Optimal Wire and Transistor Sizing for Circuits with Non-Tree Topology[1]

Lieven Vandenberghe[2]
Electrical Engineering Dept.
University of California
Los Angeles, CA 90095-1594
vandenbe@ee.ucla.edu

Stephen Boyd[2]
Information Systems Laboratory
Electrical Engineering Dept.
Stanford University
Stanford, CA 94305-9510
boyd@isl.stanford.edu

Abbas El Gamal
Information Systems Laboratory
Electrical Engineering Dept.
Stanford University
Stanford, CA 94305-9510
abbas@isl.stanford.edu

[2]Research supported in part by AFOSR (under F49620-95-1-0318) and NSF (under ECS-9222391 and EEC-9420565)

Abstract

Conventional methods for optimal sizing of wires and transistors use linear RC circuit models and the Elmore delay as a measure of signal delay. If the RC circuit has a tree topology the sizing problem reduces to a convex optimization problem which can be solved using geometric programming. The tree topology restriction precludes the use of these methods in several sizing problems of significant importance to high-performance deep submicron design including, for example, circuits with loops of resistors, *e.g.*, clock distribution meshes, and circuits with coupling capacitors, *e.g.*, buses with crosstalk between the lines. The paper proposes a new optimization method which can be used to address these problems. The method uses the dominant time constant as a measure of signal propagation delay in an RC circuit, instead of Elmore delay. Using this measure, sizing of any RC circuit can be cast as a convex optimization problem which can be solved using the recently developed efficient interior-point methods for semidefinite programming. The method is applied to two important sizing problems — sizing of clock meshes, and sizing of buses in the presence of crosstalk.

1 Introduction

The classical approach to optimal sizing of wires and transistors assumes a linear RC circuit model and uses *Elmore delay* as a measure of signal propagation delay. This approach finds its origins in [8, 13, 9]. In particular, Fishburn and Dunlop [9] were first to observe that if the resistors form a tree with the input voltage source at its root and all capacitors are grounded, the Elmore delay of an RC circuit is a *posynomial function* of the conductances and capacitances. This observation has the important consequence that convex programming, specifically geometric programming, can be used to optimize Elmore delay under area and power constraints. Geometric programming forms the basis of the TILOS program and of several extensions and related programs developed since then. We refer to [6] for a comprehensive recent survey.

The tree topology restriction, however, precludes the use of these Elmore delay methods in several sizing problems of significant importance to high performance deep submicron design including circuits with capacitive coupling between the nodes, *e.g.*, buses with crosstalk, and circuits with loops of resistors, *e.g.*, clock meshes.

In this paper we present a new optimal sizing method which can be used to address these problems. The method uses the *dominant time constant* as a measure of signal delay, instead of Elmore delay. The motivation for this choice is that the dominant time constant of a general RC circuit is a quasiconvex function of the conductances and capacitances. In particular we show that it can be optimized using semidefinite programming for which efficient methods have recently been developed. The Elmore delay, on the other hand, has no useful convexity properties except when the RC circuit has a tree topology. Moreover practical experience suggests that the numerical values of the dominant time constant are quite close to threshold delay (and to the Elmore delay).

We apply our method to two important sizing problems. The first is the problem of sizing a clock mesh (§4). The results illustrate that, to a certain extent, our method can be used to design the interconnect *topology* (in addition to sizing). The second problem is the simultaneous sizing of bus line widths and spacings taking into account coupling capacitances between neighboring bus lines (§5). This problem is important in deep submicron design where the coupling capacitance can be significantly higher than the plate capacitance. The results illustrate that optimizing dominant time constant allows us to control not only the signal propagation delay, but also indirectly the crosstalk between the lines. Since the circuit has nongrounded capacitors, this is not possible using Elmore delay.

The outline of the paper is as follows. In §2 we describe the RC circuit model considered in the paper, and define the dominant time constant. In §3 we show that sizing problems using the dominant time constant as a measure of delay lead to semidefinite programming problems, for which efficient methods have recently been developed. In §4 and §5 we describe the application of our method to sizing of clock meshes and buses with crosstalk. Additional theoretical background, and more examples, are presented in [20].

2 The dominant time constant

We consider linear resistor-capacitor (RC) circuits that can be described by the differential equation

$$C\frac{dv}{dt} = -G(v(t) - u(t)), \qquad (1)$$

where $v(t) \in \mathbf{R}^n$ is the vector of node voltages, $u(t) \in \mathbf{R}^n$ is the vector of independent voltage sources, $C \in \mathbf{R}^{n\times n}$ is the capacitance matrix, and $G \in \mathbf{R}^{n\times n}$ is the conductance matrix. We assume that C and G are symmetric and positive definite. We are interested in design problems in which C and G depend on some design parameters $x \in \mathbf{R}^m$. Specifically we assume that the matrices C and G are *affine* functions of x, *i.e.*,

$$\begin{aligned} C(x) &= C_0 + x_1C_1 + \cdots + x_mC_m, \\ G(x) &= G_0 + x_1G_1 + \cdots + x_mG_m, \end{aligned} \qquad (2)$$

where C_i and G_i are symmetric matrices.

Linear RC models of the form (1) are often used as approximate models for transistors and interconnect wires. When the design parameters are the physical widths of conductors or transistors, the conductance and capacitance matrices are affine in these parameters, *i.e.*, they have the form (2).

We are interested in how fast a change in the input u propagates to the different nodes of the circuit, and in how this propagation delay varies as a function of the variables x. To simplify notation, we will consider the autonomous equation $Cdv/dt = -Gv$. We assume the circuit starts at initial condition $v(0)$ and will study the rate at which the voltages

$$v(t) = e^{-C^{-1}Gt}v(0) \qquad (3)$$

become small. We distinguish three possible measures for the circuit propagation delay.

- *The threshold delay* at node k is the first time after which v_k stays below some given threshold level $\alpha > 0$, *i.e.*,

$$T_k^{\text{thres}} = \inf\{\ T \mid |v_k(t)| \leq \alpha \text{ for } t \geq T\ \}.$$

  We call the maximum threshold delay to any node the *critical threshold delay* of the circuit.

- The *Elmore delay* at node k is defined as

$$T_k^{\text{elm}} = \int_0^\infty v_k(t)\ dt.$$

  While T_k^{elm} is always defined, it can be interpreted as a measure of delay only when $v_k(t) \geq 0$ for all $t \geq 0$, *i.e.*, when the node voltage is nonnegative (which is the case in RC circuits with grounded capacitors if $v(0) \geq 0$). We can express the Elmore delay in terms of G, C, and $v(0)$ as

$$T_k^{\text{elm}} = e_k^T G^{-1} C v(0)$$

  where e_k is the kth unit vector. We define the *critical Elmore delay* as the largest Elmore delay at any node, *i.e.*, $T^{\text{elm}} = \max_k T_k^{\text{elm}}$.

- The *dominant time constant* is defined as

$$T^{\text{dom}} = -1/\lambda_1 \qquad (4)$$

  where $\lambda_1, \ldots, \lambda_n$ denote the poles of the circuit, *i.e.*, the eigenvalues of $-C^{-1}G$, sorted in decreasing order, *i.e.*, $0 > \lambda_1 \geq \cdots \geq \lambda_n$. The dominant time constant is a meaningful measure of delay since each voltage can be expressed as a sum of decaying exponentials

$$v_k(t) = \sum_{i=1}^n \alpha_{ik} e^{\lambda_i t},$$

  and the dominant time constant gives the time constant of the slowest of these exponentials.

The threshold delay is the most natural of these three measures, but it is difficult to handle mathematically. It depends on the design parameters x in a very complicated way. Methods for direct optimization of T^{thres} require simulating the circuit to obtain the value of T^{thres} and its derivatives (sensitivity with respect to variations in x). Such methods are necessarily local, *i.e.*, not guaranteed to find a globally optimal design, and limited to small circuits.

The Elmore delay is widely used for optimal circuit sizing (see, *e.g.*, [9, 15, 3, 16, 14, 11]). However, as pointed out in the introduction, it can be efficiently optimized only in circuits with a tree topology.

The dominant time constant T^{dom} is a complicated function of G and C (it is the negative inverse of the largest zero of the polynomial $\det(sC+G)$). However it can be expressed in a form that will be very useful to us:

$$T^{\text{dom}} = \min\{\ T \mid TG - C \geq 0\ \},$$

where the inequality means that $TG - C$ is positive semidefinite. In particular,

$$T^{\text{dom}}(x) \leq T_{\max} \Longleftrightarrow T_{\max} G(x) - C(x) \geq 0. \qquad (5)$$

In other words, an upper bound on the dominant time constant is equivalent to a *linear matrix inequality* (LMI), *i.e.*, a *convex* constraint in x, regardless of the topology of the circuit. This also means that T^{dom} is a *quasiconvex* function of x, *i.e.*, its sublevel sets $\left\{x \;\middle|\; T^{\text{dom}}(x) \leq T_{\max}\right\}$ are convex sets for all $T_{\max}$.

3 Semidefinite programming

In this section we show that sizing problems based on dominant time constant can be cast in terms of two standard optimization problems with LMI constraints.

The most common problem is the *semidefinite programming* (SDP) problem, in which we minimize a linear function subject to an LMI:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & A(x) \geq 0, \end{array} \qquad (6)$$

where $A(x) = A_0 + x_1 A_1 + \cdots + x_m A_m$, $A_i = A_i^T \in \mathbf{R}^{n\times n}$. Semidefinite programs are convex optimization problems, and can be solved very efficiently using recent interior-point methods (see, *e.g.*, [12, 19]).

As an example, suppose the area of the circuit described by (2) is a linear function of the variables x_i. This occurs when the variables represent the widths of transistors or conductors (with lengths fixed as l_i), in which case the circuit area has the form

$$a_0 + x_1 l_1 + \cdots + x_m l_m$$

where a_0 is the area of the fixed part of the circuit. We can minimize the area subject to a bound on the dominant time constant $T^{\text{dom}} \leq T_{\max}$, and subject to upper and lower bounds on the widths by solving the SDP

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m l_i x_i \\ \text{subject to} & T_{\max} G(x) - C(x) \geq 0 \\ & x_{\min} \leq x_i \leq x_{\max}, \quad i = 1, \ldots, m. \end{array} \qquad (7)$$

The optimal solutions of (7) are on the tradeoff curve, *i.e.*, they are *Pareto optimal* for area and dominant time constant. By solving this SDP for a sequence of values of $T_{\max}$, we can compute the exact optimal tradeoff between area and dominant time constant.

A second common optimization problem with LMI constraints has the form

$$\begin{array}{ll} \text{minimize} & \lambda \\ \text{subject to} & \lambda B(x) - A(x) \geq 0 \\ & B(x) > 0, \quad C(x) \geq 0, \end{array} \tag{8}$$

where A, B, and C are symmetric matrices that are affine functions of x, and the variables are x and $\lambda \in \mathbf{R}$. This problem is called the *generalized eigenvalue minimization problem* (GEVP). GEVPs are quasiconvex and can also be solved efficiently using interior-point methods [1, 12].

For example, the problem of minimizing the dominant time constant, subject to an upper bound on the area and upper and lower bounds on the variables, can be cast as a GEVP

$$\begin{array}{ll} \text{minimize} & T \\ \text{subject to} & TG(x) - C(x) \geq 0 \\ & \sum_{i=1}^{m} l_i x_i \leq A_{\max} \\ & x_{\min} \leq x_i \leq x_{\max}, \quad i = 1, \ldots, m. \end{array}$$

with variables T and x.

4 Sizing of clock meshes

The possibility of optimizing RC circuits with loops of resistors is of importance to high-performance microprocessor design where the clock signal is distributed using a mesh instead of a tree. In [7], Desai, Cvijetic, and Jensen describe the design of the clock distribution network on a DEC-alpha processor, and note, "there is a need for algorithms for sizing large non-tree networks." Minimizing the dominant time constant instead of Elmore delay is a promising technique to achieve exactly that goal.

Figure 1 shows the example that we consider in this paragraph. The circuit consists of a mesh of interconnect wire. The number of segments per row or column is N (and the number of nodes in the circuit is equal to $n = (N+1)^2$). We model each interconnect segment (the rectangular elements in Figure 1) as a π-segment, as in Figure 2. The optimization variables are the N^2 segment widths x_i (with constraints $0 \leq x_i \leq w_{\max}$).

Each node of the mesh has a capacitive load C_i. The network is driven by voltage sources with output conductance G_0. The sources switch between zero and

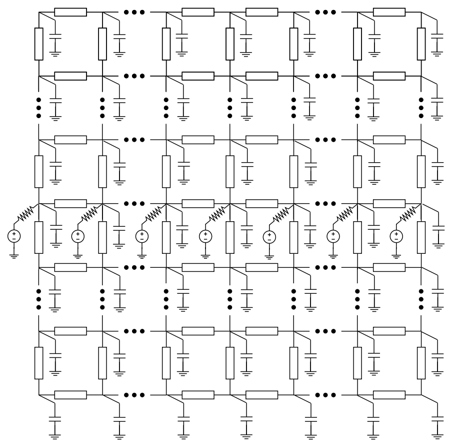

Figure 1: Clock distribution network modeled as an RC mesh. Each rectangular element represents a wire, which we model as a single π-segment as in Figure 2. The drivers switch simultaneously. We are interested in the tradeoff between delay (dominant time constant) and total dissipated power. The variables are the widths of the N^2 segments, where N is the number of segments in each column and row.

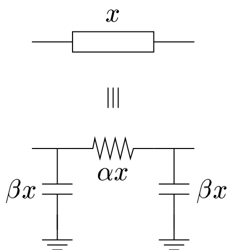

Figure 2: A segment of an interconnect wire with width x is modeled as a conductance αx and two capacitances to the ground βx.

one simultaneously and at a fixed frequency. Therefore the energy dissipated in once cycle is equal to

$$\mathbf{1}^T C(x) \mathbf{1} = \sum_{i=1}^{n} C_i + 2\beta \sum_{i=1}^{N^2} x_i,$$

which is a linear function of the variables x ($\mathbf{1}$ is the n-vector with all components equal to one). This means we can minimize the dissipated power subject to a bound on the dominant time constant by solving the SDP

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T C(x) \mathbf{1} \\ \text{subject to} & T_{\max} G(x) - C(x) \geq 0 \\ & 0 \leq x_i \leq w_{\max}, \quad i = 1, \ldots, N^2. \end{array} \tag{9}$$

By solving the SDP (9) for different values of $T_{\max}$, we can trace the exact tradeoff curve between dissipated power and dominant time constant. This is shown in Figure 4 for the numerical values

$$N = 4, \quad G_0 = 1, \quad \alpha = 1, \quad \beta = 0.5, \quad w_{\max} = 1,$$

and for load capacitances as indicated in Figure 3.

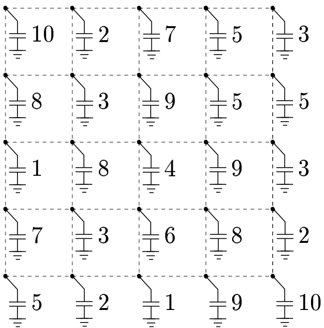

Figure 3: RC mesh example with 4×4 segments, and the numerical values of the load capacitances used in the calculation.

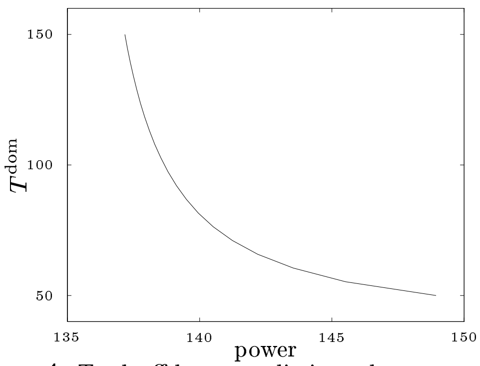

Figure 4: Tradeoff between dissipated power and dominant time constant.

Figure 5 shows the solution for two different points on the tradeoff curve ($T^{\text{dom}} = 55$ and $T^{\text{dom}} = 100$).

We note that the topology is different in the two cases. More segments are used in the circuit on the left, which has a small dominant time constant and large power consumption (large total capacitance). In the solution on the right, fewer segments are used and they are smaller, which reduces the power dissipation but increases the dominant time constant.

Figure 6 shows the fastest and the slowest step responses in both circuits, when a step input is applied simultaneously to the five voltage sources in the middle row. Note in particular that the values of the three delay measures are very close (and in fact, the dominant time constant approximates the 50%-threshold delay better than the Elmore delay). This observation is confirmed by many other examples (see [20]).

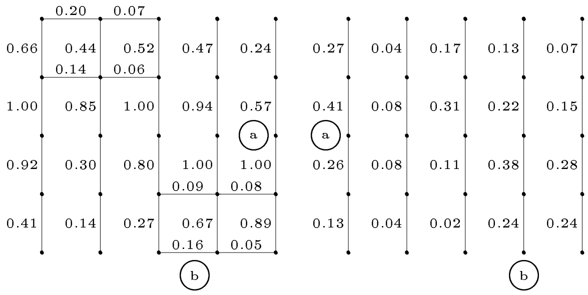

Figure 5: Optimal solution for two points on the tradeoff curve: $T^{\text{dom}} = 50$ (left) and $T^{\text{dom}} = 100$ (right). The numbers indicate the optimal widths x_i of the segments; the segments with width $x_i = 0$ are not shown.

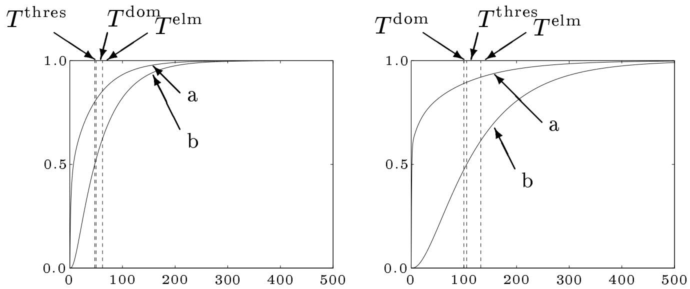

Figure 6: Step responses for the two solutions in Figure 5. The plots show the responses at the fastest (a) and the slowest (b) node in Figure 5. We also show the critical 50%-threshold delay, the critical Elmore delay, and the dominant time constant.

5 Sizing of buses with crosstalk

The second application demonstrates another important advantage of using dominant time constant instead of Elmore delay: the ability to take into account non-grounded capacitors.

The problem is to determine the optimal line widths *and* spacings for a bus taking into account the coupling capacitances between the lines. We consider an example with three wires, each consisting of five segments, as shown in Figure 7. The optimization variables are the widths w_{ij}, and the distances s_1 and s_2 between the wires.

The RC model of the three wires is shown in Figure 8. The wires are connected to a voltage source with output conductance G at one end, and to capacitive loads at the other end. As in the previous example, each segment is modeled as a π-segment, with conductance and capacitance proportional to the segment width w_{ij}. We include a parasitic capacitance between the wires. We assume that there is a capacitance between the jth segments of wires 1 and 2, and between the jth segments of wires 2 and 3, with total

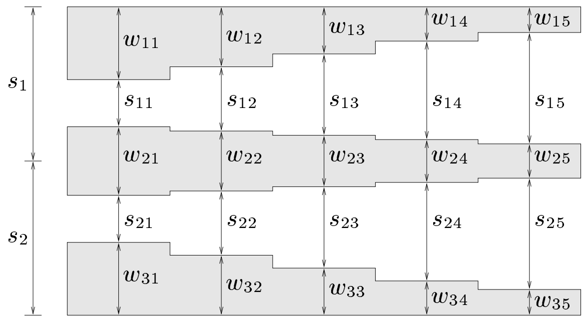

Figure 7: Wire sizing and spacing. Three parallel wires consisting of five segments each. The conductance and capacitance of the jth segment of wire i is proportional to w_{ij}. There is a capacitive coupling between the ith segments of wires 1 and 2, and between the ith segments of wires 2 and 3, and the value of this parasitic capacitance is inversely proportional to s_{1i}, and s_{2i}, respectively. The optimization variables are the 15 segment widths w_{ij} and the distances s_1 and s_2.

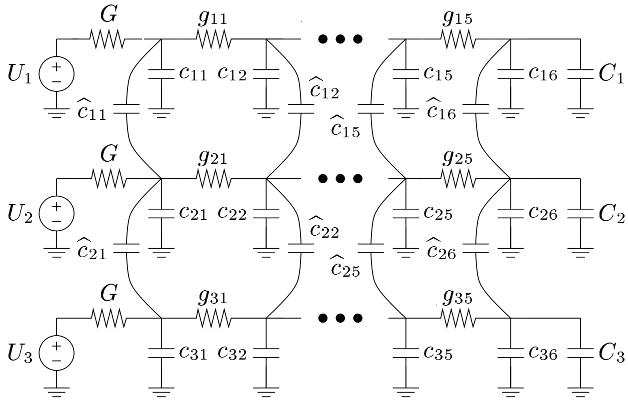

Figure 8: RC model of the three wires shown in Figure 7. The wires are connected to voltage sources with output conductance G at one end, and to load capacitors C_i at the other end. The conductances g_{ij} and capacitances c_{ij} are part of the π-models of the wire segments. The capacitances $\widehat{c}_{ij}$ model the capacitive coupling. The conductances and capacitances depend on the geometry of Figure 7 in the following way: $g_{ij} = \alpha w_{ij}$, $c_{i1} = \beta w_{i1}$, $c_{ij} = \beta(w_{ij} + w_{i(j-1)})$ $(1 < j < 6)$, $c_{i6} = \beta w_{i5}$, $\widehat{c}_{i1} = \gamma / s_{i1}$, $\widehat{c}_{ij} = \gamma/s_{ij} + \gamma/s_{i(j-1)}$ $(1 < j < 6)$, $\widehat{c}_{i6} = \gamma/s_{i5}$.

values inversely proportional to the distances s_{1j} and s_{2j}, respectively. To obtain a lumped model, we split this distributed capacitance over two capacitors: the capacitance between segments j of wires 1 and 2 is lumped in two capacitors with value γ/s_{1j}; the total capacitance between segments j of wires 2 and 3 is lumped in two capacitors with value γ/s_{2j}. This leads to the RC circuit in Figure 8.

We also impose the constraints that the distances s_{ij} between the wires must exceed a value $s_{\min}$, and that wire widths are less than $w_{\max}$. We can minimize the total width $s_1 + s_2$ of the three wires subject to a bound on the dominant time constant of the circuit, by solving the optimization problem

$$\begin{array}{ll} \text{minimize} & s_1 + s_2 \\ \text{subject to} & T_{\max} G(w) - C(w,s) \geq 0 \\ & s_{1j} = s_1 - w_{1j} - 0.5 w_{2j}, \;\; j = 1, \ldots, 5 \\ & s_{2j} = s_2 - w_{3j} - 0.5 w_{2j}, \;\; j = 1, \ldots, 5 \\ & s_{ij} \geq s_{\min}, \; i = 1, 2, \;\; j = 1, \ldots, 5 \\ & w_{ij} \leq w_{\max}, \; i = 1,2,3, \;\; j = 1, \ldots, 5, \end{array} \tag{10}$$

in the variables s_1, s_2, w_{ij}, s_{ij}. Note that the capacitance matrix contains terms that are inversely proportional to the variables s_{ij}, and therefore problem (10) is *not* an SDP. However, it is shown in [20] that by a change of variables $t_{ij} = 1/s_{ij}$, problem (10) can be reformulated as a convex optimization problem

$$\begin{array}{ll} \text{mimimize} & s_1 + s_2 \\ \text{subject to} & T_{\max} G(w) - C(w,t) \geq 0 \\ & 1/t_{1j} \leq s_1 - w_{1j} - 0.5 w_{2j}, \;\; j = 1, \ldots, 5 \\ & 1/t_{2j} \leq s_2 - w_{3j} - 0.5 w_{2j}, \;\; j = 1, \ldots, 5 \\ & 0 \leq t_{ij} \leq 1/s_{\min}, \; i = 1, 2, \;\; j = 1, \ldots, 5 \\ & w_{ij} \leq w_{\max}, \; i = 1,2,3, \; j = 1, \ldots, 5, \end{array} \tag{11}$$

with variables s_1, s_2, t_{ij}, and w_{ij}. Problem (11) can be readily cast as an SDP (see [20]).

Figures 9 through 12 illustrate the solution of (10) for two values of $T_{\max}$, assuming the parameter values

$$\begin{array}{lllll} G = 100, & C_1 = 10, & C_2 = 20, & C_3 = 30, & \\ \alpha = 1, & \beta = 0.5, & \gamma = 2, & s_{\min} = 1, & w_{\max} = 2. \end{array}$$

Figures 9 and 10 illustrate a solution for $T_{\max} = 130$. The widest wire is number three, since it drives the largest load, the narrowest wire is number one, which drives the smallest load. We also see that the smallest distance between the wires is equal to its minimum allowed value of 1.0, which means that the cross-coupling did not affect the optimal spacing between the wires. Figure 10 shows the output voltages for steps applied to one of the wires, while the two other input voltages remains zero.

Figures 11 and 12 illustrate a solution for $T_{\max} = 90$. Note that here the distance between the second

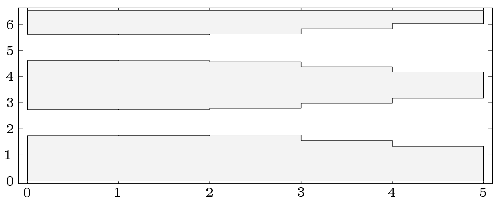

Figure 9: Solution of (10) for $T_{\max} = 130$. Note that the distance between the wires is equal to its minimal allowed value of 1.0

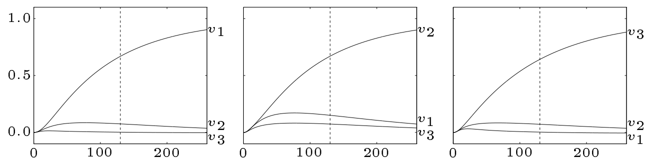

Figure 10: Responses for solution of Figure 9. The voltages at the output nodes due to a step applied to the first wire (left figure), second wire (center), or third wire (right). The dashed line marks the dominant time constant.

and third wires is larger than the minimum allowed value of 1.0. The other figures show the output voltages for the same situations as above.

Note that we cannot guarantee that the peak due to crosstalk stays under a certain level. This would be a specification in practice, but it is difficult to incorporate into the optimization problem. However we influence the level indirectly: minimizing the dominant time constant makes the crosstalk peak shorter in time (since the dominant time constant determines how fast all voltages settle around their steady-state value). Indirectly, this also tends to make the magnitude of the peak smaller (as can be seen by comparing the crosstalk levels for the two solutions in the examples).

A practical heuristic based on the dominant time constant minimization that would guarantee a given peak level is as follows. We first solve problem (10) for a given value of $T_{\max}$. Then we simulate to see if crosstalk level is acceptable. If not, we increase the spacing of the wires until it is. Then we determine the optimal wire sizes again, keeping the wires at least at this minimum distance. This iteration is continued until it converges. The dominant time constant of the final result will be at least as good as the first solution and the crosstalk level will not exceed the maximum level.

6 Conclusions

We presented a new method for wire and transistor sizing. The method uses the dominant time con-

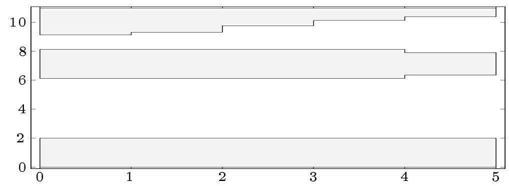

Figure 11: Solution of (10) for $T_{\max} = 90$.

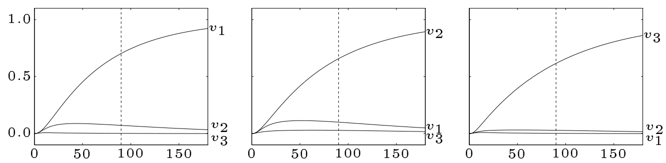

Figure 12: Responses for the solution shown in Figure 11. The voltages at the output nodes, due to to a step applied to the first wire (left figure), second wire (center), or third wire (right). The dashed line marks the dominant time constant.

stant as a measure of signal delay in RC circuits. The main advantage of using this measure is that RC circuits with general non-tree topologies can be optimally sized using convex optimization. This is in contrast to the Elmore delay sizing methods which only work for RC trees. We demonstrated the power of this method by applying it to two important examples of significant practical importance: sizing of clock meshes, and sizing of buses in the presence of crosstalk.

The method we described uses the recently developed interior-point methods for semidefinite programming (see, *e.g.*, [12, 19]). Since real world sizing problems are likely to be very large we briefly discuss the complexity of the SDP methods. Two factors determine the overall complexity of these methods: the total number of iterations and the complexity of an iteration. It can be shown that the number of iterations to solve an SDP to a given accuracy ϵ grows at most as $O(\sqrt{n}\log(1/\epsilon))$, where n is the size of the matrix $A(x)$ in (6) [12]. In practice the performance is even better than suggested by this worst-case bound. The number of iterations usually lies between 5 and 50, almost independently of problem size. For practical purposes it is therefore fair to consider the total number of iterations as constant, and to regard the complexity of an iteration as dominating the overall complexity.

Each iteration involves solving a large system of linear equations to compute search directions. Little can be said about the complexity of this computation since it largely depends on the degree to which the structure of the problem can be exploited. If the problem has no structure, *i.e.*, if the matrices A_i in (6) are completely dense, then the complexity of one iteration is $O(mn^3 + m^2n^2)$. This is the case for the general-purpose SDP solvers SP and SDPSOL [17, 21], that were used in the examples discussed in this paper. These solvers can handle problems with up to several hundred variables without difficulty, but become impractical for larger problems.

In most applications, however, there is a great deal of structure that can be exploited, and specialized solvers can be orders of magnitude more efficient than general-purpose solvers (see for a few examples, [18, 2]). In particular the SDP problems arising in dominant time constant minimization exhibit two forms of sparsity that can be exploited in a specialized solver. First, the capacitance and conductance matrices C and G are usually sparse matrices (indeed C is often diagonal). Second, each variable x_i affects only a very small number of elements of C and G (*i.e.*, the different matrices C_i and G_i in (2) are extremely sparse).

We conclude by discussing the differences between Elmore delay and the dominant time constant. The most important difference is that the dominant time constant *always* leads to tractable convex or quasiconvex optimization problems, with no restrictions on circuit topology. This follows from (5) which holds regardless of the circuit topology. Specifically, we note the following advantages.

- Elmore delay optimization applies only to circuits with one input source. The dominant time constant can be applied to circuits with multiple sources, a problem that has only recently received attention [4, 5].

- The Elmore delay in an RC tree is a posynomial function if the conductances depend on one variable only. For dominant time constant optimization the conductance and capacitances can be general affine functions of the variables.

- The circuits may contain loops of resistors, *e.g.*, clock meshes. Although for grounded capacitor RC circuits with loops of resistors, the Elmore delay is still a meaningful approximation of signal delay [10, 22, 23], it does not have the simple posynomial form as it does for RC trees, and convex optimization cannot be used to minimize it.

- The possibility of handling non-tree topologies allows us to design the topology of the interconnection itself. For example in the optimization of a clock mesh we start with a full grid of possible wire segments. After optimal wire sizes are computed, some (and often, many) of the wires have zero widths, which means they are not needed in the circuit (see also [4] for problems of designing interconnection topology).

- Dominant time constant minimization handles circuits with capacitive coupling between the nodes (see §5).

The Elmore time constant, in addition to being quite useful as a measure of delay when sizing RC trees, is sometimes more appropriate to use than the dominant time constant. We give here two examples where this is the case.

- Consider a path consisting of several stages of buffered wire segments. The total Elmore delay of the path is the sum of the Elmore delays of the segments, and is still a posynomial function that can be efficiently minimized by geometric programming. In contrast it is not possible to efficiently minimize the sum of the dominant time constants, since in general the sum of quasiconvex functions is not quasiconvex.

- The dominant time constant is useful as an alternative to the critical Elmore delay, *i.e.*, the Elmore delay to the node with the slowest response. It is not a good measure for the delay to the other nodes.

References

[1] S. Boyd and L. El Ghaoui. Method of centers for minimizing generalized eigenvalues. *Linear Algebra and Applications, special issue on Numerical Linear Algebra Methods in Control, Signals and Systems*, 188:63–111, July 1993.

[2] S. Boyd, L. Vandenberghe, and M. Grant. Efficient convex optimization for engineering design. In *Proceedings IFAC Symposium on Robust Control Design*, pages 14–23, September 1994.

[3] W. Chuang, S. S. Sapatnekar, and I. N. Hajj. Timing and area optimization for standard-cell VLSI circuit design. *IEEE Transactions on Computer-Aided Design*, 14(3):308–320, 1995.

[4] J. Cong and L. He. Optimal wiresizing for interconnects with multiple sources. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, pages 568–574, 1995.

[5] J. Cong and L. He. Optimal wiresizing for interconnects with multiple sources. *ACM Transaction on Design Automation of Electronic Systems*, 1, 1996.

[6] J. Cong, L. He, C.-K. Koh, and P. H. Madden. Performance optimization of VLSI interconnect layout. *Integration, the VLSI journal*, 21:1–94, 1996.

[7] M. P. Desai, R. Cvijetic, and J. Jensen. Sizing of clock distribution networks for high performance CPU chips. In *Proceedings of the 33rd ACM/IEEE Design Automation Conference*, pages 389–394, 1996.

[8] W. C. Elmore. The transient response of damped linear systems with particular regard to wideband amplifiers. *Journal of Applied Physics*, 19:55–63, 1948.

[9] J. P. Fishburn and A. E. Dunlop. TILOS: a posynomial programming approach to transistor sizing. In *Proceedings ICCAD'85*, pages 326–328, 1985.

[10] T.-M. Lin and C. A. Mead. Signal delay in general *RC* networks. *IEEE Transactions on Computer-Aided Design*, 3(4):321–349, 1984.

[11] N. Menezes, R. Baldick, and L. T. Pileggi. A sequential quadratic programming approach to concurrent gate and wiring sizing. In *Proceedings of the IEEE International Conference on Computer-Aided Design*, pages 144–151, 1995.

[12] Yu. Nesterov and A. Nemirovsky. *Interior-point polynomial methods in convex programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[13] J. Rubinstein, P. Penfield, and M. A. Horowitz. Signal delay in *RC* tree networks. *IEEE Transactions on Computer-Aided Design*, 2(2):202–211, 1983.

[14] S. Sapatnekar, V. B. Rao, P. 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*, 12:1621–1634, 1993.

[15] S. S. Sapatnekar. Wire sizing as a convex optimization problem: exploring the area-delay tradeoff. *IEEE Transactions on Computer-Aided Design*, 15(8):1001–1011, 1996.

[16] J.-M. Shyu, A. Sangiovanni-Vincentelli, J. P. Fishburn, and A. E. Dunlop. Optimization-based transistor sizing. *IEEE Journal of Solid-State Circuits*, 23(2):400–409, 1988.

[17] L. Vandenberghe and S. Boyd. SP*: Software for Semidefinite Programming. User's Guide, Beta Version.* Stanford University, October 1994. Available at `http://www-isl.stanford.edu/people/boyd`.

[18] L. Vandenberghe and S. Boyd. A primal-dual potential reduction method for problems involving matrix inequalities. *Mathematical Programming*, 69(1):205–236, July 1995.

[19] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, March 1996.

[20] L. Vandenberghe, S. Boyd, and A. El Gamal. Optimizing dominant time constant in RC circuits. Technical report, Information Systems Laboratory, Stanford University, November 1996.

[21] S.-P. Wu and S. Boyd. SDPSOL: *A Parser/Solver for Semidefinite Programming and Determinant Maximization Problems with Matrix Structure. User's Guide, Version Beta.* Stanford University, June 1996.

[22] J. L. Wyatt. Signal delay in *RC* mesh networks. *IEEE Transactions on Circuits and Systems*, 32:507–510, 1985.

[23] J. L. Wyatt. Signal propagation delay in *RC* models for interconnect. In A. E. Ruehli, editor, *Circuit Analysis, Simulation and Design*, volume 3, pages 254–291. Elsevier, 1987.