A Sequential Quadratic Programming Approach to Concurrent Gate and Wire Sizing*

Noel Menezes, Ross Baldick, and Lawrence T. Pileggi†
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712-1084

Abstract

With an ever-increasing portion of the delay in high-speed CMOS chips attributable to the interconnect, interconnect-circuit design automation continues to grow in importance. By transforming the gate and multilayer wire sizing problem into a convex programming problem for the Elmore delay approximation, we demonstrate the efficacy of a sequential quadratic programming (SQP) solution method. For cases where accuracy greater than that provided by the Elmore delay approximation is required, we apply SQP to the gate and wire sizing problem with more accurate delay models. Since efficient calculation of sensitivities is of paramount importance during SQP, we describe an approach for efficient computation of the accurate delay sensitivities.

1 Introduction

Recognizing that large reductions in interconnect delay can be achieved by selectively widening the branches of the interconnect tree (wire sizing), different approaches to wire sizing have been proposed [1, 2, 3, 4]. The approach in [1, 2] uses the properties of monotonicity, separability, and dominance which apply to the Elmore delay [5, 6] to determine the optimal wire sizing solution. Recognizing that monotonicity and separability do not apply to the Elmore delay under certain conditions, a sensitivity-based wire sizing heuristic is presented in [3]. The approach of [4] uses an approximate relationship between the moments of the RC tree and the delay to guide a non-linear least-squares minimizaton to find the gate and wire sizes that will yield the target delays at the critical sinks of the interconnect tree. However, there is no guarantee of optimality in the final wire sizing solution since the delay constraints at the critical sinks are formulated as equality constraints. The approach of [4] differs from the other approaches in that it uses accurate delays computed from RICE [7] (instead of the Elmore delay approximation) and a gate delay model which captures the complex interaction between the CMOS gates and the corresponding RC loads [8].

In [3], the posynomiality of the Elmore delay in terms of the widths of the RC interconnect tree (first mentioned in [9]) was used to convert the wire sizing problem into a convex program under a simple transformation. However, the "one-wire-at-a-time" TILOS heuristic of [3] does not fully exploit the convexity of the transformed wire sizing problem. In this paper, we demonstrate the efficacy of a sequential quadratic programming approach [10] to the gate and wire sizing problem under the Elmore delay approximation when a simple fixed-resistor gate delay model is used. For a simple gate delay model, the concurrent gate and wire sizing problem applied to *paths* is also posynomial under the Elmore delay model [9]. Therefore, SQP can also be applied to path delay optimization.

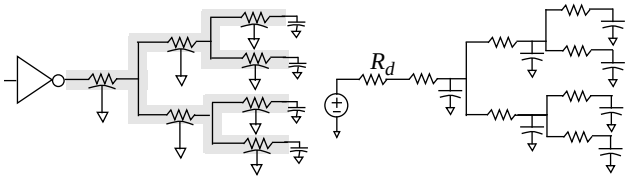
However, the error resulting from a fixed-resistor driver model in conjunction with the Elmore delay can be significant [11]. Also, during wire sizing the load on the driver changes significantly and should be reflected in a load-dependent gate delay model. Moreover, a fixed-resistor gate delay model when used in conjunction with the Elmore delay approximation implicitly assumes that the load on the gate is accurately modeled by the total interconnect capacitance, an assumption that is especially invalid for large RC-interconnect loads [12]. Furthermore, input transition time effects must be considered for proper path sensitivities. With this in mind, for situations where extreme accuracy is required we apply SQP to the gate and wire sizing problem using a gate delay model [8] which captures the interaction between the gate and the RC load with accurate delays computed by RICE. We also demonstrate the utility of our SQP approach for optimization of RC meshes.

* This work was supported in part by the Semiconductor Research Corporation under contract 95-DJ-343, the National Science Foundation under contract MIP-9157263, and IBM Corp.

† Formerly Lawrence T. Pillage. As of Jan. 1996, he will be with Carnegie Mellon University, Dept. of ECE, Pittsburgh, PA 15213.

2 The concurrent gate and wire sizing problem

Consider a gate with its associated fanout tree as shown in Figure 1a. Modeling the gate by a resistor R_d driven by



(a) (b)

Figure 1: a) A driver and its fanout tree b) Equivalent RC tree model.

a step voltage source and replacing every interconnect branch with its L-section model – lumping techniques allow for accurate and efficient analysis of complex multi-level interconnect structures [13] – we obtain the RC circuit shown in Figure 1b. The interconnect branches may lie on different layers and, therefore, exhibit different parasitics. Given target delays at certain fanout nodes of interest (*critical sinks*) in the circuit, the concurrent gate and wire sizing problem involves determining the optimal gate size and wire widths for the branches of the interconnect tree which will yield the target delays. The objective function is usually defined in terms of the area required for the circuit layout. However, given the current thrust in low-power design, power dissipation may need to be considered too. Furthermore, since certain branches of the interconnect tree may pass through congested areas of the chip the *routability* of these branches should be factored into the objective function of the optimization. Lastly, for a feasible routing solution the wires are subject to upper bounds on their allowable widths.

Let $W = \{w_1, w_2, ..., w_N, w_{N+1}\}$ be the vector of the widths of the N branches of the interconnect tree and the gate size, $w_g (= w_{N+1})$, and $a(W)$ the circuit area corresponding to W. If the delay at a critical sink j is denoted by $t_d^j(W)$ and the delay constraint at j by $\hat{t}_d^j$ the concurrent driver and wire sizing problem can be stated as:

$$\begin{aligned} &\text{minimize } a(W) = \sum_{i=1}^{N} \gamma_i l_i w_i + g_1 w_g + g_0, \\ &\text{subject to } t_d^j(W) < \hat{t}_d^j, \qquad j = 1...M, \\ &\qquad \underline{w}_i \le w_i \le \overline{w}_i, \qquad i = 1...N+1. \end{aligned} \tag{1}$$

In (1), it is assumed that $g_1 w_g + g_0$ is the area occupied by the driver – a reasonable assumption for CMOS gates. γ_i (≥ 0) denotes the *congestion coefficient* for interconnect branch i which is determined by its routability. $\underline{w}_i$ and $\overline{w}_i$ correspond to the lower and upper bounds on the width of branch i, w_i. l_i denotes the length of branch i while M denotes the number of critical sinks.

2.1 The posynomiality of the Elmore delay

The delay at any node in an RC tree can be approximated by the Elmore delay [5, 6] which is the first moment of the impulse response and a known upper bound [14] on the 50% delay. If it is assumed that the output resistance of the driver is inversely proportional to the driver size, that is $R_d = R_g / w_g$, then for the RC representation of Figure 1b the Elmore delay at any node n can be expressed as a function of the tree branch widths and the driver size as

$$T_d^n(W) = \sum_{i \in P(n)} \left(\frac{R_g}{w_g} + \frac{r_i l_i}{w_i} \right) \sum_{j \in D(i)} (c_j l_j w_j + 2 f_j l_j). \tag{2}$$

In (2), $P(n)$ denotes the set of branches from node n to the root of the interconnect tree excluding the gate resistor branch and $D(i)$ the set of branches downstream of branch i and branch i itself. r_i, c_i, and f_i represent the sheet resistance, capacitance per unit area, and the fringe capacitance per unit perimeter of branch i respectively.

Functions of the type

$$f(w_1, w_2, \ldots, w_n) = \sum_{i=1}^{m} \alpha_i \prod_{j=1}^{n} w_j^{\beta_j}, \quad \alpha_i \ge 0, \beta_j \in \Re, \tag{3}$$

are called *posynomial* (*pos*itive poly*nomial*) functions. From an optimization point of view, posynomial functions exhibit the attractive property of convexity under the transformation $w_i = e^{x_i}$ [15]. Several transistor-sizing tools [9, 16] exploit the posynomiality of the transistor-sizing problem for CMOS circuit delay optimization. Since the constants R_g, r_i, c_i, f_i, and l_i in (2) are positive, in [9] it was shown that the Elmore delay at any node in an RC interconnect tree is a posynomial function of the tree branch widths. Also, since usually $g_0, g_1 \ge 0$, the objective function in (1) is posynomial in the branch widths and the driver size.

Therefore, under the transformation $w_i = e^{x_i}$, for the Elmore delay approximation the transformed problem

$$\begin{aligned} &\text{minimize } \tilde{a}(X) = \sum_{i=1}^{N} \gamma_i l_i e^{x_i} + g_1 e^{x_g} + g_0, \\ &\text{subject to } \tilde{T}_d^j(X) \le \hat{t}_d^j, \qquad j = 1...M, \\ &\qquad \ln(\underline{w}_i) \le x_i \le \ln(\overline{w}_i), \quad i = 1...N+1. \end{aligned} \tag{4}$$

where

$$\tilde{T}_d^j(X) = \sum_{i \in P(n)} \left(\frac{R_g}{e^{x_g}} + \frac{r_i l_i}{e^{x_i}} \right) \sum_{j \in D(i)} \left(c_j l_j e^{x_j} + 2 f_j l_j \right) \tag{5}$$

is a convex program. $\tilde{T}_d^j(X)$ is used to denote the Elmore delay at node j as a function of the vector of the transformation variables, $X = \{x_1, x_2, ..., x_N, x_{N+1}\}$.

The monotonicity of the transformation $w = e^x$ also implies any local minimum of the untransformed concurrent driver and wire sizing problem described in (1) under the Elmore delay model is also a global minimum. Based on this, an iterative "one-wire-at-a-time" sensitivity-based heuristic is proposed in [3] in which a single wire is sized at each iteration of the optimization. However, more effective techniques which size several wires at a time can be used to find an optimal solution to the transformed convex form of the concurrent driver and wire sizing problem.

2.2 Posynomiality of the path delay

Consider a typical path in a CMOS circuit composed of a sequence of stages each of which consists of a driver and its associated RC interconnect tree. If a critical sink n of the RC interconnect tree of a stage k connects to the driver of next stage $k + 1$ along the path, then the delay of stage k is given by

$$T_d^n = \sum_{i \in P(n)} \left(\frac{R_g}{w_g} + \frac{r_i l_i}{w_i} \right) \sum_{j \in D(i)} \left(c_j l_j w_j + 2 f_j l_j + C_g^{k+1} \right) \quad (6)$$

where C_g^{k+1} refers to the input capacitance of the driver of stage $k + 1$. If the input capacitance of the driver of each stage is precharacterized as a function of the driver size by

$$C_g(w_g) = c_0 + c_1 w_g + c_2 w_g^2 \quad (7)$$

and the coefficients c_i of the fit are positive – a reasonable assumption for CMOS gates – the Elmore delay of stage k is posynomial in the widths of the branches of the RC fanout tree, and the sizes of the drivers of stage k and stage $k + 1$ respectively. Therefore, in [9] it was shown that for the Elmore delay approximation, the path delay which is the sum of the stage delays is posynomial too. Hence, path delay optimization can be converted into a convex program too.

3 Concurrent gate and wire sizing via SQP

One of the most common approaches to nonlinear optimization is sequential quadratic programming (SQP) [10] which reduces the nonlinear optimization to a sequence of quadratic programming (QP) subproblems. At each iteration, a QP subproblem is constructed from a quadratic approximation of the non-linear objective function and the linearization of the constraints about the solution from the previous iteration. The solution of the QP subproblem which is detemined by any general-purpose QP-solver is then used as the initial solution for the next iteration. The optimization terminates when some convergence criterion is met. Under certain conditions the SQP approach is guaranteed to converge to a solution for any convex programming problem [17].

3.1 Sequential quadratic programming

The strictly convex quadratic program can be generally expressed as

$$\begin{aligned} &\text{minimize} && \frac{1}{2} X^T Q X + X^T C, \\ &\text{subject to} && A_i^T X \le b_i, \qquad i \in I. \end{aligned} \quad (8)$$

Here, the matrix Q is symmetric and positive definite, and I is the index set for the inequality constraints.

Consider the following optimization problem

$$\begin{aligned} &\text{minimize} && F(X), \\ &\text{subject to} && h_i(X) \le 0, \qquad i = 1 \ldots m, \end{aligned} \quad (9)$$

where $F(X)$ and $h_i(X)$ are nonlinear functions of X. Let $g(X)$ be the gradient vector defined by $g(X) = \nabla F(X)$. Variable metric methods for constrained optimization [10] solve (9) by solving a sequence of the following QP subproblems

$$\begin{aligned} &\text{minimize} && \frac{1}{2} (X - \bar{X})^T B(\bar{X}) (X - \bar{X}) + (X - \bar{X})^T g(\bar{X}), \\ &\text{subject to} && (X - \bar{X})^T \nabla h_i(\bar{X}) + h_i(\bar{X}) \le 0, \quad i = 1 \ldots m, \end{aligned} \quad (10)$$

where $\bar{X}$ is the solution of the previous QP iteration. Here $B(\bar{X})$ is calculated from the Hessian of the objective function $F(X)$ as well as the Hessian of the Lagrangian function. Different techniques are used to ensure the positive definiteness of the matrix B at every iteration. In [10] it is mentioned that for convergence the second derivative of the Lagrangian function is far more important than the second derivative of the objective function in the calculation of B. Furthermore, keeping B positive semidefinite at every iteration contributes significantly to convergence.

3.2 SQP applied to the concurrent driver and wire sizing problem

For the concurrent driver and wire sizing problem under the Elmore delay model described by (4), our experimental results indicate that an SQP optimization performed with *only the Hessian of the objective function* taken into consideration provides good overall results. For this problem it can be shown that the second derivative of the objective function is positive semidefinite at every iteration thus aiding in the convergence. When applied to (4) this method compared favorably with the program NLPQL [18] which does take the Hessian of the Lagrangian into account.

In our implementation, we use $H(X)$ and $C(X)$ to denote the Hessian and Jacobian of the circuit area $\tilde{a}(X)$. At each iteration the Elmore delay constraints are linearized about an initial point $\bar{X}$ (the solution of the previous iteration), and the QP subproblem

$$\text{minimize} \quad \frac{1}{2}(X-\bar{X})^T H(\bar{X})\,(X-\bar{X}) + (X-\bar{X})^T C(\bar{X}),$$

$$\text{subject to} \quad (X-\bar{X})^T \nabla \tilde{T}_d^j(\bar{X}) \le \hat{t}_d^j - \tilde{T}_d^j(\bar{X}), \qquad (11)$$

$$j = 1...M,$$

$$\ln(\underline{w}_i) \le x_i \le \ln(\bar{w}_i), \quad i = 1...N+1,$$

solved iteratively until some convergence criterion is met.

We can write the objective function in (4) as

$$a'(X) = \sum_{i=1}^{N+1} k_i e^{x_i} + g_0. \qquad (12)$$

For clarity, in (12), $k_i = \gamma_i l_i$ for each wire i and $k_{N+1} = g_1$ for the driver. The Hessian and the Jacobian of the objective function of (4) are given by

$$H_{ij}(X) = \begin{cases} k_i e^{x_i}, & i = j, \\ 0, & otherwise, \end{cases} \qquad (13)$$

$$C_i(X) = k_i e^{x_i}.$$

Since all diagonal terms of the matrix H are positive and off-diagonal terms zero, H is always positive definite. The sensitivity of the Elmore delay with respect to the transformation variable, x_i, is computed by the chain rule as

$$\frac{\partial \tilde{T}_d^j(X)}{\partial x_j} = \frac{\partial T_d^j(W)}{\partial w_j}\frac{\partial w_j}{\partial x_j} = \frac{\partial T_d^j(W)}{\partial w_j} w_j. \qquad (14)$$

The sensitivity of the Elmore delay at a node j with respect to the widths of all wires j of the RC tree, $\partial T_d^j / \partial w_j$, is computed by the moment sensitivity computation approach described in [19]. The sensitivity of the Elmore delays at the critical sinks with respect to the gate size is given by

$$\frac{\partial T_d^j}{\partial w_g} = \frac{\partial T_d^j}{\partial R_d}\frac{\partial R_d}{\partial w_g} = -\frac{C_{tot} R_d}{w_g}, \qquad (15)$$

where C_{tot} is the total capacitance of the interconnect tree.

Since at each QP iteration of the SQP the quadratic approximation of the objective function as well as the linearization of the constraints are valid only in a certain neighborhood of $\bar{X}$, the upper and lower bounds on the variables x_i are determined based on the validity of this approximation. (Typically, in our experience the wire width sensitivities are valid within $\pm 15\%$ of the current wire width.) For the initial iterations the upper and lower bounds on x_i are set to their actual values to obtain an initial feasible solution.

Using SQP for path delay optimization requires the path delay sensitivities which are easily computed for the fixed-resistor gate delay model since it does not account for transition time effects [20].

3.3 Results

We have implemented the driver routines for the SQP algorithm on a SPARCstation 20 in 5000 lines of C code (this excludes the QP solver). The routine `e04nfc` from the NAG C library [21] is used for the QP solver. We tested our approach on several nets randomly generated on a 10 mm × 10 mm grid. The minimum and maximum allowable width of each branch is set to 1.0 μm and 6.0 μm respectively. The branches of the interconnect tree lie on different layers with parasitics as shown in Table 1.

Layer	c (fF/μm^2)	f (fF/μm)	r (Ω/□)
M1	0.08	0.03	0.14
M2	0.05	0.05	0.07
M3	0.05	0.06	0.08
M4	0.03	0.08	0.02

Table 1. Parasitics for different layers.

After a feasible solution has been found, the allowable change in the wire widths during each iteration is limited to a certain predetermined percentage of the initial widths. This is because the quadratic approximation of the objective function as well as the linearization of the constraints is valid only in a certain region around the previous solution. Experimental results indicate that fastest convergence is obtained if the allowable relative width variation is between 10 and 15%. A lower allowable variation results in slower convergence while a higher value may result in the "bouncing" of successive QP solutions. That is, at a particular iteration if the solution from the QP solver is far away from the initial solution, then this optimal QP solution is not necessarily a better solution for the original problem since neither the quadratic approximation to the objective function nor the linearization of the constraints is valid at this point. The next QP iteration around this new point may lead to a new optimal value near the original initial point without significant progress to the true optimum. The allowable width variation is dynamically varied between successive iterations: It is initially set to a high value and then progressively decreased as the QP solutions converge. We terminate our algorithm when the relative objective function decrease between two successive iterations is less than 10^{-4} and the delays are either less than or within a 10^{-3} relative factor of the target delays. We note that the optimization usually reaches the vicinity of its final solution within 10 iterations. The improvement in the objective function value after these first few iterations is exceedingly small.

The results for single-stage concurrent gate and wire sizing are presented in Table 2. Here the gate size refers to the width of the *n*-channel transistor of the inverter driver. We set the target delays at the critical sinks equal to 50%

of their delays when the branches of the tree are at minimum width and the gate size is 15 μm. For a 0.5 μm CMOS technology, we calculate the resistance of a 1 μm inverter to be 5000 Ω, that is, R_g = 5 kΩμm. The results of the concurrent gate and wire sizing are shown for gate weights of g_1 = 500 and g_1 = 1000. As expected, we see that the relative gate and interconnect area requirements vary with g_1.The runtime grows quadratically in the number of branches. For all examples, optimization using the program NLQPL [18] yielded the same cost as our SQP approach. Both NLQPL and our SQP approach are satisfactory for this optimization.

Net	No. of branches	Initial int. area (10^4 μm^2)	g_1 = 500		g_1 = 1000		CPU time for 25 iter. (s)
			Int. area (10^4 μm^2)	Gate size d_g (μm)	Int. area (10^4 μm^2)	Gate size (μm)	
ex1	14	3.457	3.795	41.54	4.035	38.05	1.42
ex2	30	4.973	5.292	45.99	5.534	39.84	4.28
ex3	48	6.796	7.095	40.29	7.335	36.03	8.49
ex4	62	8.307	8.620	44.25	8.825	39.97	13.23
ex5	100	10.831	11.058	46.60	11.188	41.78	32.64
ex6	298	20.524	20.635	48.22	20.714	43.77	254.2

Table 2. Concurrent gate and wire sizing.

4 Concurrent gate and wire sizing with accurate delays

Accurate delays to the critical sinks of the RC tree in Figure 1b can be efficiently computed using simulators like RICE [7] that are based on asymptotic waveform evaluation (AWE) [22]. Optimizing for the accurate delays, however, also requires exact delay sensitivity computation.

4.1 Delay sensitivity computation in RC trees

Consider a saturated-ramp signal with transition time, t_r, applied at the input of a system with transfer function

$$H(s) = \sum_{i=1}^{n} \frac{k_i}{s - p_i} \tag{16}$$

where the p_i's and k_i's are the poles and residues of the transfer function. The time, t_y, at which the output signal will attain the value y for an input saturated ramp of magnitude 1 with transition time, t_r, is given by the solution of the equation

$$y = \begin{cases} \dfrac{1}{t_r}\left(-t_y + \displaystyle\sum_{i=1}^{q} \frac{k_i}{p_i^2}\left(e^{p_i t_y} - 1\right)\right), & \text{if } t_y \le t_r, \\ 1 + \dfrac{1}{t_r}\displaystyle\sum_{i=1}^{q} \frac{k_i}{p_i^2} e^{p_i t_y}\left(1 - e^{-p_i t_r}\right), & \text{if } t_y > t_r. \end{cases} \tag{17}$$

The sensitivity of the y time-point, t_y, with respect to any variable w, $\partial t_y/\partial w$, can be calculated from (17) if the sensitivities of the poles and residues, $\partial p_i/\partial w$ and $\partial k_i/\partial w$, are known. That is, by differentiating (17) with respect to w and rearranging the terms of the resultant expresssion yields:

$$\frac{\partial t_y}{\partial w} = \begin{cases} \dfrac{\displaystyle\sum_{i=1}^{q} \frac{\partial p_i}{\partial w}\frac{k_i}{p_i^3}\left(p_i t_x e^{p_i t_y} - 2\left(e^{p_i t_y} - 1\right)\right) + \sum_{i=1}^{q} \frac{\partial k_i}{\partial w}\left(\frac{e^{p_i t_y} - 1}{p_i^2}\right)}{1 - \displaystyle\sum_{i=1}^{q} \frac{k_i}{p_i} e^{p_i t_y}}, & \text{if } t_y \le t_r, \\ \dfrac{\displaystyle\sum_{i=1}^{q} \frac{\partial p_i}{\partial w}\frac{k_i}{p_i^3} e^{p_i t_y}\left(p_i t_y\left(1 - e^{-p_i t_r}\right) + p_i t_r e^{-p_i t_r} - 2\left(1 - e^{-p_i t_r}\right)\right) + \displaystyle\sum_{i=1}^{q} \frac{\partial k_i}{\partial w}\frac{e^{p_i t_y}}{p_i^2} 1 - e^{-p_i t_r}}{-\displaystyle\sum_{i=1}^{q} \frac{k_i}{p_i} e^{p_i t_y}\left(1 - e^{-p_i t_r}\right)}, & \text{if } t_y > t_r. \end{cases} \tag{18}$$

The y-delay of the system, t_{dy}, which is defined as the time difference between the y time-point of the output response and the y time-point of the input signal for a fixed input ramp is $t_{dy} = t_y - yt_r$. Therefore, the y-delay sensitivity is $\partial t_{dy}/\partial w = \partial t_y/\partial w$, and the delay sensitivities are easily computed from the pole and residue sensitivities. The output transition time sensitivity is similarly computed.

4.2 Pole and residue sensitivity computation

We know that the response at any node in an RC circuit can be accurately expressed by the first few dominant poles and residues that are efficiently calculated by moment-matching techniques like AWE [22]. If the transfer function at any node of a linear circuit is described in terms of its moments, m_i, by

$$H(s) = m_0 + m_1 s + m_2 s^2 + \ldots + m_{2q-1} s^{2q-1} + \ldots, \tag{19}$$

then an approximation for the first q poles and residues can be obtained by the solution of [22]

$$
\begin{aligned}
-\left(\frac{k_1}{p_1}+\frac{k_2}{p_2}+\ldots+\frac{k_q}{p_q}\right) &= m_0,\\
-\left(\frac{k_1}{p_1^2}+\frac{k_2}{p_2^2}+\ldots+\frac{k_q}{p_q^2}\right) &= m_1,\\
&\cdots\\
-\left(\frac{k_1}{p_1^{2q}}+\frac{k_2}{p_2^{2q}}+\ldots+\frac{k_q}{p_q^{2q}}\right) &= m_{2q-1}.
\end{aligned}
\tag{20}
$$

The pole- and residue-sensitivities can be calculated directly from (20) if the moment sensitivities are known. Partially differentiating (20) with respect to w, for q = 2, for example, results in

$$
\begin{bmatrix}
\frac{1}{p_1} & \frac{1}{p_2} & \frac{-k_1}{p_1^2} & \frac{-k_2}{p_2^2}\\
\frac{1}{p_1^2} & \frac{1}{p_2^2} & \frac{-2k_1}{p_1^3} & \frac{-2k_2}{p_2^3}\\
\frac{1}{p_1^3} & \frac{1}{p_2^3} & \frac{-3k_1}{p_1^4} & \frac{-3k_2}{p_2^4}\\
\frac{1}{p_1^4} & \frac{1}{p_2^4} & \frac{-4k_1}{p_1^5} & \frac{-4k_2}{p_2^5}
\end{bmatrix}
\begin{bmatrix}
\frac{\partial k_1}{\partial w}\\ \frac{\partial k_2}{\partial w}\\ \frac{\partial p_1}{\partial w}\\ \frac{\partial p_2}{\partial w}
\end{bmatrix}
= -
\begin{bmatrix}
\frac{\partial m_0}{\partial w}\\ \frac{\partial m_1}{\partial w}\\ \frac{\partial m_2}{\partial w}\\ \frac{\partial m_3}{\partial w}
\end{bmatrix}.
\tag{21}
$$

In general (21) is a linear system of $2q$ equations with $2q$ unknowns, which is readily solved for small values of q. If necessary, for large values of q, frequency scaling [22] can be used to prevent ill-conditioning of this system due to the powers of poles in the denominator of the matrix coefficients. Moment sensitivities for general RC trees as well as meshes can be computed by the adjoint sensitivity approaches described in [20, 23].

The pole and residue sensitivity computation technique described above can also be used to generate the voltage and current waveform sensitivities [23].

4.3 Near-posynomiality of the accurate delays

Unlike for the Elmore delay model, we cannot prove that posynomiality of the accurately-computed delays. It has been shown that for RC trees the Elmore delay represents an upper bound on the 50% delay [14]. We, therefore, rely on the fact that the Elmore delay is a reliable indicator of the actual delay and hypothesize that the wire sizing problem is posynomial for the accurate 50% delays.

In Table 3 we show the results of applying SQP for optimization of the transformed wire sizing problem to the nets of Table 2 using the accurate 50% delays as well as the Elmore delays. The delays are accurately and efficiently computed using RICE [7]. A fixed driver with R_d = 50Ω is assumed. The target delays at the critical sinks are set to a fraction of the delays when the branches of the tree are at minimum width. An input transition time of 0.5 ns is assumed.

Net	Final interconnect area (10^4 μm^2) for a delay reduction of			
	15%		30%	
	Elmore	RICE	Elmore	RICE
ex1	3.723	3.738	—	—
ex2	5.224	5.240	5.822	5.898
ex3	7.047	7.052	7.717	7.840
ex4	9.018	8.950	9.403	9.609
ex5	11.331	11.321	12.705	12.615
ex6	20.918	20.916	21.849	21.791

Table 3. Comparison of solutions for the Elmore delay model and the accurate delay model.

In [24] it was shown that the cost of the optimal trees routed using the Elmore delay model is very close to the cost of the optimal trees using the accurate delay thereby demonstrating the *fidelity* of the Elmore delay model for routing trees. The results of Table 3 strongly suggest that the fidelity of the Elmore delay model holds for wire sizing too – the cost of the solutions obtained for wire sizing under the Elmore delay model and those obtained using the accurate RICE-computed delays for a certain percentage delay reduction are nearly the same. This also indicates that the wire sizing problem using the actual delays is near posynomial. (It should be noted that the reduction in the actual delay for a certain net may be different from the Elmore delay reduction.)

5 Driver and wire sizing taking the CMOS gate–RC load interaction into account

Modeling the gate by a constant resistor as we have done until now can result in errors up to 30% [11]. Furthermore, this gate model when used in conjunction with the Elmore delay model implicitly assumes that the load presented by the RC fanout tree is the total capacitance of the tree. In [12] it was shown that, for large RC loads, errors up to 100% can be observed by using a total capacitance approximation of the load. The second-order driving point admittance of the load, which is modeled by a π-circuit, was shown to be a better load approximation for on-chip RC interconnect [12]. If the 50% delay, and the 10-50% and 10-90% output transition times of the driver are precharacterized as a function of the load capacitance and the input transition time, the single-resistor voltage-ramp gate delay model presented in [8] accurately estimates the driver delay as well as the driver output waveform. More importantly, it can also be used to determine the waveforms and, therefore, delays at the critical sinks of the RC fanout tree directly using RICE.

5.1 Single-stage driver and wire sizing

The single-resistor voltage-ramp model of [8] for the driver shown in Figure 1 yields a circuit model similar to that used for concurrent driver and wire sizing with the accurate delays at the critical sinks computed by the timing analysis technique of [4]. The computation of the sensitivities of the critical sink delays with respect to the widths of the interconnect branches and the size of the driver is described in [20].

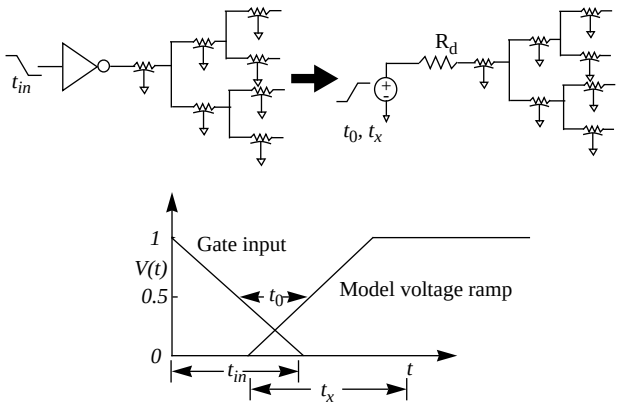


Figure 2: The voltage-ramp delay model of [8].

Figure 3 illustrates the importance of the voltage-ramp delay model for an accurate wire sizing solution by comparing it with the solution obtained using a simple fixed-resistor model for the driver. Again SQP is used for optimization in the transformed domain. It is observed that the fixed-resistor model underestimates the area required for the solution. The simple fixed-resistor model can be used during the earlier stages of design, for example during layout when the lengths of the wire segments or the parasitics are only estimates. For later stages of design when accuracy is an issue the voltage-ramp gate delay model must be used. The results obtained by optimizing the examples of Table 2 for a delay reduction of 50% are shown in Table 4. The CPU times shown in the last column seem to grow cubically in the number of wires. Dramatic improvements in runtime can be achieved by recognizing that for large nets most wires do not influence the optimization [25]. The heuristics of [25] can be used to eliminate the sizes of these wires as optimization variables.

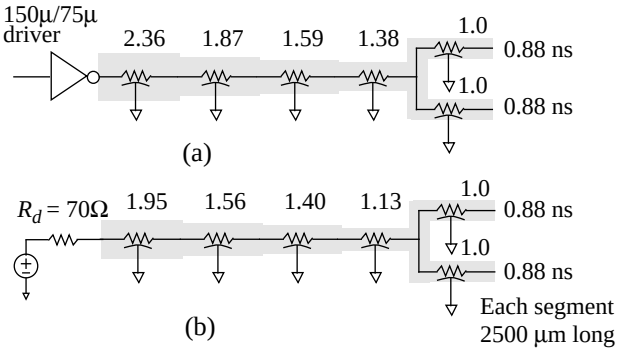


Figure 3: Wire sizing solutions using a) the voltage-ramp delay model b) a fixed-resistor model.

Net	No. of branches	Initial int. area (10^4 µm^2)	g_1 = 500		g_1 = 1000		CPU time for 25 iter. (s)
			Int. area (10^4 µm^2)	Gate size d_g (µm)	Int. area (10^4 µm^2)	Gate size (µm)	
ex1	14	3.457	4.368	41.15	4.452	39.21	8.36
ex2	30	4.973	5.864	38.92	6.438	36.89	22.7
ex3	48	6.796	7.218	36.05	7.632	34.73	55.3
ex4	62	8.307	9.295	41.75	9.392	39.01	89.9
ex5	100	10.831	11.812	36.71	12.024	35.58	269.1
ex6	298	20.524	21.974	42.85	21.862	45.68	5844

Table 4. Concurrent gate and wire sizing results with the voltage-ramp gate delay model.

5.2 Driver and wire sizing for paths

SQP can be applied in the transformed domain for path delay optimization using the voltage-ramp delay model in conjunction with the exact delays. The path delays are computed by the timing analysis technique described in [4]. Due to transition time effects, however, calculating the sensitivities of the path delay with respect to the circuit component sizes is not so trivial as for the Elmore delay model and is described in [20]. The interconnect-dominated path of Figure 1 was optimized by SQP for a path delay of 0.75 ns. The resultant driver and wire sizes are shown.

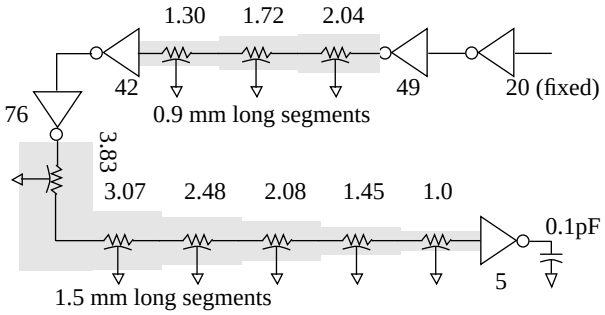


Figure 4: Path delay optimization.

6 SQP applied to RC meshes

For RC interconnect meshes the Elmore delay is not posynomial in the widths of the mesh branches and, therefore, the convexity of the Elmore delay under an exponential transformation does not hold for RC meshes.

However, even without a convexity proof we apply SQP to experiment with delay reduction with RC meshes by adding links in the RC tree and determining which links to eliminate. For example, the delay at sink 1 for the circuit in Figure 5 is 0.36 ns. The wiring segments of this net cannot be sized up for further delay reduction because of constraints imposed by neighboring segments from other nets. A link (possibly routed on another layer) is introduced between the root node and node 1, and SQP optimization carried out with the lower bounds on the widths set to zero. To realize a 0.29 ns delay at node 1, the optimization yields a zero estimate for the width of the 4000 μm segment indicating that it should be eliminated and sizes the 6000 μm segment to a 4 μm width.

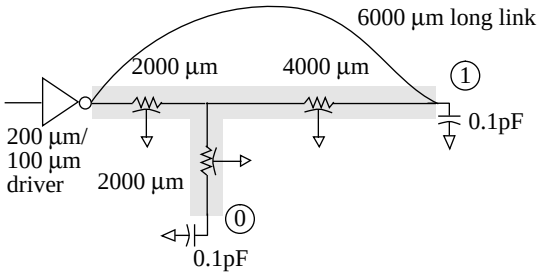


Figure 5: Delay reduction by RC mesh optimization.

7 Conclusions

We have demonstrated the application of a classical optimization approach to delay optimization via gate and wire sizing. The convexity of the optimization under a simple transformation for the Elmore delay model with a simple fixed-resistor model for the gate is used to justify the use of the same transformation for a more complex gate delay model used with the accurate RICE-computed delays. The utility of SQP for delay reduction using wire meshes is also demonstrated.

In addition, a general-purpose delay sensitivity computation technique for linear circuits which can be applied to other problems as well is described.

References

[1] J. Cong, and K.-S. Leung, "Optimal wire sizing under the distributed Elmore delay model," *Proc. of the Intl. Conf. on Computer-Aided Design*, November 1993.

[2] J. Cong and C.-K. Koh, "Simultaneous driver and wire sizing for performance and power optimization," *Proc. of the Intl. Conf. on Computer-Aided Design*, November 1994.

[3] S. S. Sapatnekar, "RC interconnect optimization under the Elmore delay model," *Proc. 31st ACM/IEEE Design Automation Conference*, June 1994.

[4] N. Menezes, S. Pullela, and L. T. Pillage, "Simultaneous gate and interconnect sizing for circuit-level delay optimization," *Proc. 32nd ACM/IEEE Design Automation Conference*, June 1995.

[5] J. Rubinstein, P. Penfield, Jr., and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 202-211, July 1983.

[6] W. C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *J. Applied Physics*, vol. 19, no. 1, pp. 55-63, Jan. 1948.

[7] C.L. Ratzlaff, N. Gopal, and L.T. Pillage, "RICE: Rapid Interconnect Circuit Evaluator", *Proc. 28th ACM/IEEE Design Automation Conference*, June 1991.

[8] F. Dartu, N. Menezes, J. Qian, and L.T. Pillage, "A gate-delay model for high-speed CMOS circuits," *Proc. 31st ACM/IEEE Design Automation Conferenc*e, June 1994.

[9] J. P. Fishburn and A. E. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," *Proc. of the Intl. Conf. on Computer-Aided Design*, November 1985.

[10] M. J. D. Powell, "A fast algorithm for nonlinearly constrained optimization calculations," *Lecture Notes in Mathematics*, No. 630, ed. G. A. Watson, pp. 144-157, Springer Verlag.

[11] J. K. Ousterhout, "A switch-level timing verifier for digital MOS VLSI," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, no. 3, pp. 336-349, July 1985.

[12] J. Qian, S. Pullela, and L. T. Pillage, "Modeling the effective capacitance for the RC interconnect of CMOS gate," *IEEE Trans. Computer-Aided Design*, vol. 12, no. 12, pp. 1526-1535, Dec. 1994.

[13] N. Gopal, "Fast evaluation of VLSI interconnect structures using moment-matching methods," Ph.D. dissertation, The University of Texas at Austin, December 1992.

[14] R. Gupta, B. Tutuianu, B. Krauter, and L. T. Pillage, "The Elmore delay as a bound for RC trees with generalized input signals," *Proc. 32nd ACM/IEEE Design Automation Conference*, June 1995.

[15] J. G. Ecker, "Geometric programming: methods, computations and applications" *SIAM Review*, vol. 22, no. 3, pp. 338-362, July 1980.

[16] S. S. Sapatnekar, et al., "An exact solution to the transistor sizing problem for CMOS circuits using convex optimization," *IEEE Trans. Computer-Aided Design*, vol. 12, no. 11, pp. 1621- 1634, May 1992.

[17] J. Stoer, "Principles of sequential quadratic programming methods for solving nonlinear programs," *Computational Mathematical Programming* (ed. K. Schittkowski), NATO ASI Series, vol. 15, Springer Verlag, Berlin.

[18] K. Schittkowski, "NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems," *Annals of Operations Research*, vol. 5, pp 485-500, 1985/86.

[19] N. Menezes, S. Pullela, F. Dartu, and L. T. Pillage, "RC interconnect synthesis—a moment fitting approach," *Proc. Intl. Conf. on Computer-Aided Design*, November 1994.

[20] N. Menezes, "Methodologies for RC interconnect synthesis," Ph.D. dissertation, The University of Texas at Austin, *in preparation*.

[21] *NAG C library manual – Mark 3*, vol. 2, The Numerical Algorithms Group Ltd., Downers Grove, Illinois.

[22] L.T. Pillage and R.A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, no. 4, pp. 352-366, April 1990.

[23] J. Y. Lee, X. Huang, and R. Rohrer, "Pole and zero sensitivity calculation in asymptotic waveform evaluation," *IEEE Trans. Computer-Aided Design*, vol. 11, no. 5, pp. 586 - 597, May 1992.

[24] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, "Fidelity and near-optimality of Elmore-based routing constructions," *Proc. of the Intl. Conf. Computer Design*, October 1993.

[25] S. Pullela, "Reliable interconnect design for on-chip clock distribution," Ph.D. dissertation, The University of Texas at Austin, May 1995.