transistor sizing problem for CMOS circuits using convex optimization," *IEEE Transactions on Computer-Aided Design*, vol. 12, pp. 1621–1634, Nov. 1993.

List of figures

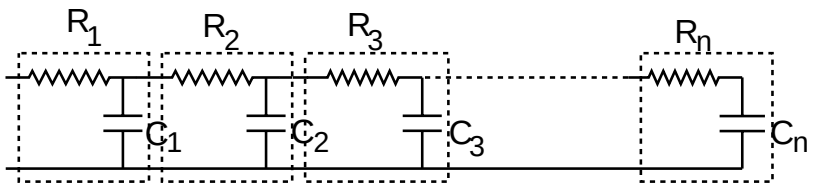


Figure 1: RC model of interconnect

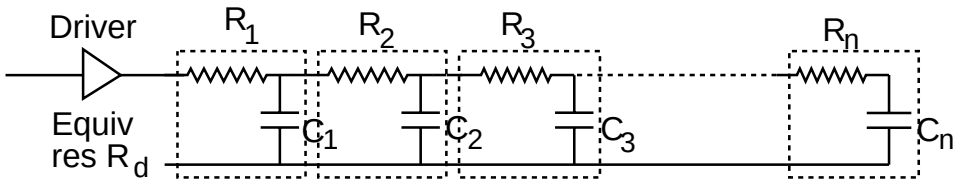


Figure 2: RC line driven by a gate

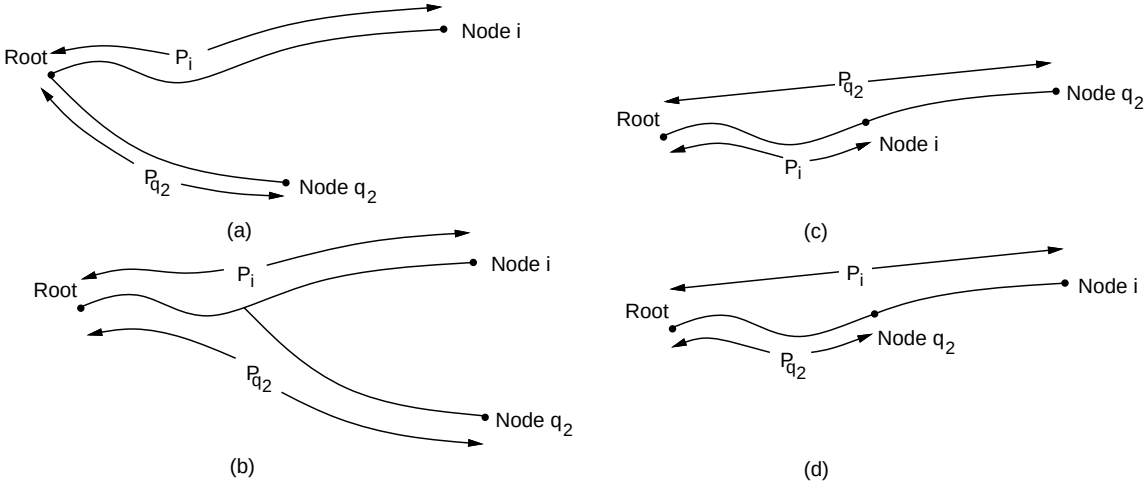


Figure 3: Proof of Theorem 1

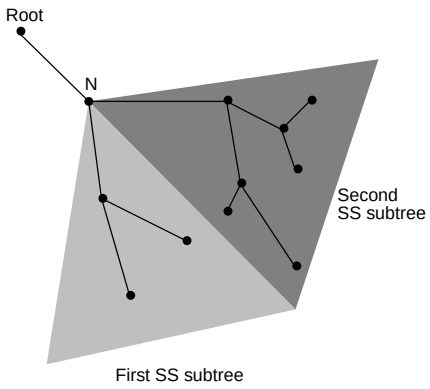


Figure 4: Single-stem subtrees rooted at node N

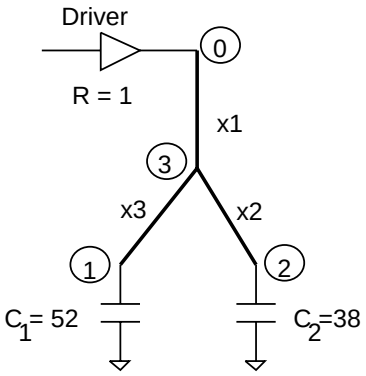


Figure 5: Counterexample for separability

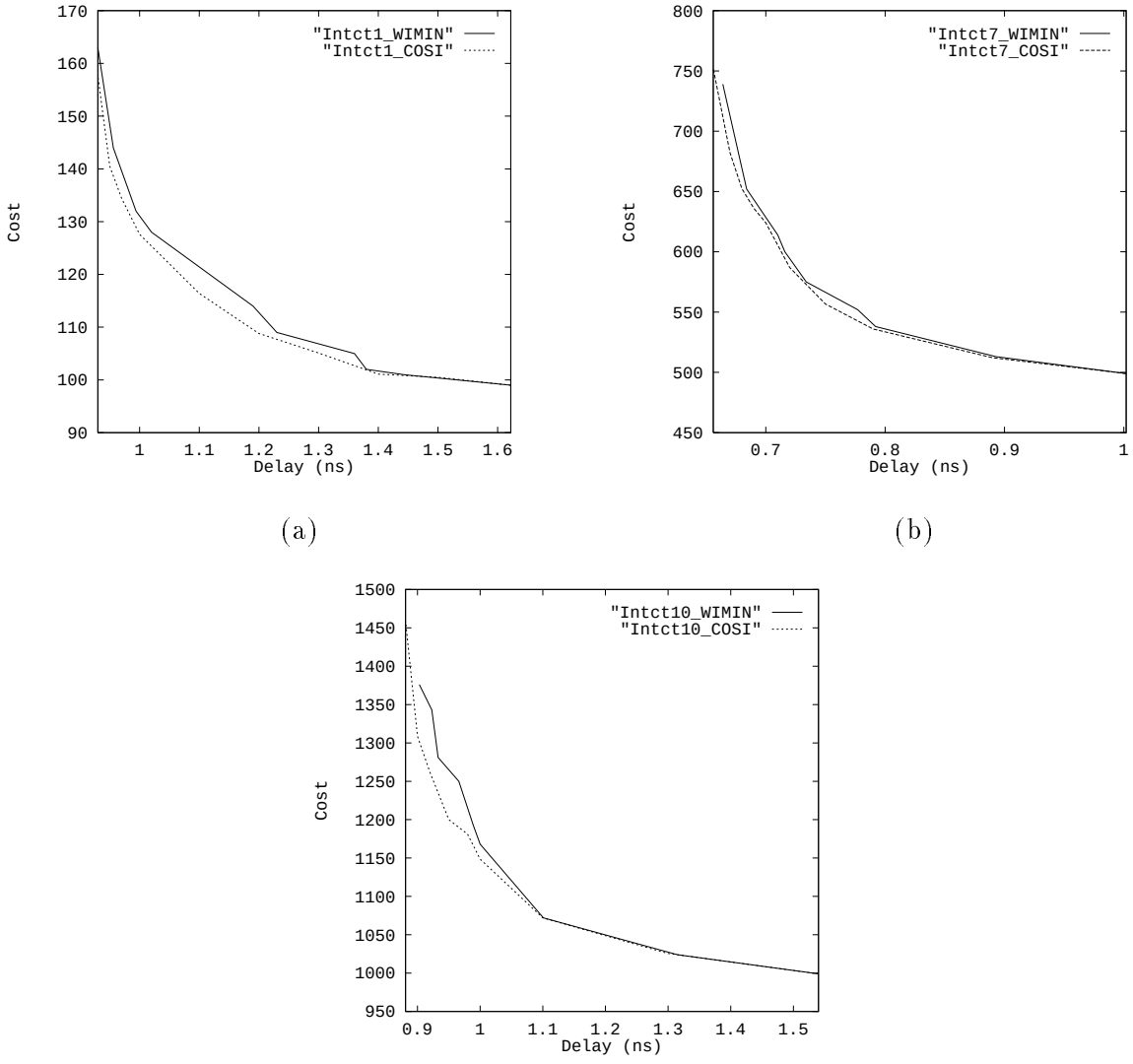
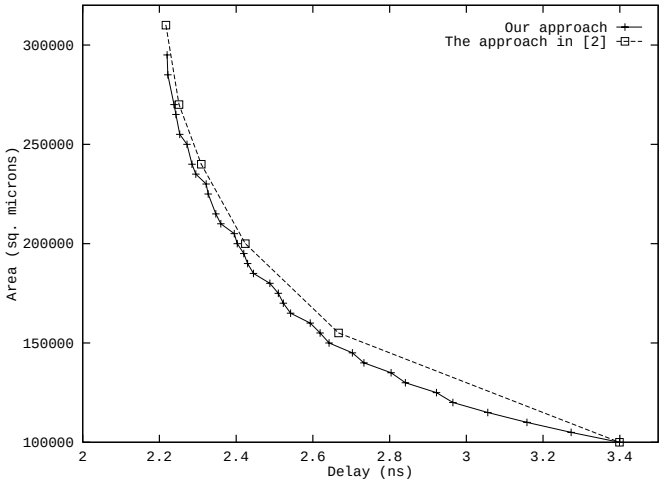
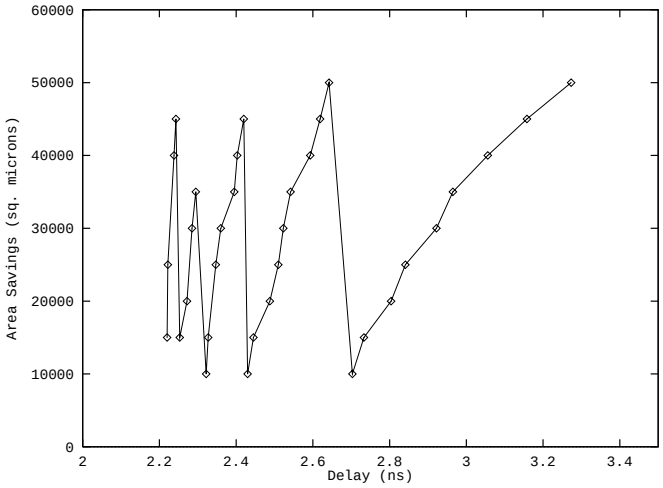


Figure 6: Cost vs. delay for (a) Intct1 (b) Intct7 (c) Intct10



(a)



(b)

Figure 7: (a) Area-delay curves for the two approaches (b) Cost savings using our approach