A Geometric Programming-based Worst-Case Gate Sizing Method Incorporating Spatial Correlation

Jaskirat Singh Vidyasagar Nookala Zhi-Quan Luo Sachin S. Sapatnekar
Department of Electrical & Computer Engineering
University of Minnesota
Minneapolis, MN 55455
{jsingh,vidya,luozq,sachin}@ece.umn.edu

***Abstract*— We present an efficient optimization scheme for gate sizing in the presence of process variations. Our method is a worst-case design scheme, but it reduces the pessimism involved in traditional worst-casing methods by incorporating the effect of spatial correlations in the optimization procedure. The pessimism reduction is achieved by employing a bounded model for the parameter variations, in the form of an *uncertainty ellipsoid*, which captures the spatial correlation information between the physical parameters. The use of the uncertainty ellipsoid, along with the assumption that the random variables, corresponding to the varying parameters, follow a multivariate Gaussian distribution, enables us to size the circuits for a specified timing yield. Using a posynomial delay model, the delay constraints are modified to incorporate uncertainty in the transistor widths and effective channel lengths due to the process variations. The resulting optimization problem is relaxed to a Geometric Program and is efficiently solved using convex optimization tools. The effectiveness of our robust gate sizing scheme is demonstrated by applying the optimization on the ISCAS '85 benchmark circuits and testing the optimized circuits by performing Monte Carlo simulations to model the process variations. Experimental results show that the timing yield of the robustly optimized circuits improves manifold over the traditional deterministically sized circuits. For the same transistor area, the circuits sized by of our robust optimization approach have, on an average, 12% fewer timing violations as compared to the gate sizing solutions obtained via the traditional, deterministically based guard-banding method.**

I. Introduction to Robust Gate Sizing

The limitations of the manufacturing process in the current technologies leads to random variations in various circuit parameters such as the transistor width, channel length, and oxide thickness, which may cause a large spread in the circuit performance measures such as the delay and power. Since it is impossible to control process-driven variations, it is essential for the design tools to account for these uncertainties to enable the design of robust circuits that are as insensitive to the device parameter variations as possible.

The optimization of gate sizes offers a degree of flexibility in addressing this issue. The gate sizing problem determines an optimal set of transistor sizes, defined as the ratio of the transistor width (w) to the effective channel length (L_e), that minimize the area or power consumption of a combinational circuit, subject to meeting the specified delay constraints. Conventional gate sizing tools employ a static timing analysis (STA) routine to generate the delay constraints by adding intermediate variables at the output of each gate in the circuit, and then solve the resulting optimization problem to determine the widths of the devices in the circuit. The minimum length is chosen for all the devices.

However, due to the fact that the nominal designs are perturbed by the random process variations, a large number of chips may fail to meet the original delay specifications. This leads to a reduction in the timing yield of the circuit, defined as the fraction of total chips whose delay does not exceed the original specified value. An obvious way to increase the timing yield of the circuit is to design for the worst-case scenario, e.g., choose a delay specification of the circuit much tighter than the required delay. Unless this new specification is appropriately selected, this could lead to large overheads in terms of the circuit area and the power, as the optimizer may have to aggressively size the critical as well as the non-critical paths. Hence, it is necessary to develop smart worst-casing methodologies in the presence of process uncertainties, that keep the area and the power budgets within reasonable bounds.

In this work, we present a novel worst-casing scheme, based on robust optimization theory. In our method, we modify the delay constraints to incorporate uncertainty in the parameters due to the process variations. An *uncertainty ellipsoid* method is used to model the random parameter variations, assuming normal distribution of parameters. Spatial correlations of intra-die parameter variations are incorporated in the optimization procedure. We impose no restriction on the sign of correlation factor, i.e., the parameters may be positively or negatively correlated. The resulting optimization problem is relaxed to a geometric program (GP), and is efficiently solved using convex optimization tools. By using the well-known *Chi-square* probability distribution function, the desired timing yield can be parameterized into the optimization formulation. Our formulation is based on the principle of adding uncertainty related, parameter correlation-aware, margins to delay constraints at the output pin of each logic gate. However, by using these guard-bands for the delay constraints at the output of each node in the circuit graph[1], instead of the whole path delay, leads to a problem of overestimation of the effect of variations. We reduce this problem by employing a graph pruning technique to reduce the number of intermediate nodes in the circuit graph, and the corresponding arrival time variables in the optimization formulation. The use of variable size uncertainty ellipsoid at different topological levels of the circuit graph helps in further removing the extra timing margins in the constraints.

[1] The graph obtained by modeling each pin of a gate as a vertex, and each pin-to-pin connection, in the whole circuit, as an edge, is referred to as the circuit graph or the timing graph.

The organization of this paper is as follows. We review the previous work on uncertainty-aware gate sizing in Section II. Section III covers the preliminaries of geometric programming, the traditional gate sizing formulation, the ellipsoid set and the Chi-square probability distribution. In Section IV, we present our formulation of the robust sizing problem, and use a simple example to explain the details of this formulation. Section IV-C points out the problem of overestimation of the effect of variations in our robust formulation. The graph pruning technique and the use of variable amounts of timing margins at different topological levels of the circuits, as methods to reduce this pessimism in the robust formulation, are described in Sections IV-D and IV-E. Experimental results are presented in Section V, and Section VI concludes this paper.

II. Previous Work

Traditional gate sizing methodologies [1], [2] solve the deterministic optimization problem of gate sizing without accounting for variations in parameters. These methods use posynomial delay constraints and formulate the problem as a geometric program. Section III-B reviews the formulation used in these conventional gate sizing works. While the method of [1] performs sizing based on a sensitivity-based heuristic, [2] offers an exact optimization algorithm to perform gate sizing, based on convex programming techniques. There have been several recent attempts to perform uncertainty-aware gate sizing to reduce the timing violations or increase the timing yield. In [3], the gate sizing problem is formulated as a nonlinear optimization problem with a penalty function added to improve the distribution of timing slacks. One of the first works on statistical gate sizing [4], proposes formulation of statistical objective and timing constraints, and solves the resulting nonlinear optimization formulation. In other works on robust gate sizing [5–8], the central idea is to capture the delay distributions by performing a statistical static timing analysis (SSTA), as opposed to the traditional STA, and then use either a general nonlinear programming technique or statistical sensitivity-based heuristic procedures to size the gates. In [9], the mean and variances of the node delays in the circuit graph are minimized in the selected paths, subject to constraints on delay and area penalty.

Some of the abovementioned variation-aware gate sizing works are heuristics [6–8] without provable optimality properties. The sensitivity-based approaches optimize the statistical cost function in a local neighborhood, and cannot guarantee convergence to the globally optimal solution. Others rely on nonlinear nonconvex optimization techniques [4], [5], [9], which are either not scalable to practical circuits or may get stuck in locally optimal solutions. Some of these works [4], [5] ignore important statistical properties of varying parameters such as the spatial correlations.

In [10], the authors present an interesting approach to optimize the statistical power of the circuit, subject to timing yield constraints under convex formulation of the problem as a second-order conic program. However, the formulation suffers from the same problem of overestimation of statistical nodal delay constraints as [11], which will be explained in Section IV-C, and we partially correct this by the techniques described in Section IV-D and IV-E. More importantly, the solution in [10] relies on a local search over the gate configuration space to identify a size that will absorb the slack assigned by the optimization solution. Such a method based on local searches has to assume that the delay of the gate depends only on the fixed local choices, e.g., a particular size and the fanout load of a gate. In reality, the gate delay is also a function of the slope of the signals at the input pins of the gate, which in turn are functions of the sizes of the fanin gates and the interconnect delay. Hence, although local search method of [10] works well for simple delay models as functions of output load only, it is unlikely to work for a realistic delay model also considering input slews.

Recently a novel method for optimizing the binning yield of a chip was proposed in [12]. This method provides a binning yield loss function that has a linear penalty for delay of the circuit exceeding the target delay, and proves the convexity of this formulation. However, the method has to rely on an SSTA engine to evaluate the gradient of the binning yield loss function for optimization purposes. This could potentially make the overall procedure considerably slow for many iterations of the optimization loop. As the objective function in the optimization formulation in this work is non-differentiable, the procedure could also run into some serious numerical problems while generating the subgradients of the objective function.

In this work, we propose a novel gate sizing technique based on robust optimization theory [13]. For simplicity, our implementation uses the Elmore delay based model, but our approach is applicable to any posynomial delay model, such as the rich class of generalized posynomial delay models proposed in [14]. In our method, we first generate posynomial constraints by performing an STA. We then add *robust constraints* to the original constraints set by modeling the intra-chip random process parameter variations as Gaussian variables, contained in a constant probability density *uncertainty ellipsoid* [15], centered at the nominal values. The method of [16] also uses the ellipsoid uncertainty model, but for optimization of small size analog circuits. We use the well known Chi-square distribution tables to assign a timing yield value in our optimization constraints. Under the ellipsoid uncertainty model, the resulting optimization formulation is relaxed to be a GP, and is efficiently solved using the convex optimization tools. Furthermore, using a GP to perform robust gate sizing ensures that the optimizer finds a global minimum, which is not guaranteed in the case of a general nonlinear program. The relaxation of the robust counterpart of the conventional deterministic GP-based gate sizing solution as another GP is a major contribution of this work; in general, it is not true that the robust versions of convex programs are also convex programs [13].

Our robust gate sizing scheme is a type of worst-case design method, but by incorporating spatial correlations in the design procedure, we reduce some pessimism in the design. Spatial intra-die correlations between the parameter variations are incorporated in the optimization scheme by using a grid-based spatial correlation model used in [17] and [18]. In addition, we show that the nodal constraints formulation adds pessimism, and reduce some of this pessimism by employing the graph pruning technique of [19]. Heuristic methods for assigning smaller timing margins at lower topological levels of the circuit graph, and increasing the guard-banding at higher levels, by employing different sized uncertainty ellipsoids, also help in reducing the effects of this pessimism.

We focus on the intra-die variations in L_e and w parameters; however, the method can be easily modified to include inter-die variations. Process-driven variations in the interconnect widths and thickness can also be included in our method. The following sections in this paper, describe in details the various steps of our robust gate sizing method.

III. Preliminaries

In this section, we will review some of the basic tools and formulations that we build on to obtain our robust optimization formulation.

A. Geometric Programming

A function is called a *monomial* function if it can be written in the form:

$$\begin{aligned} f(\mathbf{x}) &= c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n} \\ &= c \prod_{i=1}^{n} x_i^{a_i} \end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathbf{R}^{\mathbf{n}}_{++}$, $c > 0$ and $a_i \in \mathbf{R}$. The variables in a monomial function, and the coefficient c are strictly positive, and the exponents a_i can be any real numbers.

A sum of monomials is called a *posynomial* function. It can be written as:

$$f(x) = \sum_{j=1}^{k} c_j \prod_{i=1}^{n} x_i^{a_{ij}} \tag{2}$$

where $c_k > 0$.

From Equations (1) and (2), a geometric program can be defined as an optimization problem of the form:

$$\begin{aligned} &\text{Minimize} && f_0(x) \\ &\text{Subject to} && f_i(x) \leq 1, \quad i = 1, \cdots, m \\ & && h_i(x) = 1, \quad i = 1, \cdots, p \end{aligned} \tag{3}$$

where $f_0, \cdots, f_m$ are posynomial function as in Equation (2), and $h_1, \cdots, h_m$ are monomial functions as in Equation (1).

Geometric programs are not, in general, convex optimization problems. However, by a simple transformation of variables, $x_i = e^{y_i}$ in the objective and the constraint functions of Equation (3), they can be converted to a convex program [13], and hence can be efficiently and globally solved using the convex optimization methods.

B. Deterministic Gate Sizing as a Geometric Program

The conventional deterministic gate sizing problem is formulated as:

$$\begin{aligned} &\text{Minimize} \quad Area = \sum_{i=1}^{n} a_i x_{i_0} \\ &\text{Subject to:} \end{aligned} \tag{4}$$

$$\begin{cases} t_i \leq T_{spec} \quad \forall i \in PO \\ t_j + d_{ji}(\mathbf{X_0}) \leq t_i \quad \forall j \in fanin(i) \\ \vdots \\ x_{min} \leq x_{i_0} \leq x_{max} \quad \forall gate \;\; i \end{cases}$$

where x_{i_0} represents the nominal size of the gate, a_i is some weighting factor such as the number of transistors in a gate cell, t_j are the intermediate input arrival time variables at the fanin of gate i, d_{ji} is the delay of gate i, from the j^{th} input pin to the output pin, as a function of the vector $\mathbf{X_0}$ of the nominal gate sizes, T_{spec} is the specified target delay, x_{min} and x_{max} are the lower and upper bounds on the gate sizes, respectively.

Using the Elmore delay model[2], each gate i in the circuit can be replaced by an equivalent $R_{on_i} C_i$ element, where R_{on_i} represents the effective on resistance of the pull-up or the pull-down network, and the term C_i subsumes the source, drain and gate capacitances of the transistors in the gate. The expressions for R_{on_i} and C_i for a gate i are given by:

$$R_{on_i} = \frac{c_1 L_{e_i}}{w_i}, \qquad C_i = c_2 L_{e_i} w_i + c_3 \tag{5}$$

where, the constants c_1, c_2 and c_3 can be derived from [2]. Both the capacitances and the on resistance of the transistors in a gate are posynomial functions of the gate size, characterized by the widths w of the transistors in the gate. Consequently, the term $R_{on_i} C_i$, which is the equivalent delay contribution of gate i in the circuit, is also a posynomial function of w.

From Equations (4) and (5), the delay constraints at each node of the circuit graph can be written as:

$$\begin{aligned} t_i &\leq T_{spec} \quad \forall i \in PO \\ t_j + \sum_l K_l \prod_k x_{k_0}^{a_{kl}} &\leq t_i \quad \forall j \in fanin(i) \end{aligned} \tag{6}$$

where, K_l is a constant coefficient of the l^{th} monomial term in the posynomial delay expression, and can be derived from (5), x_k represents the width of gate k , and a_k is the exponents of the k^{th} components of the $\mathbf{X_0}$ vector, $\in \{-1, 0, 1\}$. By substituting Equation (6) in Equation (4) for all gates in the circuit, the conventional transistor sizing is formulated as a GP optimization problem of Equation (3), having a posynomial objective function and posynomial constraints, which can be solved using the convex optimization techniques. In Section IV, we show how the robust version of the standard GP formulation, for the deterministic case, can be converted to another GP.

C. The Ellipsoidal Uncertainty Set

For any vectors Ω and $\Omega_0 \in R^n$, and a non-singular matrix $P \in R^{n \times n}$, an ellipsoid set U is defined as [15]:

$$U = \{\mathbf{\Omega} : (\mathbf{\Omega} - \mathbf{\Omega_0})^T P^{-1} (\mathbf{\Omega} - \mathbf{\Omega_0}) \leq \psi^2\} \tag{7}$$

If P is a symmetric and positive definite matrix, an alternative representation of (7) is realized by substituting, $P^{-1/2}(\mathbf{\Omega} - \mathbf{\Omega_0}) = \mathbf{u}$ as:

$$U = \{\mathbf{\Omega_0} + P^{1/2}\mathbf{u} | \;\; ||\mathbf{u}||_2 \leq \psi\} \tag{8}$$

where $||\mathbf{u}||_2 = \mathbf{u}^T\mathbf{u}$ is the 2-norm of vector $\mathbf{u}$. For a symmetric and positive definite matrix P, the matrix $P^{1/2}$ can be computed by the eigen decomposition of P. The ellipsoid

[2] Traditional gate sizing methods of [1] and [2] also use the Elmore delay. In any GP based formulation, the Elmore delay model is used for simplicity. Alternatively, generalized posynomial delay models [14], which have a higher accuracy, can be used for the GP formulation.

represents a n-dimensional region, where the vector Ω varies around the center point Ω_0. The vector $\mathbf{u}$ characterizes the movement of Ω around Ω_0.

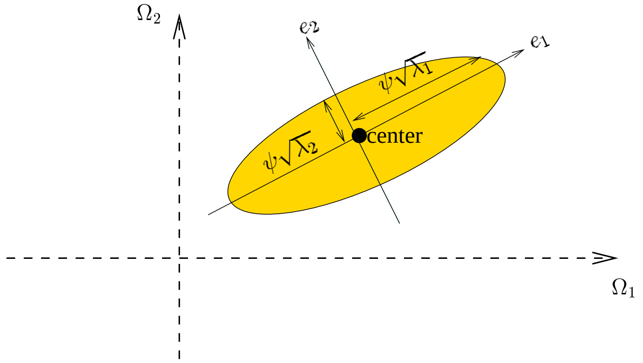

Fig. 1. An uncertainty ellipsoid set in two dimensions. The ellipsoid set is used as a bounded model for multivariate normal parameter variations.

Figure 1 illustrates the ellipsoid in R^2. The half-lengths of the axis of the ellipsoid are a factor ψ of the square roots of the eigenvalues, λ_1 and λ_2, of the matrix P, and the direction of the axis is given by the eigenvectors of P, e_1 and e_2.

Considering the vector $\mathbf{\Omega}$ to consist of random variables corresponding to the parameters of variations, with an associated covariance matrix given by P, and assuming that the parameters of variation follow a Gaussian distribution, the ellipsoid set described in Equations (7) and (8), can be used as a bounded model of variations. In particular, it can be shown that the constant probability density contours of a multivariate normal distribution represent an ellipsoid set. The joint probability distribution function (PDF) of the multivariate normal random vector $\mathbf{\Omega}$, with a covariance matrix P is:

$$f_\Omega(\mathbf{\Omega}) = \frac{1}{(2\pi)^{n/2}|P|^{1/2}} e^{\left\{-\frac{1}{2}(\mathbf{\Omega}-\mathbf{\Omega_0})^T P^{-1}(\mathbf{\Omega}-\mathbf{\Omega_0})\right\}} \tag{9}$$

where $|P|$ is the determinant of the covariance matrix P, and n is the number of components in the variation vector $\mathbf{\Omega}$. It is clear from Equation (9), that the PDF of a multivariate normal distribution would be a constant c, if $(\mathbf{\Omega} - \mathbf{\Omega_0})^T P^{-1}(\mathbf{\Omega} - \mathbf{\Omega_0}) = c$. This relation represents precisely the surface of an ellipsoid given by Equation (7), with $c = \psi^2$. Since the covariance matrix P is symmetric and positive definite [15], we can also equivalently represent the constant probability ellipsoid as Equation (8). Thus from the discussion above, by assuming normality of parameter distribution, the ellipsoid set can be regarded as a high-dimensional region inside which the parameters randomly vary. This bounded model of parameter variations in the form of an ellipsoid set is referred to as an *uncertainty ellipsoid*. In Section IV, we use this uncertainty ellipsoid model to simplify our robust constraints and formulate the robust GP optimization problem.

D. Chi-square Distribution

If r_i are n independent normally distributed random variables with means μ_i and variances σ_i^2, the random variable $z = \sum_{i=1}^{n}(\frac{r_i-\mu_i}{\sigma_i})^2$ is distributed according to the Chi-square distribution (χ_n^2), with n degrees of freedom [15]. The Chi-square distribution is a special case of gamma distribution, and for a random variable z following the Chi-square distribution, the cumulative density function (CDF) of z is given by [20]:

$$F(z;n) = \frac{\gamma(n/2, z/2)}{\Gamma(n/2)} \tag{10}$$

where Γ is the gamma function, and γ is the incomplete gamma function [20].

Referring back to Equation (7), it can be proved that the random variable $z = (\mathbf{\Omega} - \mathbf{\Omega_0})^T P^{-1}(\mathbf{\Omega} - \mathbf{\Omega_0})$ is χ_n^2 distributed [15]. Therefore, the solid ellipsoid given by Equation (7) can be assigned a prespecified amount of probability α as:

$$\alpha = F_{\chi_n^2}(\psi^2) \tag{11}$$

where F is the Chi-square CDF function given by Equation (10).

As will be explained in Section IV, we use the uncertainty ellipsoid to pad the deterministic delay constraints, and with the prespecified probability α given by the lower bound on timing yield specification, we define the size of the ellipsoid. This determines the amount of margin required for each delay constraint.

IV. Variation-Aware Gate Sizing

A. Effect of Variations on Constraints

The deterministic posynomial constraints of (6) can be represented as:

$$t_j + f_{ji}(\mathbf{X_0}) \quad \leq \quad t_i \tag{12}$$

where $t_j + f_{ji}(\mathbf{X_0}) = \mathbf{t_j} + \sum_\mathbf{l} \mathbf{K_l} \prod_\mathbf{j} \mathbf{x_{j0}^{a_{jl}}}$ represents the j^{th} constraint function, $\mathbf{X_0}$ is the vector representing the nominal gate sizes x_{0_i} for all gates. The conventional GP optimization assigns a set of optimal x_0 to the vector $\mathbf{X_0}$, so that each delay constraint is satisfied, i.e., $t_j + f_i(\mathbf{X_0}) \leq t_i$ for all constraints i, and the area objective is minimized.

However, due to the effect of process variations, the posynomial delay models of the gate can no longer be assumed to be deterministic quantities. Thus, the constraint inequalities at each node should be rewritten as:

$$t_j + f_{ij}(\mathbf{X_0}, \mathbf{\Omega}) \quad \leq \quad t_i \tag{13}$$

where $\mathbf{\Omega}$ is the random vector of perturbations around the nominal values of the parameters. For the cases when the new value of the constraint function $t_j + f_{ji}(\mathbf{X_0}, \mathbf{\Omega}) > t_i$, the effect of the random process variations leads to the original constraints being violated and a possible timing failure for the circuit.

Assuming that the random parameter perturbations around the nominal values are small, the new value of the gate delay model $f_i(\mathbf{X_0}, \mathbf{\Omega})$ can be approximated by a first order Taylor series expansion as:

$$\begin{aligned} f_{ji}(\mathbf{X_0}, \mathbf{\Omega_0} + \delta\mathbf{\Omega}) &= f_{ji}(\mathbf{X_0}, \mathbf{\Omega_o}) + \sum_j \left.\frac{\delta f_{ji}(\mathbf{X_0}, \mathbf{\Omega})}{\delta(\Omega_j)}\right|_{\Omega_{j0}} (\Omega_j - \Omega_{j0}) \\ &= f_{ji}(\mathbf{X_0}, \mathbf{\Omega_0}) + \nabla_{\mathbf{\Omega_0}} f_{ji}(\mathbf{X_0}, \mathbf{\Omega})\delta\mathbf{\Omega} \\ &= \sum_l K_l \prod_j x_{j0}^{a_{jl}} + \nabla_{\mathbf{\Omega_0}}(\sum_l K_l \prod_j x_j^{a_{jl}} \delta\Omega) \end{aligned} \tag{14}$$

where $\nabla_{\mathbf{\Omega_0}}$ represents the gradient calculated at the nominal values of the parameters, and $\delta\mathbf{\Omega}$ represents the zero-mean random variation in the parameters such as transistor width, effective channel length and oxide thickness, around the nominal values. Note that the coefficient K_l also depends on the parameters, and therefore should be regarded as a function $K_l(\mathbf{\Omega})$ of the perturbation vector.

In (14) the term, $\nabla_{\mathbf{\Omega_0}}(\sum_l K_l \prod_j x_j^{a_j})\delta\mathbf{\Omega}$ is the variational term representing the effect of process variations, added to the nominal term $\sum_l K_l \prod_j x_{j_0}^{a_j}$. To safeguard against the uncertainty of process variations, it is necessary to meet the constraint, $t_j + f_i(\mathbf{X_0}, \mathbf{\Omega}) < t_i$, for the maximum value of the variational term. In other words:

$$\begin{aligned} & t_j + \sum_l K_l \prod_j x_{j_0}^{a_{jl}} + \\ & \max_{\forall \delta\Omega \in U} (\nabla_{\mathbf{\Omega_0}} (\sum_l K_l \prod_j x_j^{a_{jl}} \delta\Omega) \quad \leq t_i \end{aligned} \tag{15}$$

Next, we show that by employing the concept of an uncertainty ellipsoid U, the constraint of (15) can be transformed to a set of posynomial constraints, so that the robust optimization formulation remains a GP, and can be efficiently solved. Our robust GP formulation is applicable for all cases where the original constraints are in the form of a generalized posynomial [14].

We use the uncertainty ellipsoid to model the process variations that randomly perturb the transistor parameters around the nominal values for which they were designed. As the random vector $\mathbf{\Omega}$ of uncertain parameters varies around the nominal parameter vector Ω_0, the variations are considered to be bounded within the ellipsoid regions defined by (8). In other words, referring to Equation (8), the variation $\delta\Omega$ from Ω_0 is given by $\delta\Omega = P^{1/2}\mathbf{u}$ with $\|\mathbf{u}\|_2 \leq \psi$.

Alternatively, we could have chosen the variation $\delta\Omega$ in the parameters to be bounded in an n-dimensional box given by $\Omega_{min} \leq \delta\mathbf{\Omega} \leq \Omega_{max}$. However, using the box as a model for bounded variation, ignores any correlation information between the random components of $\mathbf{\Omega}$, as each component can move independently inside a box, assuming any values between the minimum and maximum range. Thus, optimizing for a maximum variation in such a box region would translate to an overly pessimistic design. Moreover, an n-dimensional box modeling of parameter variations would be accurate only in the highly unlikely case when all parameters are statistically independent with respect to each other, and follow a uniform distribution. Most parameters have been observed to follow a distribution that resembles a Gaussian one. The advantage of using the ellipsoid uncertainty model is that it not only accurately models the region of variation for normally distributed parameters, any correlations between the parameters is directly captured by appropriately constructing the elements of the covariance matrix P. The covariance matrix can be derived from a spatial correlation model such as the ones used in [17] and [18].

In the next section, we show with the aid of a small example, the use of the uncertainty ellipsoid model in converting the constraint of (15) to a set of posynomial constraints, and formulating the robust GP for gate sizing in the presence of process variations.

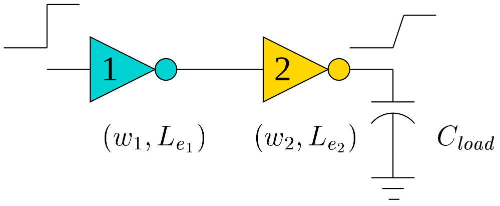

Fig. 2. A simple example circuit to explain the geometric program formulation for robust gate sizing problem.

B. Robust GP formulation

We use a simple example to explain the procedure to incorporate the process variation effects in the delay constraints set. We use the toy circuit of Figure 2, comprising of just one driver gate and one load gate, for this illustration, but the idea can be generalized to arbitrarily large circuits. In this example, we consider the widths (w_1, w_2) and the effective channel lengths (L_{e_1}, L_{e_2}) of the two gates as the only varying parameters. The scheme can be directly extended to include other parameters.

Applying the Elmore delay model to the gates of circuit of Figure 2, and for simplicity, neglecting the interconnect delay and the effect of drain and source capacitances of the driver gate, the delay constraint for the circuit can be written as:

$$\frac{K_1 L_{e_1} L_{e_2} w_2}{w_1} + \frac{K_2 L_{e_2}}{w_2} \quad \leq \quad T_{spec} \tag{16}$$

where K_1 and K_2 are constants. As explained in Section IV, to ensure that the delay constraint of (16) is met under the effect of random process variations, the first order Taylor series expansion of the constraint function results in the following relation:

$$\begin{aligned} & \frac{K_1 L_{e_{1_0}} L_{e_{2_0}} w_{2_0}}{w_{1_0}} + \frac{K_2 L_{e_{2_0}}}{w_{2_0}} + \\ \max_{\forall \delta w, \delta L_e \in U} & \Bigg(\frac{K_1 L_{e_{1_0}} L_{e_{2_0}} \delta w_2}{w_{1_0}} + \frac{K_1 L_{e_{2_0}} w_{2_0} \delta L_{e_1}}{w_{1_0}} + \\ & \frac{K_1 L_{e_{1_0}} w_{2_0} \delta L_{e_2}}{w_{1_0}} + \frac{K_2 \delta L_{e_2}}{w_{2_0}} - \\ & \frac{K_1 L_{e_{1_0}} L_{e_{2_0}} w_{2_0} \delta w_1}{w_{1_0}^2} - \frac{K_2 L_{e_{2_0}} \delta w_2}{w_{2_0}^2} \Bigg) \quad \leq T_{spec} \end{aligned} \tag{17}$$

where w_0 and L_{e_0} represent, respectively, the nominal values of the transistor w and L_e, and δw and δL_e are, respectively, the random variations in w and L_e. Employing the ellipsoid uncertainty model of (8) for the random parameter variations, leads to:

$$\begin{bmatrix} \delta w_1 \\ \delta w_2 \\ \delta L_{e_1} \\ \delta L_{e_2} \end{bmatrix} = \begin{bmatrix} (P^{1/2}\mathbf{u})_1 \\ (P^{1/2}\mathbf{u})_2 \\ (P^{1/2}\mathbf{u})_3 \\ (P^{1/2}\mathbf{u})_4 \end{bmatrix} \tag{18}$$

where P is the covariance matrix of the random vector $\mathbf{\Omega}$ consisting of the variations in gate w and L_e of the driver and the load gate of Figure 2, and $\mathbf{u}$ is the vector bounding the variation within the 4-dimensional ellipsoid centered at the nominal values of w and L_e, with $\|\mathbf{u}\|_2 \leq \psi$.

We introduce a vector ϕ to collect the coefficients of the variational parameters of (17) as:

$$\phi = \begin{bmatrix} \frac{-K_1 L_{e_{1_0}} L_{e_{2_0}} w_{2_0}}{w_{1_0}^2} \\ \frac{K_1 L_{e_{1_0}} L_{e_{2_0}}}{w_{1_0}} - \frac{K_2 L_{e_{2_0}}}{w_{2_0}^2} \\ \frac{K_1 L_{e_{2_0}} w_{2_0}}{w_{1_0}} \\ \frac{K_1 L_{e_{1_0}} w_{2_0}}{w_{1_0}} + \frac{K_2}{w_{2_0}} \end{bmatrix} \quad (19)$$

From the definitions in (18) and (19), (17) can be rewritten as:

$$\frac{K_1 L_{e_{1_0}} L_{e_{2_0}} w_{2_0}}{w_{1_0}} + \frac{K_2 L_{e_{1_0}}}{w_{2_0}} + \max_{\forall \mathbf{u}} \left(\langle P^{1/2}\phi, \mathbf{u} \rangle \right) \leq T_{spec} \quad (20)$$

where $\langle a, b \rangle$ represents the inner product of vectors a and b.

Since the covariance matrix P is symmetric and positive definite [15]:

$$P^{1/2}\phi = Q \; diag(\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_n}) \; Q^T \phi \quad (21)$$

where Q is the matrix containing eigenvectors of P, and $\lambda_i, \cdots, \lambda_n$ are the n eigenvalues of P. Next, defining $M = P^{1/2} = Q \; diag(\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_n}) \; Q^T$, the positive and negative terms of the elements of vector $M\phi$ can be separated as:

$$P^{1/2}\phi = M\phi = \eta_{\mathbf{1}} + \eta_{\mathbf{2}} \quad (22)$$

where $\eta_{\mathbf{1}}$ and $\eta_{\mathbf{2}}$ contain all positive and negative terms, respectively, of the elements of the vector[3] $M\phi$.

From the well-known result of the Cauchy Schwartz inequality[4]:

$$< a, b > \quad \leq \quad ||a||_2 \cdot ||b||_2 \quad (23)$$

and from Equations (21) and (22), along with the fact that in the ellipsoid uncertainty model, $||\mathbf{u}||_2 \leq \psi$, a sufficient condition[5] for (20) is:

$$\frac{K_1 L_{e_{1_0}} L_{e_{2_0}} w_{2_0}}{w_{1_0}} + \frac{K_2 L_{e_{1_0}}}{w_{2_0}} + \psi||\eta_{\mathbf{1}}||_2 + \psi||\eta_{\mathbf{2}}||_2 \leq T_{spec} \quad (24)$$

We then introduce two additional *robust variables* r_1 and r_2 as:

$$\begin{aligned} r_1 &= \psi||\eta_{\mathbf{1}}||_2, \quad \text{i.e.,} \quad r_1^2 = \psi^2 {\eta_{\mathbf{1}}}^T \eta_{\mathbf{1}} \\ r_2 &= \psi||\eta_{\mathbf{2}}||_2, \quad \text{i.e.,} \quad r_2^2 = \psi^2 {\eta_{\mathbf{2}}}^T \eta_{\mathbf{2}} \quad (25) \end{aligned}$$

The inequality of (24) is then replaced by the following relaxed constraints:

$$\begin{aligned} \frac{K_1 L_{e_{1_0}} L_{e_{2_0}} w_{2_0}}{w_{1_0}} + \frac{K_2 L_{e_{1_0}}}{w_{2_0}} + r_1 + r_2 &\leq T_{spec} \quad (26) \\ \psi^2 {\eta_{\mathbf{1}}}^T \eta_{\mathbf{1}} r_1^{-2} &\leq 1 \quad (27) \\ \psi^2 {\eta_{\mathbf{2}}}^T \eta_{\mathbf{2}} r_2^{-2} &\leq 1 \quad (28) \end{aligned}$$

As the optimizer tries to minimize the value of the robust variables r_1 and r_2, the relaxed inequality constraints of (27) and (28) would enforce the equality constraint of Equation (25).

The inequality of (26) is clearly a posynomial with the robust variables r_1 and r_2 added to the original variable list of the gate w and the intermediate arrival time variables t (not used in this example). From Equation (22), by construction, all the elements of $\eta_{\mathbf{1}}$ are posynomials, and all the elements of $\eta_{\mathbf{2}}$ are negative of posynomials. Thus, the quadratic terms ${\eta_{\mathbf{1}}}^T \eta_{\mathbf{1}}$, and ${\eta_{\mathbf{2}}}^T \eta_{\mathbf{2}}$ are a summation of monomials with positive coefficients. Consequently, the constraints of (27) and (28) are also posynomials. Hence, by following the procedure described in the above equations, we convert the non-robust posynomial constraint of (16) to a set of robust posynomial constraints of (26-28), by introducing two additional variables. It is worth emphasizing that unlike [11], the robust GP formulation presented in this section does not restrict the elements of the P matrix to be only nonnegative, i.e., the method can handle both positively and negatively correlated parameters.

Next, we address the issue of assigning a timing yield parameter to the optimization formulation. As discussed in Section III-D, we can assign a prespecified probability α to the uncertainty ellipsoid model of variations by using the χ_n^2 distribution. From Equation (11), we can determine ψ^2 as the upper $100\alpha^{th}$ percentile of the χ_n^2 distribution from the standard tables of the Chi-square CDF. For instance, for the example circuit of Figure 2, corresponding to $\alpha = 0.9$ or 90%, the value of ψ determined from the χ_4^2 CDF tables, for the four-dimensional ellipsoid, is $\psi = 2.79$. The value assigned to ψ, determines the size of the uncertainty ellipsoid used to pad the nominal terms in the timing constraints. The prespecified probability α serves as the lower bound on the timing yield, because the robust constraints formulated using the ellipsoid margin corresponding to such an α, would be satisfied for at least $\alpha\%$ of all cases. Since there are other points outside the ellipsoid set of the specified probability value that may not cause timing violations, the timing yield could be more than α.

For a general circuit, the procedure described for the example circuit of Figure 2 is repeated for each constraint. Thus, by addition of at most two additional variables for each constraint, robustness against the process uncertainties is added to the original constraint set, while still maintaining the desirable posynomial structure of the constraints. By this procedure, we convert the conventional GP formulation of the gate sizing problem to a robust gate sizing problem, which is also a GP and hence, can be efficiently solved using the convex optimization machinery.

C. Overestimation of Variations

The optimization formulation described in Section IV, adds margins to the deterministic constraints generated by an STA procedure. Due to the fact that separate margins are added at each node of the circuit graph, instead of the whole path, the resulting formulation could result in a large overestimation of the variational component of the circuit delay, which could lead to excessive design penalties.

To understand the problem of this overestimation of variation, consider a simple example circuit consisting of m chain of inverters as shown in Figure 3. For this simple circuit,

[3] Note that the eigen decomposition of the P matrix, to obtain $M = P^{1/2}$, has a one time cost associated with it. For a given correlation model, the covariance matrix P does not change for different circuits or different placements of a circuit. Hence, the eigen decomposition of P can be obtained in a precharacterization step.

[4] In our case, the equality in (23) also holds, as there are some points in the ellipsoid set which have $\langle P^{1/2}\phi, \mathbf{u} \rangle = ||P^{1/2}\phi||_2 \cdot ||\mathbf{u}||_2$.

[5] An equivalent condition for (20) is: $\left(\frac{K_1 L_{e_{1_0}} L_{e_{2_0}} w_{2_0}}{w_{1_0}} + \frac{K_2 L_{e_{1_0}}}{w_{2_0}} + \psi||(\eta_{\mathbf{1}} + \eta_{\mathbf{2}})||_2 \right) \leq T_{spec}$. However, this does not lead to the formulation of posynomial constraints of (27) and (28).

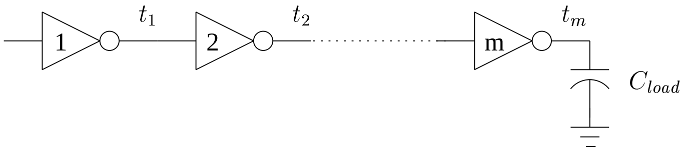

Fig. 3. An example of a chain of inverters circuit to explain the problem of overestimation of variations in the robust GP formulation.

an STA module would generate the following block-based constraints:

$$
\begin{array}{rcl}
d_1(\mathbf{X_0}) & \leq & t_1 \\
t_1 + d_2(\mathbf{X_0}) & \leq & t_2 \\
 & \vdots & \\
t_{m-1} + d_m(\mathbf{X_0}) & \leq & t_m \\
t_m & \leq & T_{spec}
\end{array}
\qquad (29)
$$

where d_i is the delay of the i^{th} inverter, which is a function of the vector of nominal gate sizes $\mathbf{X_0}$. By the method explained in Section IV, the equivalent robust constraints for the example circuit of Figure 3, can be written as:

$$
\begin{array}{rcl}
d_1(\mathbf{X_0}) + \max\limits_{\forall \delta\Omega \in U} (\nabla_{\mathbf{\Omega_0}} d_1(\mathbf{X_0}, \mathbf{\Omega})\delta\mathbf{\Omega}) & \leq & t_1 \\
t_1 + d_2(\mathbf{X_0}) + \max\limits_{\forall \delta\Omega \in U} (\nabla_{\mathbf{\Omega_0}} d_2(\mathbf{X_0}, \mathbf{\Omega})\delta\mathbf{\Omega}) & \leq & t_2 \\
 & \vdots & \qquad (30) \\
t_{m-1} + d_m(\mathbf{X_0}) + \max\limits_{\forall \delta\Omega \in U} (\nabla_{\mathbf{\Omega_0}} d_m(\mathbf{X_0}, \mathbf{\Omega})\delta\mathbf{\Omega}) & \leq & t_m \\
t_m & \leq & T_{spec}
\end{array}
$$

It is easy to see that for the simple circuit of Figure 3, the delay is given by the whole path delay as $d_1(\mathbf{X_0}, \mathbf{\Omega}) + \cdots + d_m(\mathbf{X_0}, \mathbf{\Omega})$. Thus, the effect of variations can be accounted for by a simple robust constraint of the form:

$$
\begin{array}{r}
d_1(\mathbf{X_0}) + \cdots + d_m(\mathbf{X_0}) + \qquad (31) \\
\max\limits_{\forall \delta\Omega \in U} (\nabla_{\mathbf{\Omega_0}} (d_1(\mathbf{X_0}, \mathbf{\Omega})) + \cdots + d_m(\mathbf{X_0}, \mathbf{\Omega})\delta\mathbf{\Omega}) \quad \leq T_{spec}
\end{array}
$$

For any m nonnegative functions, $y_1, \cdots, y_m$, the following inequality is well-known:

$$\max(y_1 + \cdots + y_m) \leq \max y_1 + \cdots + \max y_m \qquad (32)$$

Therefore, for the variation terms in the constraints of (30) and (31), the following inequality holds:

$$\max_{\forall \delta\Omega \in U} (\nabla_{\mathbf{\Omega_0}} \sum_i d_i(\mathbf{X_0}, \mathbf{\Omega})\delta\mathbf{\Omega}) \leq \sum_i \max_{\forall \delta\Omega \in U} (\nabla_{\mathbf{\Omega_0}} d_i(\mathbf{X_0}, \mathbf{\Omega})\delta\mathbf{\Omega}) \qquad (33)$$

It is clear from (30), (31) and (33), that the approach of adding the variational component of delay at each node leads to extra guard-banding.

Another way to understand the amount of pessimism introduced in the formulations is by realizing that the actual probability of failure, p_{fail1}, for the circuit of Figure 3 is given by:

$$p_{fail1} = Pr(d_1(\mathbf{X_0}, \mathbf{\Omega}) + \cdots + d_m(\mathbf{X_0}, \mathbf{\Omega})) > T_{spec} \qquad (34)$$

On the other hand, the probability of failure, p_{fail2}, as computed by the padding of constraints at the each node in the circuit graph of Figure 3 is given by:

$$
\begin{array}{r}
p_{fail2} = [Pr(d_1(\mathbf{X_0}, \mathbf{\Omega}) > t_1)] \cup [Pr(t_1 + d_2(\mathbf{X_0}, \mathbf{\Omega}) > t_2)] \cup \\
\cdots \cup [Pr(t_{m-1} + d_m(\mathbf{X_0}, \mathbf{\Omega}) > T_{spec})] \quad (35)
\end{array}
$$

Clearly, from Equations (34) and (35), $p_{fail1} \leq p_{fail2}$. Thus, the robust GP formulation attempts to safeguard against a probability of timing failure that is greater than the actual failure probability, which could lead to extra design margins.

For a simple circuit similar to the one in Figure 3, it is trivial to trace the path delay, and then add margin to the whole path delay constraint. However, in general, the number of paths in a circuit graph can be exponential in the number of nodes. Therefore, enumeration of paths has a prohibitive cost for large circuits consisting of thousands of gates.

To reduce the problem of unnecessary padding at the intermediate nodes in the circuit, without incurring the exponential cost of formulating the path-based constraints, we employ a graph pruning technique proposed in [19]. The following section discusses this pruning method.

D. Graph Pruning

In [19], the authors propose a technique to reduce the number of variables, constraints and redundancy in the circuit optimization formulation, by removing the internal nodes and the original edges connected to them in the circuit graph. We adapt this graph pruning technique to our method to reduce the pessimism in our gate sizing formulation.

This technique alters the delay constraints formulation by operating on the timing graph of the circuit. An initial timing graph of the circuit is constructed by representing each pin of a gate in the circuit as a vertex, and the connections between an input and an output pin of the same gate, and between an output pin of a gate and an input pin of its fanout gate, as edges in the graph. The arrival time at a pin of a gate is used to annotate the edge originating at the node corresponding to that pin. Two additional nodes, representing the primary inputs (PI) and primary outputs (PO) are added to the vertex set of the graph. Figure 4 shows a simple circuit and its corresponding timing graph.

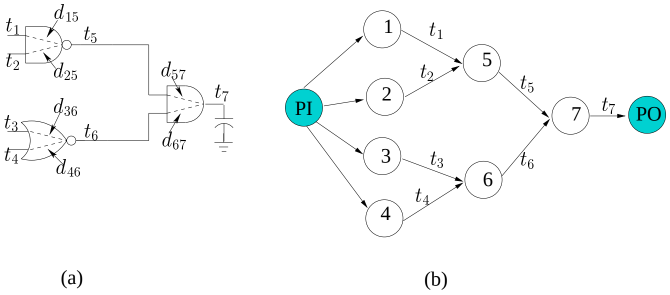

Fig. 4. A simple example circuit to illustrate the graph pruning method. (a) A two-level combinatorial circuit. (b) Timing graph for the circuit.

In the graph pruning method, the nodes of the graph are iteratively screened for a possible elimination by evaluating the cost of this node removal. The cost is typically expressed as some simple function of change in the number of variables and constraints in the optimization formulation, after the vertex under consideration is removed from the graph. If the evaluated cost is negative, implying a reduction in the problem size, the node is removed, and subsequently all incoming and outgoing edges of this node are also pruned from the graph.

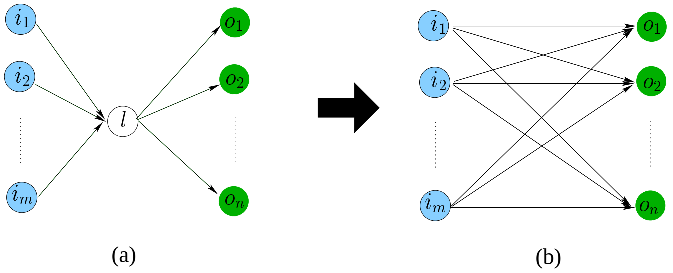

Fig. 5. A segment of the timing graph of a circuit to illustrate the removal of a node in the graph pruning method. (a) The original graph segment. (b) The graph segment after pruning node l.

The change in the formulation of delay constraints by a node removal can be understood by considering a segment of a circuit graph shown in Figure 5. In the above figure, we assume that node l meets the removal criterion according to the pruning cost. This node has m fanins, $i_1, \cdots, i_m$, and n fanouts, $o_1, \cdots, o_m$. The timing constraints for this graph segment before the node removal, as depicted by the graph segment of Figure 5(a) are:

$$\begin{aligned} t_{i_k} + d_{i_k,l} &\leq t_l \quad \forall k \in 1, \cdots, m \\ t_l + d_{l,o_j} &\leq t_{o_j} \quad \forall j \in 1, \cdots, n \end{aligned} \tag{36}$$

After eliminating node l, and the corresponding arrival time variable t_l, from the above constraint set, we obtain:

$$t_{i_k} + d_{i_k,l} + d_{l,o_j} \leq t_{o_j} \quad \forall k \in 1, \cdots, m, \quad \forall j \in 1, \cdots, n \tag{37}$$

These new constraints are shown graphically in Figure 5(b). The two sets of constraints in (36) and (37) are equivalent, and no timing information is lost in transforming from one set to the other. Since the pruning cost determines the nodes to be removed, a cost function constructed to reduce the problem size, e.g., a weighted sum of change in the number of variables and number of constraints, results in making the optimization formulation more compact after every pruning step.

1) Example of the Pruning Procedure: The application of the graph pruning method of [19] to reduce the pessimism in our optimization formulation can be best explained using a simple example circuit, and its corresponding timing graph. For this we refer back to the circuit of Figure 4. As shown in the figure, the arrival times at each pin of the logic gates are represented by variables $t_1, \cdots, t_7$. For simplicity, it is assumed that the interconnects have zero delay and that all primary inputs arrive at a time $t = 0$. The d_{ji} variables in Figure 4(a), represent the pin to pin delay of a logic gate. Figure 4(b) shows the corresponding timing graph for the example circuit. By employing an STA procedure, the delay constraints at the output of pin of each gate in the circuit of Figure 4(a) can be written as:

$$\begin{aligned} 0 &\leq t_i \quad i \in \{1,2,3,4\} \\ t_1 + d_{15}(\mathbf{X_0}, \mathbf{\Omega}) &\leq t_5 \\ t_2 + d_{25}(\mathbf{X_0}, \mathbf{\Omega}) &\leq t_5 \\ t_3 + d_{36}(\mathbf{X_0}, \mathbf{\Omega}) &\leq t_6 \\ t_4 + d_{46}(\mathbf{X_0}, \mathbf{\Omega}) &\leq t_6 \\ t_5 + d_{57}(\mathbf{X_0}, \mathbf{\Omega}) &\leq t_7 \\ t_6 + d_{67}(\mathbf{X_0}, \mathbf{\Omega}) &\leq t_7 \\ t_7 &\leq T_{spec} \end{aligned} \tag{38}$$

where $\mathbf{X_0}$ is the vector consisting of the sizes of the three gates of Figure 4(a), and $\mathbf{\Omega}$ is the random vector corresponding to the process uncertainties. From the discussion in Section IV-C, adding margins for each of the constraints of (38) can result in excessive guard-banding against the effect of variations, and hence a pessimistic design.

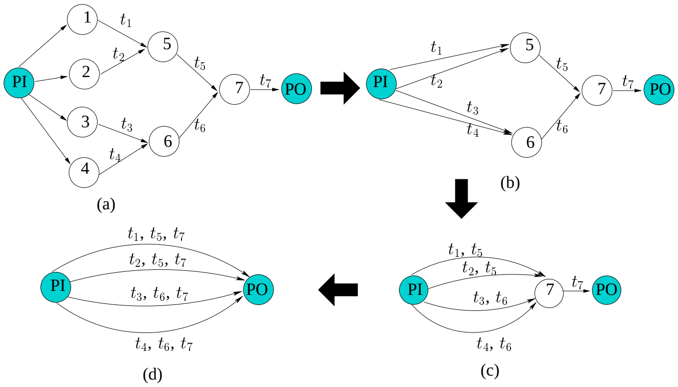

Fig. 6. The graph pruning method applied to the example circuit of Figure 4. (a) The original circuit graph. (b) Graph after removing nodes 1, 2, 3 and 4. (c) Graph after removing nodes 5 and 6. (d) The final pruned graph.

As described in the previous section, the circuit timing graph of Figure 4(b), and the corresponding constraints formulation of (38) can be altered by selectively removing nodes from the graph. Figure 6 illustrates the application of the graph pruning technique on the example circuit of Figure 4. For this specific example, the pruning cost chosen is simply the difference in the number of variable and constraints after removing a node from the graph. Figure 6(a) shows the graph obtained after eliminating nodes 1, 2, 3 and 4 in the original graph. Similarly, Figure 6(b) represents the graph after removing nodes 5 and 6, as well. The final pruned graph, obtained after removing all nodes except the PI and the PO nodes is shown in Figure 6(d). For each pruned node, a new edge is added between the fanin and fanout nodes of the removed node, and the new edge is annotated with the pruned arrival times. This annotation is required to generate the timing constraints at the end of the pruning procedure.

From the edge annotations, and the original constraints of (38), the constraints corresponding to the final pruned circuit graph of Figure 6(d) can be written as:

$$\begin{aligned} d_{15}(\mathbf{X_0}, \mathbf{\Omega}) + d_{57}(\mathbf{X_0}, \mathbf{\Omega}) &\leq T_{spec} \\ d_{25}(\mathbf{X_0}, \mathbf{\Omega}) + d_{57}(\mathbf{X_0}, \mathbf{\Omega}) &\leq T_{spec} \\ d_{36}(\mathbf{X_0}, \mathbf{\Omega}) + d_{67}(\mathbf{X_0}, \mathbf{\Omega}) &\leq T_{spec} \\ d_{46}(\mathbf{X_0}, \mathbf{\Omega}) + d_{67}(\mathbf{X_0}, \mathbf{\Omega}) &\leq T_{spec} \end{aligned} \tag{39}$$

In the above set of constraints, the pruning method eliminates all nodes, except the ones corresponding to primary inputs and the primary output. Since all intermediate arrival time variables t_i are pruned, the above formulation does away with the problem of keeping redundant margins for the constraints at the output pin of each node. It should be emphasized that the example circuit of Figure 4 is an extremely simple case for which the pruning method can eliminate all intermediate nodes, and arrive at the path delay constraints of (39). Therefore, the problem of overestimation of effect of variation, as described in Section IV-C is completely resolved for this example circuit. In general, for practical circuits, the graph pruning procedure could determine some nodes unsuitable for pruning, and some intermediate nodes could still remain in the final pruned circuit graph. However, due to the removal of many intermediate nodes, the pessimism in the robust optimization formulation is considerably reduced.

2) Practical Issues in Using Graph Pruning for the Robust GP Formulation: By removing a node with m fanins and n fanouts from the circuit graph, the change Δ_{con}, in the number of constraints is $\Delta_{con} = 2(mn - (m+n))$, and the change Δ_{var}, in the number of variables is $\Delta_{var} = -2$, as the variables corresponding to both rise and fall delays of the pruned node are eliminated. A pruning criterion can thus be established as some function $f_{cost}(\Delta_{con}, \Delta_{var})$, of change in the number of variables and constraint. The pruning procedure operates iteratively, in which the nodes with the lowest nonpositive f_{cost} are pruned in the first pass. After the first iteration, the number of fanins and fanouts of the unpruned nodes are recalculated due to the addition of new edges in the pruned graph. This iterative method continued until all nodes in the graph produce a positive f_{cost}. At this point, no more nodes can be removed from the graph according to the given pruning metric. Typically, the pruning criterion is chosen as $f_{cost} = a.\Delta_{con} + b.\Delta_{var}$, where a and b are some normalized weighting factors. However, due to some practical problems in applying the graph pruning method to our formulation, we use a slightly modified pruning cost function. The following discussion explains these practical issues.

From (37), the number of d_{ji} terms, corresponding to the posynomial gate delay models, increase in every constraint during the pruning procedure. This results in the following problem for our robust GP formulation. Referring back to our robust GP method described in Section IV-B, we modify each delay constraint to include the terms corresponding to the maximum effect of variations inside the bounded uncertainty ellipsoid model. This is achieved by adding to each constraint, new robust variables r_1 and r_2, defined in Equation (25), and including additional constraints to the formulation, given by (27) and (28), as $\psi^2 {\boldsymbol{\eta_1}}^T \boldsymbol{\eta_1} r_1^{-2} \leq 1$ and $\psi^2 \boldsymbol{\eta_2}^{\mathbf{T}} \boldsymbol{\eta_2} r_2^{-2} \leq 1$. For constraints at each node of circuit graph, the vector ϕ is typically sparse, as this vectors consist of entries corresponding to a few parameters, affecting only a single gate delay. As a result, the vectors $\boldsymbol{\eta_1}$ and η_2, derived, respectively, from the positive and negative terms of the elements of $P^{1/2}\phi$ are also sparse. However, during the graph pruning method, as the intermediate nodes are removed, the number of d_{ji} terms increase in every constraint. Thus, the sparsity of ϕ vector, and consequently, the sparsity of $\boldsymbol{\eta_1}$ and $\boldsymbol{\eta_2}$ is adversely affected. Moreover, as these vectors become dense, the number of monomial terms in the quadratic expansion of the constraints $\psi^2 {\boldsymbol{\eta_1}}^T \boldsymbol{\eta_1} r_1^{-2}$, and $\psi^2 \boldsymbol{\eta_2}^{\mathbf{T}} \boldsymbol{\eta_2} r_2^{-2}$ grow rapidly. As a result many constraints have monomial terms involving a large number of variables. Consequently, the constraint Jacobian matrix becomes very dense, which can considerably slow down the gradient computations required by the convex optimization methods, such as the interior point algorithm.

To overcome this issue of potential slow down of the gate sizing procedure, due to the increase in density of the constraint Jacobian matrix, we modify the pruning cost to include a penalty term related to increasing the number of terms in the η_1 and η_2 vectors. We define $Mono_{num}$ as the maximum number of monomial terms in all the constraints affected by removing the node under consideration. The cost of pruning this node is then calculated as:

$$f_{cost} \quad = \quad a\Delta_{con} + b\Delta_{var} + c\max(Mono_{num} - Mono_{spec}, 0) \tag{40}$$

where c is a weight factor, and $Mono_{spec}$ is a user specified quantity to represent the maximum number of monomial terms allowed in each constraint. A higher value of $Mono_{spec}$ could result in more pruning, but at the cost of a potential slow down in obtaining the solution of the GP optimization problem. Thus, by adjusting the $Mono_{spec}$ parameter, the user can choose an engineering tradeoff between the runtime and the amount of pessimism reduction desired in the gate sizing procedure.

In the next section, we elaborate on another heuristic method to further reduce the pessimism in our formulation.

E. Using Variable Size Ellipsoids

The graph pruning procedure of [19], explained in Section IV-D, helps in eliminating many intermediate arrival time variables, and reduce the problem of variation overestimation in our formulation. However, as described in the previous section, it may not be possible to remove all intermediate nodes from the graph, and leave only the ones corresponding to the primary inputs and the primary outputs unpruned. The number of fanins and fanouts of a node increase monotonically during the pruning procedure. Therefore, for a given pruning cost of Equation (40), if a node is unsuitable for pruning in any iteration of the pruning method, i.e., it has a positive pruning cost, it will never be pruned under the same criterion. Due to the presence of the unpruned nodes in the circuit graph, the pessimism in our optimization formulation is not completely eradicated.

We present another method, to be employed after the graph pruning procedure, to further reduce the excessive margins from the timing constraints formulated at the unpruned nodes of the graph. This method is based on setting variable margins at different topological levels of the circuit. We use a simple example circuit consisting of just two inverters to explain this method.

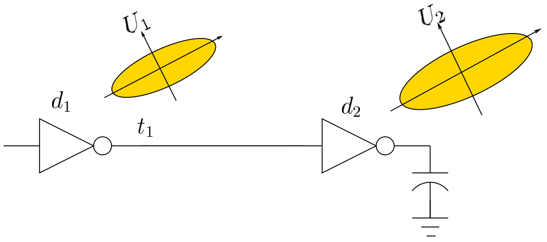

Fig. 7. An example circuit to explain the use of variable size ellipsoids to reduce the pessimism in the robust GP formulation.

Consider the circuit of Figure 7 consisting of two inverter gates. For this simple circuit, the intermediate node, corresponding to the output pin of the first inverter, can be easily removed to formulate the path delay constraint. However, for the purposes of exposition of the method of using variable ellipsoids, we do not employ any pruning and formulate the constraints for this circuit as:

$$\begin{aligned} d_1(\mathbf{X_0}) + \max_{\forall \delta\boldsymbol{\Omega} \in U_1} (\nabla_{\boldsymbol{\Omega}_\mathbf{0}} d_1(\mathbf{X_0}, \boldsymbol{\Omega})\delta\boldsymbol{\Omega}) &\leq t_1 \quad (41) \\ t_1 + d_2(\mathbf{X_0}) + \max_{\forall \delta\boldsymbol{\Omega} \in U_2} (\nabla_{\boldsymbol{\Omega}_\mathbf{0}} d_2(\mathbf{X_0}, \boldsymbol{\Omega})\delta\boldsymbol{\Omega}) &\leq T_{spec} \quad (42) \end{aligned}$$

We use different guard-bands for the constraints (41) and (42), by employing two uncertainty ellipsoids, U_1 and U_2 given by:

$$U_1 = \{\boldsymbol{\Omega} : (\boldsymbol{\Omega} - \boldsymbol{\Omega}_\mathbf{0})^T P^{-1} (\boldsymbol{\Omega} - \boldsymbol{\Omega}_\mathbf{0}) \leq \psi_1^2\} \quad (43)$$

$$U_2 = \{\boldsymbol{\Omega} : (\boldsymbol{\Omega} - \boldsymbol{\Omega}_\mathbf{0})^T P^{-1} (\boldsymbol{\Omega} - \boldsymbol{\Omega}_\mathbf{0}) \leq \psi_2^2\} \quad (44)$$

where $\psi_1 < \psi_2$. As explained in Section III-D, we can use the CDF tables of the χ_n^2 distribution to associate probability values, α_1 and α_2 with the ellipsoids U_1 and U_2, respectively. As $\psi_1 < \psi_2$, it follows that $\alpha_1 < \alpha_2$.

A simple probabilistic analysis to achieve the timing yield of the circuit of Figure 7, provides insights into the idea of using variable ellipsoids. Using the bounded ellipsoid model for parameter variations, we first define two random variables β_1 and β_2 as:

$$\begin{aligned} \beta_1 &= \max_{\forall \delta\boldsymbol{\Omega} \in U_1} (\nabla_{\boldsymbol{\Omega}_\mathbf{0}} d_1(\mathbf{X}, \boldsymbol{\Omega})\delta\boldsymbol{\Omega}) - (\forall_{\delta\boldsymbol{\Omega}} d_1(\mathbf{X}, \boldsymbol{\Omega})) \quad (45) \\ \beta_2 &= \max_{\forall \delta\boldsymbol{\Omega} \in U_2} (\nabla_{\boldsymbol{\Omega}_\mathbf{0}} d_2(\mathbf{X}, \boldsymbol{\Omega})\delta\boldsymbol{\Omega}) - (\forall_{\delta\boldsymbol{\Omega}} d_2(\mathbf{X}, \boldsymbol{\Omega})) \quad (46) \end{aligned}$$

The random variables defined in Equations (45) and (46), relate to the values α_1 and α_2 as :

$$\begin{aligned} \alpha_1 &\leq Pr(\beta_1 > 0) \quad (47) \\ \alpha_2 &\leq Pr(\beta_2 > 0) \quad (48) \end{aligned}$$

By using a smaller ellipsoid U_1 to guard-band the timing constraint of (41), we associate a smaller probability α_1, as a lower bound on the chance that this small design margin would be sufficient to meet the constraint in the face of variations. However, even if the design margin is not sufficient to meet this constraint, corresponding to the case that $\beta_1 < 0$, by employing a larger ellipsoid U_2, and the corresponding bigger probability α_2, to pad the timing constraint of (42), we have a better chance to compensate for the violation of constraint (41). Mathematically, if A is the probabilistic event that constraint (41) is not met, and B is the event that the circuit fails to meet the specified delay, the following relation holds[6]:

$$\begin{aligned} Pr(B/A) = {} & Pr(\beta_1 < 0) Pr(\beta_2 > 0/\beta_1 < 0) Pr((|\beta_1| > |\beta_2|)/\beta_2 > 0, \\ & \beta_1 < 0) + Pr(\beta_2 < 0) Pr(\beta_1 < 0/\beta_2 < 0) \quad (49) \end{aligned}$$

The use of a larger ellipsoid U_2 with an associated lower bound probability $\alpha_2 \leq Pr(\beta_2 > 0)$, ensures that for the cases when $\beta_1 < 0$, the term $Pr(\beta_2 < 0)$ and the conditional probability term $Pr((|\beta_1| > |\beta_2|)/\beta_2 > 0, \beta_1 < 0)$ in Equation (49) are reasonably small. Therefore, the scheme of using a smaller design margin for a lower topological level, followed by a sufficiently large design margin for higher levels can still provide the necessary guard-banding to achieve the desired timing yield.

For a general circuit with k topological levels, we employ k uncertainty ellipsoids, $U_1, U_2, \cdots, U_k$, characterized by the constants, $\psi_1, \psi_2, \cdots, \psi_k$, with $\psi_1 < \psi_2 < \cdots < \psi_k$. Since it is extremely difficult to relate the individual ellipsoid sizes with the timing yield specification, we heuristically chose ψ_k to correspond to the lower bound on the specified timing yield α_k, and progressively decrease the constants $\psi_{k-1}, \cdots, \psi_1$. The value of ψ_k is determined from the tables of the χ_n^2 distribution. The margins at logic levels, $1, \cdots, k-1$, are determined by setting:

$$\alpha_i = \alpha_k - \gamma.(k - i) \quad i = 1, \cdots, k-1 \quad (50)$$

where γ is an empirically determined factor. Using smaller timing margins at lower topological levels, as compared to choosing the same margin at all levels, corresponding to the lower bound on timing yield α_k, helps in reducing the pessimism in our formulation.

It should be noted that this scheme of using variable sized ellipsoids is employed for the unpruned nodes, only after the graph pruning step. The graph pruning method of [19], followed by the heuristic scheme of keeping variable guard-bands at different topological levels of the final pruned circuit, significantly reduces the problem of overestimation of variation in our gate sizing procedure.

F. Incorporating Spatial Correlations

We use the grid based spatial correlation model of [18] and [17] to incorporate the intra-die correlations between the parameters variations that exhibit spatial dependence, such as the transistor w and L_e.

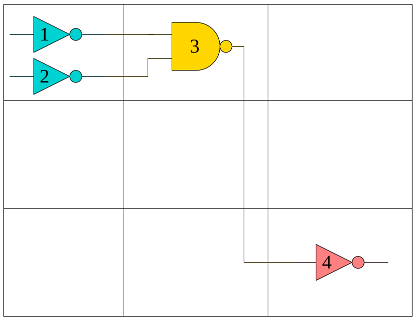

Fig. 8. A grid based spatial correlation model. The layout is divided into a 3×3 grid. The gates in the same grid are assumed to have a perfect correlation. Gates in the nearby grids are assigned a high correlation factor, and the gates in far away grids are assigned a low or a zero correlation factor.

Figure 8 refers to such a model, where the layout area is partitioned into $m = 9$ grids. The widths (channel lengths) of the devices located in the same grid are assigned a perfect correlation factor, device widths (channel lengths) in nearby grids are assigned a high correlation factor, and the ones in far away grids have a low or zero correlation factor. As seen in Figure 8, gates {1,2} have perfect correlation between their widths (channel lengths), gates {1,3} and {2,3} have high correlations, where as gates {1,4} and {2,4} are uncorrelated.

[6] Since the parameters of the two inverters may be correlated, Equation (49) contains terms corresponding to conditional probabilities.

For a random vector $\mathbf{\Omega}$ representing the variations in w and L_e, and its corresponding covariance matrix P, the entry $P_{ij} = \sigma_i \sigma_j \rho_{ij}$ denotes the covariance between components i and j of $\mathbf{\Omega}$, where σ is the standard deviation of each random variable, and ρ_{ij} is the correlation factor between the random variables i and j. By employing the spatial correlation model of Figure 8, the correlation factor between all elements of $\mathbf{\Omega}$ is computed, and stamped out in matrix P. The ellipsoid uncertainty model, described in Section III-C, then incorporates the impact of correlations in the robust optimization formulation.

The following simple example explains how the correlations are captured by the uncertainty ellipsoid. Consider a simple constraint involving the transistor widths of two gates:

$$t_j + \frac{K_1 w_1}{w_2} \quad \leq \quad t_i \tag{51}$$

For simplicity, we assume that the gate widths, w_1 and w_2, are the only two varying parameters, and the other parameters are subsumed in the constant K_1. Furthermore, we assume that the gates are placed in the same grid of the spatial correlation model, hence, the variations in the two gate widths are same, i.e., $\delta w_1 = \delta w_2$. If the nominal gate sizes are also assumed to be identical, i.e., $w_{1_0} = w_{2_0}$, the effect of process variation cancels out in the numerator and denominator of (51), and no guard-banding is required. To verify that the ellipsoid uncertainty correctly incorporates this perfect correlation scenario, we apply our robust optimization procedure to the constraint of (51). Generating a first order Taylor series expansion of the constraint around the nominal values (w_{1_0}, w_{2_0}), and applying the ellipsoid uncertainty yields:

$$\begin{aligned} & t_j + \frac{K_1 w_{1_0}}{w_{2_0}} + \\ \max_{\forall \mathbf{u} | \|\mathbf{u}\|_2 \leq \psi} & \left(\frac{K_1 (P^{1/2}\mathbf{u})_1}{w_{2_0}} - \frac{K_1 w_{1_0} (P^{1/2}\mathbf{u})_2}{w_{2_0}^2} \right) \quad \leq t_i \end{aligned} \tag{52}$$

However, since we have perfect correlation between w_1 and w_2, the correlation factor, $\rho_{12} = \rho_{21} = 1$. Therefore, the correlation matrix P is given by:

$$P \quad = \quad \begin{bmatrix} \sigma_1^2 & \sigma_1 \sigma_2 \\ \sigma_1 \sigma_2 & \sigma_2^2 \end{bmatrix}$$

Furthermore, since the variations in w_1 and w_2, and the mean values are same, we must have $\sigma_1 = \sigma_2$. It then follows that for all vectors $\mathbf{u} = [u_1, u_2]$, which characterize the uncertainty ellipsoid, we have $(P^{1/2}u)_1 = \sigma_1^2 u_1 + \sigma_1 \sigma_2 u_2 = (P^{1/2}u)_2 = \sigma_2^2 u_2 + \sigma_1 \sigma_2 u_1$, and the variational term in (51) is:

$$\frac{K_1 (P^{1/2}\mathbf{u})_1}{w_{2_0}} - \frac{K_1 w_{1_0} (P^{1/2}\mathbf{u})_2}{w_{2_0}^2} \quad = \quad 0$$

Thus, the ellipsoid uncertainty model easily captures the effects of correlations between random variables, and incorporates the same in the optimization procedure. Incorporating the correlations in gate sizing optimization procedure reduces the pessimism involved with a worst-casing scheme, and provides opportunities for saving expensive design resources.

G. The Complete Sizing Procedure

The complete gate sizing procedure can be recapitulated by the following steps:

1) Generate the initial non-robust timing constraints by an STA procedure.
2) On the original circuit graph, employ the graph pruning method of [19], described in Section IV-D, to remove as many intermediate nodes as possible according to the pruning cost function of Equation (40).
3) For the final pruned graph, generate new timing constraints using the edge annotations in the final pruned graph.
4) Generate a first order Taylor series expression for each constraint at the nominal values of the parameters.
5) Employing the uncertainty ellipsoid model, transform each constraint to a set of robust constraints as described in Section IV-B. For this step, use variable size ellipsoids at each topological level of the circuit, as explained in Section IV-E.
6) Solve the resulting GP by using convex optimization tools.

The solution of the convex optimization problem provides the gate sizes for the circuit that minimize the area objective, subject to the specified timing yield constraints.

V. Experimental Results

The proposed robust gate sizing procedure was implemented in C++, and an optimization software [21] was used to solve the final GP. All experiments were performed on P-4 Linux machines with a clock speed of 3.2GHz, and 2GB of memory. The robust gate sizing technique was applied to the ISCAS 85 benchmark circuits. The cell library selected comprised inverters, and two and three input NAND and NOR gates. We assume capacitive loading for the gates. For simplicity we consider the variations in the transistor width, and the effective channel length as the only sources of variation. However, our approach can be easily extended to incorporate various other parameters of variation for the gate and interconnect delays. We use a simple Elmore delay model to generate posynomial gate delay models. Our approach can work just as well for any other posynomial based delay models, such as the ones based on generalized posynomials proposed in [14].

We use the spatial correlation model of [18] and [17] to generate the elements of the covariance matrix P. To use these spatial correlation models, we first place the circuits using the placement tool Capo [22], and then divide the chip area into different number of grids, depending on the circuit size, so that each grid size is no greater than 50 μ × 50 μ. The standard deviations of the w and L_e parameters are chosen from [23] for a 100 nm technology node. Using this spatial correlation model, all the elements of the covariance matrix P are obtained to be nonnegative, which simplifies the implementation of the robust constraint generation process. However, the formulation, as described in Section IV-B, does not impose any sign restrictions for the elements of the P matrix. The objective function chosen for the optimization is to minimize $Area = \sum_i a_i w_{i_0}$, where a_i is the number of transistors in gate i. For each circuit, the value of T_{spec} is chosen to be the point of 15% slack, i.e., $T_{spec} = D_{min} + 0.15(D_{max} - D_{min})$, where D_{min} and D_{max} are, respectively, the minimum and the maximum possible delays of the circuit, found by setting all gates to the minimum and the maximum size, respectively.

We implement the graph pruning technique of [19] to address the problem of overestimation of variation. As described in Section IV-D.2, we set the pruning cost of a node as $f_{cost} = a\Delta_{con} + b\Delta_{var} + c\max(Mono_{num} - Mono_{spec}, 0)$. For this cost function, we choose $a = 1.5$, $b = 1$, $c = 1$. We choose different values for the term $Mono_{spec}$, that determines the maximum number of monomial terms allowed in each constraint. As described in Section IV-E, we employ smaller sized uncertainty ellipsoids at lower topological levels of the circuit, and progressively increase the ellipsoid size at higher logic levels. The size of the largest ellipsoid employed at the highest logic level k, characterized by ψ_k, is chosen to correspond to the lower bound on the timing yield specification, α_k. The value of ψ_k is determined from the tables of the χ^2_n distribution. The margins at logic levels, $1, \cdots, k-1$, are determined by using Equation(50) and choosing the factor γ to be in the interval of $[0.05, 0.10]$, which corresponds to a 5%-10% decrement from the value of α_k, that specifies the lower bound on the timing yield. The value of each ψ_i, corresponding to the α_i in Equation (50), is determined from the CDF tables of the Chi-square distribution.

In the first set of experiments, we compare the gate sizing solution obtained by our method with a deterministic gate sizing solution. The deterministic gate sizing is also formulated as a GP, using the formulation of Section 4, but it does not take into account the effect of parameter variations. For our robust optimization procedure, we set the lower bound on timing yield, $\alpha_k = 85\%$, and choose the value of $Mono_{spec} = 35$. To simulate the effect of parameter variations, we perform Monte Carlo analysis. We refer to the set of gate sizes obtained from the deterministic, and the robust optimization as $\mathbf{X_{0_{det}}}$ and $\mathbf{X_{0_{rob}}}$, respectively. Using these sizes, we generate $10,000$ samples each, from two multivariate normal distributions, $N_1(\mathbf{X_{0_{det}}}, P)$ and $N_2(\mathbf{X_{0_{rob}}}, P)$. Next, we perform an STA for each of these samples, and record the number of times the circuit meets the specified target delay. The timing yield of the two optimizations are then determined as $Yld_{det} = n_{det} \times 100/M$, and $Yld_{rob} = n_{rob} \times 100/M$, where n_{det} is the number of samples drawn from the $N_1(\mathbf{X_{0_{det}}}, P)$ distribution that meet the timing requirements, and n_{rob} is the number of samples drawn from the $N_2(\mathbf{X_{0_{rob}}}, P)$ distribution that meet the specified target delay. The total number of Monte Carlo samples is given by $M = 10000$. Table I contains the relevant data for this comparison.

Ckt	Gates	Deterministic Design			Robust Design		
		Ar	$Yld_{det}\%$	Time (sec)	Ar	$Yld_{rob}\%$	Time (sec)
C432	616	1.00	22.31%	3	1.12	99.91%	15
C499	1262	1.00	30.34%	2	1.18	99.94%	23
C880	854	1.00	28.46%	8	1.10	99.92%	18
C1355	1202	1.00	32.34%	12	1.15	98.89%	31
C1908	1636	1.00	35.14%	18	1.14	99.56%	159
C2670	2072	1.00	39.91%	30	1.17	99.83%	189
C3540	2882	1.00	33.31%	25	1.08	98.82%	212
C5315	4514	1.00	38.46%	43	1.12	98.76%	579
C6288	5548	1.00	37.45%	58	1.14	99.22%	742
C7552	6524	1.00	34.78%	90	1.17	99.13%	845

TABLE I

A TIMING YIELD COMPARISON OF DETERMINISTIC AND ROBUST GATE SIZING SOLUTIONS.

The first column in Table I lists the benchmark circuit, and the number of gates in each circuit is shown in column two. The timing yield of the deterministically sized circuits, $Yield_{det}$, is listed in column four of the table. Since the non-robust gate sizing method does not take into account the effect of variations, the timing yield, as expected, is quite low for these circuits. Our robust sizing method, eliminates these timing violations by keeping adequate design margins. Column seven list the timing yield, $Yield_{rob}$, of the robustly sized circuits. It should be noted that a value of $\alpha_k = 85\%$, as a lower bound on the timing yield, is sufficient to provide an actual yield of about 99% for all benchmark circuits. The area overhead that the robust circuits have to employ to safeguard against the parameter variations is shown in sixth column of Table I. At the cost of an area increase of about 8% to 18%, the robustly sized circuits are able to eliminate almost all timing violations. The runtimes of the deterministically, and robustly sized circuits are listed, respectively, in columns five and eight of the table. As seen in the table, the robust methods is much slower than the deterministic sizing procedure. The steps of employing graph pruning, and the increased problem size of the robust gate sizing procedure due to the presence of robust variables and constraints lead to this relatively higher runtimes. However, the overall runtimes of the gate sizing method are very reasonable.

We perform another series of experiments to compare our approach with a gate sizing methodology employing a conventional worst-case design approach. The worst-case designs are obtained by iteratively solving the standard GP, but for delay specifications tighter than the original required target delay, until the area of the worst-case design is the same as that of the robust design. These circuits are thus designed using an in-built guard-band, determined by the difference of the original target delay and the tighter delay specification. Furthermore, to explore the area-robustness tradeoff we vary the size of the largest uncertainty ellipsoid used, by choosing different values of the factor α_k, that determines the lower bound on the timing yield of the robustly sized circuits. For these experiments, as before, we set the values of $Mono_{spec} = 35$, to define the pruning cost function of Equation (40). Having sized these circuits, we perform Monte Carlo simulations to determine the timing yield of the worst-case and the robust circuits.

Table II lists the results of these experiments. As seen from the table, the number of timing violations reduces with increase in area, for both the worst-case and the robust circuits. However, in all cases, our robust design has a better timing yield than the worst-case design having the same area. On an average, the robust design has about 12% greater timing yield than the worst-case design having the same area. The better performance of our robust sizing solution is not surprising because of the fact that the spatial correlation information, stored in the P matrix, is used by the optimization scheme. The worst-case circuit is expected to have a large overhead, since designing by setting tighter delay specifications results in rendering critical some of the earlier non-critical paths. Therefore, the optimizer now has to aggressively size the gates on these paths, which results in greater transistor area than actually required. Since, the runtimes for our robust gate sizing solutions are not prohibitively high, the user can run the optimization for different values of α_k, to select the amount of robustness required against the process uncertainties, at the cost of additional chip area.

Ckt	Timing Yield for the Same Area Worst-Case (WC) and Robust (Rob) Designs											
	$\alpha_k = 0.55$			$\alpha_k = 0.65$			$\alpha_k = 0.75$			$\alpha_k = 0.85$		
	WC	Rob	Ar	WC	Rob	Ar	WC	Rob	Ar	WC	Rob	Ar
C432	45.63%	68.65%	1.05	86.78%	97.03%	1.08	91.62%	98.14%	1.10	93.12%	99.91%	1.12
C499	51.45%	63.45%	1.08	67.12%	74.28%	1.11	85.12%	97.01%	1.14	94.20%	99.94%	1.18
C880	52.36%	67.52%	1.03	77.38%	88.50%	1.06	88.42%	97.34%	1.08	92.38%	99.92%	1.10
C1355	55.78%	75.21%	1.08	66.17%	84.89%	1.11	82.66%	98.11%	1.13	91.43%	98.89%	1.15
C1908	50.67%	72.76%	1.06	70.69%	87.14%	1.10	84.53%	96.67%	1.12	93.89%	99.56%	1.14
C2670	56.32%	73.68%	1.08	72.86%	88.21%	1.11	89.23%	95.33%	1.14	92.34%	99.83%	1.17
C3540	60.22%	78.14%	1.02	76.15%	89.12%	1.04	89.32%	95.56%	1.06	94.14%	98.82%	1.08
C5315	55.81%	74.98%	1.05	75.50%	87.67%	1.08	90.56%	96.89%	1.10	93.45%	98.76%	1.12
C6288	55.39%	77.16%	1.07	69.79%	88.12%	1.10	85.78%	95.78%	1.12	91.91%	99.22%	1.14
C7552	49.08%	70.48%	1.08	66.21%	85.56%	1.12	83.89%	94.54%	1.15	90.11%	99.13%	1.17

TABLE II

A COMPARISON OF THE ROBUST AND WORST CASE GATE SIZING DESIGNS USING THE SAME AREA.

Ckt	Gates	Deterministic Design			Rob_1 Design			Rob_2 Design		
		Ar	Yld_{det}%	Time (sec)	Ar	Yld_{rob_1}%	Time (sec)	Ar	Yld_{rob_2}%	Time (sec)
C432	616	1.00	22.31%	3	1.08	97.03%	15	1.15	98.32%	14
C499	1262	1.00	30.34%	2	1.11	74.28%	23	1.17	76.78%	21
C880	854	1.00	28.46%	8	1.06	88.50%	18	1.14	90.23%	16
C1355	1202	1.00	32.34%	12	1.11	84.89%	31	1.20	85.34%	27
C1908	1636	1.00	35.14%	18	1.10	87.14%	159	1.22	89.12%	123
C2670	2072	1.00	39.91%	30	1.11	88.21%	189	1.24	89.03%	158
C3540	2882	1.00	33.31%	25	1.04	89.12%	212	1.17	90.32%	181
C5315	4514	1.00	38.46%	43	1.08	87.67%	579	1.23	89.32%	398
C6288	5548	1.00	37.45%	58	1.10	88.12%	742	1.24	90.45%	587
C7552	6524	1.00	34.78%	90	1.12	85.56%	845	1.27	87.29%	693

TABLE III

A COMPARISON OF ROBUST GATE SIZING SOLUTIONS, WITH AND WITHOUT USING GRAPH PRUNING AND VARIABLE SIZE ELLIPSOIDS.

In the next set of experiments, we investigate the usefulness of the graph pruning method, and employing different sized ellipsoids, in reducing the pessimism in our robust formulation. We first employ graph pruning, and use variable sized ellipsoids to optimize the benchmark circuits. At the highest topological circuit level, we use the largest ellipsoid corresponding to a value of $\alpha_k = 0.65$. At the lower topological levels, we progressive decrease the ellipsoid size by choosing a lower α, as given by Equation (50). We use a value of $Mono_{spec} = 35$ to set the pruning cost according to Equation (40). These circuits are referred to as Rob_1 designs. Next, we optimize the benchmark circuits without any pruning, and using the same sized ellipsoids at all nodes, determined by the values of $\alpha_k = 0.65$. These optimized circuits are referred to as Rob_2 designs.

Table III contains the results of these experiments. The yields of the two designs, Yld_{rob_1} and Yld_{rob_2}, are listed, respectively, in columns seven and ten of the table. The area employed by the Rob_1 and Rob_2 designs are shown, respectively, in columns six and nine of the table. As seen from this data in Table III, the designs employing the heuristic techniques of graph pruning, and using variable size ellipsoids use about 7% to 15% lesser circuit area compared to the design without any pruning, and using a constant size ellipsoid. The timing yields of Rob_2 designs are only slightly better, $< 2\%$ for all circuits, compared to the timing yields of Rob_1 design. This indicates that employing the graph pruning method, and the strategy of keeping variable guard-bands for the timing constraints, leads to considerable pessimism reduction in our optimization formulation, without a significant loss in the timing yield of the circuit. The runtimes for the Rob_2 designs are smaller compared to Rob_1 designs. This is due to the fact the robust constraints of (27) and (28) have fewer monomial terms for the procedure not employing any pruning compared to the one that prunes some intermediate nodes. As a result, the constraint functions are sparser for the former method, which helps in speeding up the optimization. The absence of the graph pruning step also makes the procedure for Rob_2 design run faster.

In the last set of experiments, we explore the tradeoff obtained by tuning the pruning cost function by changing the value of the $Mono_{spec}$ term, which regulates the maximum number of monomials allowed in a constraint. This term in the pruning cost of Equation (40) helps in preventing the constraint Jacobian matrix from becoming immoderately dense. Table IV contains the results of these experiments. As seen in the table, as the value of $Mono_{spec}$ term is increases, the runtime of the procedure increases. For the larger benchmark circuits, the slow down of the optimizer is significant, e.g., for C6288 circuit, the runtime increases by almost 40% by increasing the value of the $Mono_{spec}$ term from 20 to 50. This is due to the fact that for larger circuits, with thousands of constraints, the sparsity of the large constraint matrix has a greater impact on the speed of the convex optimization tool. Although, the runtime of the robust optimization method increases, for higher values of $Mono_{spec}$ term, there is also a greater reduction of pessimism in the formulation, due to more aggressive pruning. This results in lesser use of the circuit area for a higher valuer of $Mono_{spec}$ term. For example, for C6288 circuit, there is a 5% reduction in area by increasing the value of $Mono_{spec}$ from 20 to 50. The timing yield is not significantly impacted by changing the value of the $Mono_{spec}$ term. Based on this runtime and reduction in circuit area tradeoff, the user can appropriately set the value of $Mono_{spec}$ term to be employed in the pruning cost function of Equation (40).

Ckt	$Mono_{spec} = 20$			$Mono_{spec} = 35$			$Mono_{spec} = 50$		
	Ar	Yld	Time (sec)	Ar	Yld	Time (sec)	Ar	Yld	Time (sec)
C432	1.09	97.58%	15	1.08	97.03%	15	1.07	96.89%	17
C499	1.11	74.89%	22	1.11	74.28%	23	1.10	74.10%	25
C880	1.07	88.91%	18	1.06	88.50%	18	1.05	87.78%	20
C1355	1.12	85.12%	29	1.11	84.89%	31	1.10	83.67%	33
C1908	1.10	87.89%	147	1.10	87.14%	159	1.09	86.57%	172
C2670	1.13	88.95%	176	1.11	88.21%	189	1.10	87.34%	231
C3540	1.06	90.05%	200	1.04	89.12%	212	1.04	88.78%	294
C5315	1.09	88.34%	504	1.08	87.67%	579	1.07	86.89%	681
C6288	1.13	89.57%	657	1.10	88.12%	742	1.08	87.34%	920
C7552	1.14	86.78%	784	1.12	85.56%	845	1.10	84.12%	1027

TABLE IV

A COMPARISON OF THE ROBUST GATE SIZING DESIGNS OBTAINED BY CHANGING THE PRUNING COST FUNCTION OF EQUATION (40).

VI. CONCLUSION

In this paper were present a gate sizing procedure as a worst-casing methodology that attempts to keep smart design margins to safeguard against the effect of variations. Assuming a multivariate normal distribution for the process-driven parameter variations, an uncertainty ellipsoid set is employed as a bounded model for these variations. This uncertainty ellipsoid, defined by the appropriate covariance matrix of the varying parameters, incorporates the effect of spatial correlations in the optimization set up. The multivariate Gaussian assumption for parameter distributions allows the use of Chi-square CDF tables to specify a lower bound on the timing yield of the circuit. Using posynomial delay models, the optimization formulation for the gate sizing procedure is relaxed to a geometric program, that is solved using convex optimization tools. To reduce the pessimism associated with the node-based formulation, we employ the techniques of graph pruning and heuristically choosing variable sized ellipsoids at different topological levels of the circuit. Experimental results show that for the same transistor area, the circuits sized by of our robust optimization approach have, have fewer timing violations as compared to the gate sizing solutions obtained via the traditional, deterministically based guard-banding method.

REFERENCES

[1] J. Fishburn and A. Dunlop. TILOS: A Posynomial Programming Approach to Transistor Sizing. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 326–328, 1985.

[2] 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 of Integrated Circuits and Systems*, 12:1621–1634, Nov 1993.

[3] X. Bai, C. Visweswariah, P. N. Strenski, and D. J. Hathaway. Uncertainty-Aware Circuit Optimization. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 58–63, 2002.

[4] E. T. A. F. Jacobs and M. R. C. M. Berkelaar. Gate Sizing Using a Statistical Delay Model. In *Proceedings of IEEE Design Automation and Test in Europe*, pages 283–291, 2000.

[5] S. H. Choi, B. C. Paul, and K. Roy. Novel Sizing Algorithm for Yield Improvement under Process Variation in Nanometer Technology. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 454–459, 2004.

[6] D. Sinha, N. V. Shenoy, and H. Zhou. Statistical Gate Sizing for Timing Yield Optimization. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 1037–1042, 2005.

[7] A. Agarwal, K. Chopra, D. Blaauw, and V. Zolotov. Circuit Optimization Using Statistical Static Timing Analysis. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 338–342, 2005.

[8] K. Chopra, S. Shah, A. Srivastava, D. Blaauw, and D. Sylvester. Parametric Yield Maximization using Gate Sizing based on Efficient Statistical Power and Delay Gradient Computation. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 1023–1028, 2005.

[9] S. Raj, S. B. K. Vrudhala, and J. Wang. A Methodology to Improve Timing Yield in the Presence of Process Variations. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 448–453, 2004.

[10] M. Mani, A. Devgan, and M. Orshansky. An Efficient Algorithm for Statistical Power under Timing Yield Constraints. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 309–314, 2005.

[11] J. Singh, V. Nookala, T. Luo, and S. Sapatnekar. Robust Gate Sizing by Geometric programming. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 315–320, 2005.

[12] A. Davoodi and A. Srivastava. Variability Driven Gate Sizing for Binning Yield Optimization. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 959–964, 2006.

[13] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[14] K. Kasamsetty, M. Ketkar, and S. S. Sapatnekar. A New Class of Convex Functions for Delay Modeling and their Application to the Transistor Sizing Problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19:779–788, Jul 1998.

[15] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, Upper Saddle River, NJ, 2002.

[16] Y. Xu, K. L. Hsiung, X. Li, I. Nausieda, S. Boyd, and L. Pileggi. OPERA: Optimization with Ellipsoidal Uncertainty for Robust Analog IC Design. In *Proceedings of ACM/IEEE Design Automation Conference*, pages 632–637, 2005.

[17] H. Chang and S. S. Sapatnekar. Statistical Tming Analysis Considering Spatial Correlations Using a Single PERT-like Traversal. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 621–625, 2003.

[18] A. Agarwal, D. Blaauw, and V. Zoltov. Statistical Timing Analysis for Intra-Die Process Variations with Spatial Correlations. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 900–907, 2003.

[19] C. Visweswariah and A. R. Conn. Formulation of Static Circuit Optimization with Reduced Size, Degeneracy and Redundancy by Timing Graph Manipulation. In *Proceedings of IEEE/ACM International Conference on Computer Aided Design*, pages 244–252, 1999.

[20] M. H. Degroot and M. J. Schervish. *Probability and Statistics*. Addison Wesley, Boston, MA, 2002.

[21] MOSEK Optimization Software. Available at `http://www.mosek.com`.

[22] A. Caldwell, A. B. Kahng, and I. Markov. Capo: a large-scale fixed-die placer. available at `http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/Placement`.

[23] S. Nassif. Delay Variability: Sources, Impact and Trends. In *Proceedings of IEEE International Solid State Circuit Conference*, pages 368–369, 2000.