Convexity-Based Optimization for Power-Delay Tradeoff using Transistor Sizing

Mahesh Ketkar, and Sachin S. Sapatnekar
Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN 55455

Priyadarsan Patra
Strategic CAD Labs,
Intel Corporation, Hillsboro, OR 97124

Abstract

This paper solves the delay-power optimization problem by employing accurate optimization techniques. A new class of functions, called generalized posynomials, which is a superset of the set of posynomials, is used to model the delay and the power. The mapping of these generalized posynomials to regular posynomials, which allows the use of existing posynomial solvers, is shown. We show how all constraints representing the circuit can be represented compactly in posynomial form. Finally, the results of power optimization are presented.

1 Introduction

It is well known that in the past, trends in performance, density and power have followed the scaling theory. For these trends to continue, two important issues that need special attention are power delivery and power dissipation [1]. Typically, each generation of chips has shown about a 30% reduction in gate delay and about same or more reduction in energy dissipation per transition. The total power dissipation can be scaled down by reducing either the supply voltage, frequency or die size, each of which results in reduced performance. Therefore, the problem of optimization for low power to meet the stringent performance constraints continues to be a very relevant problem.

Several approaches that perform circuit level optimization for area or power have been published. In [2], the authors published a linear programming based method for gate sizing for power-delay tradeoffs, using a piecewise linear gate delay model. Such a model is simplistic and inaccurate in the present deep submicron regime. In [3], the power optimization problem is solved by transistor sizing and ordering. Power dissipation is modeled accurately by incorporating fanout capacitances and gate transition measure. A "pin-delay" model is developed based on the delay model used in SIS. However, in the absence of any convexity properties, they use heuristic techniques to solve the problem. The evident drawback of employing a heuristic approach is that there is no guarantee that the solution will be optimal, even after application of any refinement techniques. In [4], only transistor reordering is used for the power-performance optimization, but the important aspect of transistor sizing is not considered. Recently an accurate technique for circuit optimization has been presented in [5], using a nonlinear optimization technique. Simulation followed by time domain sensitivity computation is used to provide gradients to a nonlinear optimizer. However, drawbacks include the destruction of convexity properties due to transient simulation-based modeling, the need for circuit-level simulation which tends to make the optimization computationally intensive, and the pattern dependent nature of the timing analysis which requires the user to supply the simulation vectors.

Thus, although several attempts have been made to solve the problem, the lack of convexity based techniques tends to keep the results away from optimality. The solution is to use the models that posses convexity properties and hence lend themselves to accurate optimization techniques. It is observed in [6] that the Elmore model posses convexity properties, and several transistor sizing techniques have made use of the this model [6,7]. However, the Elmore model is inaccurate for deep submicron technologies. Its failure to correctly consider several factors such as transistor nonlinearities, the effect of the position of switching transistor, the input transition times, and the transistors opposing the transition, are a few of the reasons for its inaccuracy.

In [8], we had developed a convex model for gate delay that overcomes these limitations. For the sake of completeness, we have included a brief treatment of the model in section 2. The theoretical underpinning of this model development is a result that defines a new class of functions that is shown to work well for modeling circuit delays. This set of functions includes the set of posynomials as a proper subset, and therefore, we refer to these functions as *generalized posynomials*.

In this paper, we show the precise relation between a generalized posynomial programming problem and a posynomial programming problem. We illustrate this in Section 4, and show that geometric programming solvers may be used for generalized posynomial programs. This is a powerful result, since we now have a large new class of functions that can accurately capture the delay behavior of a circuit, but may use conventional fast methods, which exploit the structure of the problem, for solving them.

Before proceeding to present the convex models, a few words on the importance of convex optimization are in order. A convex optimization problem involves the minimization of a convex function over a convex set (for definitions of these terms, the reader is referred to [9]). A problem of the type

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{such that} & g_i(\mathbf{x}) \leq 0, 1 \leq i \leq m \\ & \mathbf{x} \in \mathbf{R}^n \end{array} \qquad (1)$$

where $\mathbf{R}^n$ is the n-dimensional real space and $\mathbf{x}$ a vector in $\mathbf{R}^n$, is a convex programming problem if $f(\mathbf{x})$ and $g_i(\mathbf{x}), 1 \leq i \leq m$, are convex functions. The advantage of a convex programming formulation is that the problem is known to have the property that any local minimum is also a global minimum, and efficient optimization algorithms for solving such a problem are available. Therefore, it is desirable to attempt to express any optimization problem using a convex formulation, as far as possible, under the caveat that the accuracy of the modeling functions for the objective and the constraints must be preserved.

In the context of transistor sizing, this requires the derivation of convex closed-form expressions for the path delay; as a result, this will satisfy the requirement of relation (1) that each timing constraint is of the form $g_i(\mathbf{x}) \leq 0$. In [8], where we solve the problem of area optimization, we have incorporated these models into a TILOS [6] like optimizer. However, the use of heuristics has inherent drawbacks as there can be no guarantee of optimality. For the model to fully exploit its convexity properties, it is necessary to couple it with accurate mathematical optimizers. In this paper, we show, for the first time, how these models lend themselves effectively to traditional posynomial solvers. We show the mapping from generalized posynomials to posynomials which enables these models to be used with a geometric programming optimizer.

The transistor size optimization problem is formally stated as

$$\begin{array}{ll} \text{minimize} & Power \\ \text{subject to} & Delay \leq T_{spec}. \end{array} \quad (2)$$

An important contribution of this paper is the accurate solution of this power-delay tradeoff problem. The accuracy stems from the fact that we use an accurate generalized posynomial modeling framework coupled with formal optimization techniques. A mapping between generalized posynomials and traditional posynomials is shown, so that a fast posynomial solver can be used as the optimization engine. This optimizer, unlike the convex optimizer in [7], requires all constraints to be explicitly listed. We show how this may be done efficiently while preserving the posynomial nature of all constraints. Finally, we differ from [8], which uses a TILOS-like heuristic optimizer for area-delay optimization, and we can guarantee a global minimum using this approach.

2 Convex delay model

2.1 Delay abstractions and posynomial delay modeling

The delay characteristics of the output waveform at a gate may be represented by two numbers:

(1) the *delay*, i.e., the difference in the time when the output waveform crosses 50% of its final value, and the corresponding time for the input waveform.

(2) the *output transition time*, i.e., the time required for the waveform to go from 10% to 90% of its final value.

In much of the previous work on transistor sizing, the circuit delay has been expressed in the form of a class of functions known as posynomials. A posynomial is a function p of a positive variable $\mathbf{x} \in \mathbf{R}^n$ that has the form

$$p(\mathbf{x}) = \sum_j \gamma_j \prod_{i=1}^{n} x_i^{\alpha_{ij}} \quad (3)$$

where the exponents $\alpha_{ij} \in \mathbf{R}$ and the coefficients $\gamma_j \in \mathbf{R}^+$. In the positive orthant in the $\mathbf{x}$ space, posynomial functions have the useful property that they can be mapped onto a convex function through an elementary variable transformation, $(x_i) = (e^{z_i})$.

The Elmore delay model used, for example, in TILOS [6] and iCONTRAST [7], employed the following form of expressions for the path delay.

$$D(\mathbf{x}) = \sum_{i,j=1}^{n} a_{ij} \frac{x_i}{x_j} + \sum_{i=1}^{n} \frac{b_i}{x_i} + K \quad (4)$$

where $a_{ij}, b_i, K \in \mathbf{R}^+$ are constants and, $\mathbf{x} = [x_1, \cdots, x_n]$ is the vector of transistor sizes. Equation (4) represents a posynomial function, and hence convex optimizers can efficiently take advantage of the convexity properties of such a delay expression to obtain an optimal solution. Notice that the Elmore delay expressions are a small subset of the set of posynomials; specifically they are posynomials whose exponents belong to the set {-1,0,1}.

2.2 Generalized posynomials

Posynomials and convex functions are a rich class of functions better delay estimates can be obtained by fully exploiting this richness. Here, we describe a generalized posynomial form that is useful for delay modeling and can be proven to have convexity properties.

A generalized posynomial function $G_k(\mathbf{x}), \mathbf{x} \in \mathbf{R}^n$, where $k \geq 0$ is called the order of the function, is defined recursively as follows:

1. A generalized posynomial of order 0, G_0, is the posynomial form defined earlier:

$$G_0(\mathbf{x}) = \sum_j \gamma_j \prod_{i=1}^{n} x_i^{\alpha_{ij}}, \quad (5)$$

where the exponents $\alpha_{ij} \in \mathbf{R}$ and the coefficients $\gamma_j \in \mathbf{R}^+$.

2. A generalized posynomial of order k is defined as

$$G_k(\mathbf{x}) = \sum_j \gamma_j \prod_{i=1}^{n} \left[G_{k-1,i}(\mathbf{x})\right]^{\alpha_{ij}}, \quad (6)$$

where the exponents $\alpha_{ij} \in \mathbf{R}^+$ and the coefficients $\gamma_j \in \mathbf{R}^+$, and $G_{k-1,i}(\mathbf{x})$ is a generalized posynomial of order $k-1$.

Specifically, the generalized posynomial of first order, is given by

$$f(\mathbf{x}) = \sum_i \gamma_i \prod_{j=1}^{m} \left(\sum_{l=1}^{p_i} \omega_{ijl} \prod_{s=1}^{n} x_s^{a_{ijls}} \right)^{\beta_{ij}} \quad (7)$$

where each $\beta_{ij} \in \mathbf{R}$, each $a_{ijls} \in \mathbf{R}$, each $\gamma_i \in \mathbf{R}^+$, and each $\omega_{ijl} \in \mathbf{R}^+$.

The following theorem parallels the relationship between posynomials and convex functions, and is proved in [8]. As a consequence of this, we will loosely refer to generalized posynomials as being convex in future discussions.

Theorem 1: If the range of interest of $\mathbf{x}$ is restricted to the positive orthant where each $x_i > 0$, then under the variable transformation from the space $\mathbf{x} \in \mathbf{R}^n$ to the space $\mathbf{z} \in \mathbf{R}^n$ given by $x_i = e^{z_i}$, the generalized posynomial function f of equation (6) is mapped to a convex function in the $\mathbf{z}$ domain, provided $\beta_{ij} \geq 0$ for each i, j.

Stripping Equation (7) of its complicated notation, one may observe that the term in the innermost bracket, $\sum_{l=1}^{m} \omega_{ijl} \prod_{s=1}^{p} x_s^{a_{ijls}}$, represents a posynomial function. Therefore, a generalized posynomial of first order is similar to a posynomial, except that the place of the x_i variable in Equation (3) is taken by a posynomial. Similarly, a generalized posynomial of order k uses a generalized posynomial of order $k-1$ in place of the x_i variables in Equation (3). The class of posynomials is, by definition a proper subset of the class of generalized posynomials.

2.3 Generalized posynomial delay model

The parameters that have an observable effect on gate delay are the widths of transistors in the gate, input transition time, and loading capacitance. We refer to these input parameters as characterization variables. Our aim is to develop a generalized posynomial delay model which is dependent on all these characterization variables.

We attempted the use of several types of functions to achieve the desired levels of accuracy. The general form of expression that provided consistently good results for different gate types is as follows

$$\text{Delay} = \sum_{j=1}^{m} P_j \cdot \prod_{i=1}^{n} (x_i^{\Delta} + c_{ij})^{\beta_{ij}} + C \qquad (8)$$

Here, the x_i's are characterization variables, and the c_{ij}'s, β_{ij}'s, C, and P_j's are real constants, referred to collectively as *characterization constants*. The parameter Δ is set to either -1 or 1, depending on the variable, as will soon be explained. The problem of characterization is that of determining appropriate values for the characterization constants.

Due to the curve-fitting nature of the characterization procedure (akin to standard cell characterization), it is not possible to ascribe direct physical meanings to each of these terms. However, it can be seen that the gate delay increases as the loading capacitance, the width of the transistor opposing transition, and the input transition time are increased, and decreases as the switching transistor width is increased, implying that an appropriate choice for the parameter Δ for the first three variables is 1, while for the last one it is -1. Note that this is not as restrictive as the Elmore form since, among other things, the β_{ij}'s and c_{ij}'s provide an additional degree of freedom that was not available for the Elmore delay form.

A two-step methodology is adopted to complete the characterization. In the first step, a number of circuit simulations are performed to generate points on a grid. In the second, a least-squares procedure is used to fit the data to a function of the type in Equation (8). A series of simulations is performed to collect the experimental data using the HSPICE circuit simulator. Since the number of points increases exponentially with an increase in number of variables, judicious pruning is applied to the characterization space.

Gate	Delay		
	Output Transition	Mean	Deviation
Inv	Rise	0.31%	2.83 %
	Fall	1.29 %	2.82 %
Nor2	Rise	1.82 %	2.56 %
	Fall	11.10 %	5.06 %
Nand2	Rise	5.18 %	6.17 %
	Fall	-0.46 %	3.58 %
Nor3	Rise	0.24 %	1.76 %
	Fall	24.2 %	7.64 %
Nand3	Rise	9.21 %	5.98 %
	Fall	1.26 %	2.29 %
AOI3	Rise	5.21 %	6.38 %
	Fall	0.86 %	6.27 %

Table 1: Accuracy of the delay characterization approach

The determination of the characterization constants was performed by solving the following nonlinear program that minimizes the sum of the squares of the percentage errors over all data points.

$$\text{minimize} \ \sum_{i=0}^{N} \left[\frac{D_{estim}(i) - D_{actual}(i)}{D_{actual}(i)} \right]^2 \qquad (9)$$

Here, N is the number of data points, and $D_{estim}(i)$ and $D_{actual}(i)$, respectively, represent the values given by Equation (8), and the corresponding measured value at the i^{th} data point. This nonlinear programming problem is solved using the MINOS optimization package [10] to determine the values of characterization constants.

For a library-based design, a full characterization of all cells can be carried out and its complexity is comparable to the complexity of characterizing the library using any other means. For general full custom designs, the number of SPICE data points to be generated for the curve fit increases exponentially with the number of characterization variables. It is computationally expensive to perform such a large number of simulations and hence an alternative strategy is suggested. We precharacterize a set of logic structures such that any gate can be mapped to one of the elements of this set with some acceptable loss of accuracy. It is important to emphasize that the use of these mapping strategies only serves to reduce the complexity of the characterization procedure. If one is willing to invest the CPU time required to perform the characterizations for each gate type, then this procedure is unnecessary. Gate delays are affected by the position of the switching transistor because of the discharging of internal capacitances, and the development of separate primitives for different switching scenarios is necessary to tackle this issue.

We developed a set of logic structures that can model all the static gates, including complex gates and sequential elements, with acceptable accuracy. Details about the characterization procedure and the set of logic structures used for the precharacterization are provided in [8]. Table 1 shows results of model validation on various gates. The model delay values are compared with those obtained by using SPICE. The results show that all the gate delay models have acceptable accuracy.

For the problem to be posed as a convex programming problem it is necessary to prove that the delays of individual paths satisfy the convexity properties. The proof is omitted from the discussion here, and interested readers are referred to [8].

3 Power dissipation model

In this paper we consider dynamic power dissipation.

The switching power dissipation model we use is given below.

$$P_{i_{sw}} = \frac{1}{2} \cdot V^2 \cdot C_i \cdot TD_i \tag{10}$$

where $P_{i_{sw}}$ is the average switching power dissipation, V is power supply voltage, C_i is the output capacitance, and TD_i is the transition density, all corresponding to gate n_i. The transition density is defined [11] as $\lim_{T\to\infty} n(T)/T$, where $n(T)$ represents the number of transitions the gate performs in time T. The use of transition density allows a gate with lower switching probabilities to be sized larger.

The short circuit power is known to be dependent on the input transition time to a great extent, and hence can be controlled by placing constraints on the input transition time for each gate. With proper constraints on the input transition time, the short circuit power is no more than 10% to 20% of the total power. It will be shown in next section that the transition time, τ_i, and the loading capacitance, C_i, can be expressed in terms of generalized posynomials and hence equation (10) is also a generalized posynomial.

Although we recognize the importance of leakage power in the deep submicron regime in the future, the rationale behind considering only the dynamic power is that this power continues to account for the major percentage of total power.

The proof of convexity of circuit power dissipation is similar to that of the proof of convexity of path delay and the details can be found in [8].

4 Problem formulation in detail

Traditionally posynomial programs have been solved by using geometric optimizers. In this work, we have used generalized posynomials to model the gate delay and power dissipation. While these functions are provably equivalent to convex functions, a precise relationship between generalized posynomial programming problems and posynomial programming problems would permit the use of geometric optimizers to solve the problem formulated here. In this section, we describe a technique for transforming the set of delay constraints described by generalized posynomials into a posynomial form.

4.1 Conversion of generalized posynomials to posynomials

As will be shown shortly, the transformation is carried out by the introduction of additional variables. Consider the constrained generalized posynomial below:

$$\sum_i \gamma_i \prod_{j=1}^{m} \left(\sum_{l=1}^{p_i} \omega_{ijl} \prod_{s=1}^{n} x_s^{a_{ijls}} \right)^{\beta_{ij}} \le D_{req}, \tag{11}$$

where D_{req} is the required delay time, and $\beta_{ij} > 0 \, \forall \, i, j$. Note that the term in the parenthesis is a regular posynomial. If we substitute that posynomial by the variable y_j, the result is the constraint

$$\sum_i \gamma_i \prod_{j=1}^{m} (y_j)^{\beta_{ij}} \le D_{req}. \tag{12}$$

It is well known that any constraint where a posynomial function is required to be less than or equal to a constant, is equivalent to a convex set under the variable transformation; we will refer to such a constraint as a posynomial constraint. However, the above substitution also requires that the following equality must be satisfied:

$$\sum_{l=1}^{p_i} \omega_{ijl} \prod_{s=1}^{n} x_s^{a_{ijls}} = y_j. \tag{13}$$

This equality can be represented by a pair of inequalities, only one of which is a posynomial constraint. This implies that the constraint set is no longer transformable to a convex set in the x and y variables.

We will now make use of a subtle observation. If we relax the equality in (13) into a "$\le$" inequality, then for any variable assignment that satisfies the relaxed set of constraints, since $\gamma_i > 0$ and $\beta_{ij} \ge 0$, it must be true that

$$\sum_i \gamma_i \prod_{j=1}^{m} \left(\sum_{l=1}^{p_i} \omega_{ijl} \prod_{s=1}^{n} x_s^{a_{ijls}} \right)^{\beta_{ij}} \le \sum_i \gamma_i \prod_{j=1}^{m} (y_j)^{\beta_{ij}}. \tag{14}$$

In conjunction with constraint (12), this implies that the constraint (11) must be satisfied.

Therefore, we may replace each such generalized posynomial constraint of order 1 by the set of inequalities given by

$$\begin{aligned} \sum_i \gamma_i \prod_{j=1}^{m} y_j^{\beta_{ij}} &\le D_{req} \\ \sum_{l=1}^{p_i} \omega_{ijl} \prod_{s=1}^{n} x_s^{a_{ijls}} &\le y_j \end{aligned} \tag{15}$$

It is instructive to note that this substitution technique may be used, in general, for generalized posynomials of order k. This is easily seen, since the above procedure reduces an order k generalized posynomial constraint to an order $k-1$ generalized posynomial constraint; this process may be carried out recursively until posynomial constraints are obtained.

4.2 Introduction of intermediate variables

As mentioned in section 2, the path delay can be proved to be a convex function. However, the number of input to output paths increases exponentially with an increase in the number of gates in the combinational logic. We avoid path enumeration by the introduction of intermediate variables. For each gate, n_i, we introduce the following variables:
D_{i_r} : Arrival time at the output of gate n_i for the rise transition at the output.
D_{i_f} : Arrival time at the output of gate n_i for the falling transition at the output.
τ_{i_r} : Rise transition time at the output of gate n_i.
τ_{i_f} : Fall transition time at the output of gate n_i.

In addition, we introduce variables model the pin to pin delay. D_{ij_r} represents the delay from gate n_i to gate n_j for the rise transition at the output of gate n_j. Similarly, D_{ij_f} represents delay from gate n_i to gate n_j for the fall transition at the output of gate n_j. To illustrate the constraint formation, consider a subcircuit shown below.

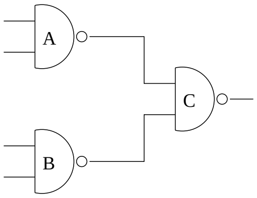

Figure 1: An example circuit

Then the constraints related to gate C are,

$$\begin{aligned}
(D_{a_f} + D_{ac_r})/D_{c_r} &\leq 1 \\
(D_{a_r} + D_{ac_f})/D_{c_f} &\leq 1 \\
(D_{b_f} + D_{bc_r})/D_{c_r} &\leq 1 \\
(D_{b_r} + D_{bc_f})/D_{c_f} &\leq 1 \qquad (16)
\end{aligned}$$

The above set of equations represents the constraints on the arrival time at the output of the gate C. All the gates are assumed to be inverting, but similar equations can be derived if noninverting gates are used. These equations along with the equations resulting from the conversion of generalized posynomials corresponding to $D_{ac_r}, D_{ac_f}, D_{bc_r}, D_{bc_f}$, to regular posynomials, and the constraints imposed on the output delay, form the complete set of posynomial equations to be solved by the geometric optimizer.

5 Experimental Results

We have implemented the transistor sizing optimization tool which integrates constraint generation program and generalized posynomial solver. Our constraint generation program writes the objective function and constraints in the generalized posynomial form. For optimization we used the new proprietary Generalized Posynomial Optimizer from Intel for many of our circuits. This solver is extremely efficient since it utilizes the properties of geometric programs to arrive at a solution. We have optimized several circuits for power, placing constraints on the input-to-output delay. We optimized the circuits for varying target delay and output transition time constraints. The results are tabulated in the Table 2. First two columns give the test circuit name and transistor count. Third column titled Dout$_{spec}$ represents the input-to-output delay constraint, while the fourth column titled τout$_{spec}$ represents the transition time constraint placed on the output. Last column represents the power obtained on optimization. The execution times of optimization, using geometric program solver, vary from about a second, for very small circuits, to about 8 minutes, for moderately sized circuits. As expected, the power dissipation increases as the constraints are made tighter.

Circuit	Transistor Count	Dout$_{spec}$ (ps)	τout$_{spec}$ (ps)	Power (mw)
inv10	20	720	400	0.191
		648	360	0.214
		576	320	0.261
		500	280	0.443
c17	24	600	400	0.634
		540	360	0.705
		480	320	0.804
		420	280	0.959
s27	42	900	400	0.648
		810	360	0.739
		720	320	0.891
		630	280	1.314
comb1	70	900	450	0.448
		800	400	0.546
		720	360	0.757
		640	320	1.574
comb2	200	1669	360	0.634
		1502	324	0.717
		1335	288	0.922
		1168	251	2.062
s298	582	1027	350	3.705
		926	315	4.141
		823	280	5.438
		720	245	5.877

Table 2: Results of sizing various circuits

6 Conclusion and future work

We have presented a generalized posynomial delay model for CMOS gates that is better suited for modern technologies than the Elmore model, but maintains the convexity properties. We have shown the use of the model in the context of optimization for low power. We have presented a new mapping technique that can be used to convert generalized posynomials to regular posynomials enabling the use of conventional mathematical optimizers. Results show that in addition to the useful property of convexity, the model is also computationally efficient. In future we plan to incorporate leakage power and present an approach that is suitable for future technologies where leakage current can no longer be neglected.

7 Acknowledgements:

We would like to thank Professor Michael Todd at Cornell University for his useful comments.

References

[1] V. De and S. Borkar, “Technology and design challenges for low power and high performance,” in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 163–168, 1999.

[2] M. R. C. M. Berkelaar, P. H. W. Buurman, and J. A. G. Jess, “Computing the entire active area/power consumption versus delay trade-off curve for gate sizing with a piecewise linear simulator,” in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 474–480, 1994.

[3] C. H. Tan and J. Allen, “Minimization of power in VLSI circuits using transistor sizing, input ordering, and statistical power estimation,” in *Proceedings of the 1994 International Workshop on Low Power Design*, pp. 75–80, 1994.

[4] S. C. Prasad and K. Roy, “Circuit optimization for minimization of power consumption under delay constraint,” in *Proceedings of the 1994 International Workshop on Low Power Design*, pp. 15–20, 1994.

[5] A. Conn, I. M. Elfadel, W. W. Molzen, P. R. O’Brien, P. N. Strenski, C. Visweswariah, and C. B. Whan, “Gradient-based optimization of custom circuits using a static-timing formulation,” in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 452–459, 1999.

[6] 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.

[7] 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.

[8] M. Ketkar, K. Kasmasetty, and S. S. Sapatnekar, “Convex delay models for transistor sizing,” in *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 655–660, 2000.

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

[10] Department of Operations Research, Stanford University, *MINOS 5.4 USER’S GUIDE*, 1995.

[11] F. N. Najm, “Transition density: A new measure of activity in digital circuits,” *IEEE Transactions on Computer-Aided Design*, vol. 12, pp. 310–323, Feb. 1993.