

# Timing Budgeting under Arbitrary Process Variations\*

Ruiming Chen and Hai Zhou

NuCAD, Electrical Engineering and Computer Science  
Northwestern University

## Abstract

Timing budgeting under process variations is an important step in a statistical optimization flow. We propose a novel formulation of the problem where budgets are statistical instead of deterministic as in existing works. This new formulation considers the changes of both the means and variances of delays, and thus can reduce the timing violation introduced by ignoring the changes of variances. We transform the problem to a linear programming problem using a robust optimization technique. Our approach can be used in late-stage design where the detailed distribution information is known, and is most useful in early-stage design since our approach does not assume specific underlying distributions. In addition, with the help of block-level timing budgeting, our approach can reduce the timing pessimism. Our approach is applied to the leakage power minimization problem. The results demonstrate that our approach can reduce timing violation from 690ps to 0ps, and the worst total leakage power by 17.50% on average.

## 1 Introduction

Process variation has become a critical problem in modern VLSI fabrication. The traditional corner-based analysis and optimization techniques become prohibitive. Statistical static timing analysis (SSTA) techniques as [1, 2] propagate distributions instead of single values, and achieve good efficiency. Based on them, many statistical optimization techniques [3–8] also emerged. We are considering the following statistical optimization problem. Given the constraint on the maximal delay from primary inputs to primary outputs in a combinational circuit, minimize the worst total cost (e.g., leakage power) such that the timing yield is at least a given value. Fig. 1 shows a general flow of statistical circuit optimization. The key sub-problem is the timing budgeting, i.e., redistribute the timing slacks to elements. Fig. 2 shows an example of timing budgeting.

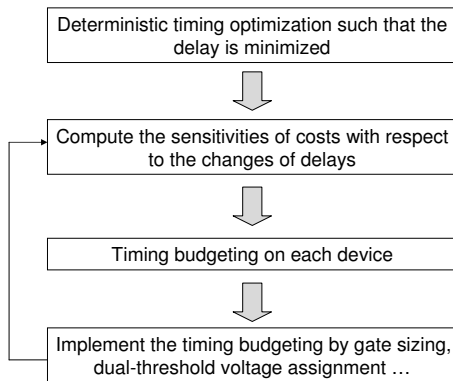


Figure 1: A general flow of statistical optimization.

There are already several works [6, 9–11] solving the timing budgeting problem. Most of them do not consider process

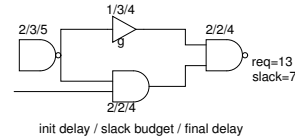


Figure 2: Timing budgeting is the redistribution of slacks.

variations. Mani *et al.* [6] transformed the timing budgeting problem under uncertainty to a second order conic programming (SOCP) problem. All these works put budgets on the mean of the delay, but when a gate is sized up or down, both the mean and the variance of the delay change. Without considering the change of the variance, the final result may suffer a big timing violation. For example, as shown in Fig. 2, the gate  $g$  needs to be sized down. Suppose the delays before sizing and after sizing are  $d_g$  and  $d'_g$  respectively, and the budget on the mean of the delay is  $b$ . After the gate is sized down, the delay budget is fully utilized, so the worst delay of the gate is expected to satisfy

$$\mu(d'_g) + 3\sigma(d'_g) = (\mu(d_g) + b) + 3\sigma(d_g).$$

But the variance may increase, then the actual worst delay of the gate

$$\mu(d'_g) + 3\sigma(d'_g) > (\mu(d_g) + b) + 3\sigma(d_g),$$

which means the actual worst delay is greater than the expected worst delay. So the timing constraint is violated. Thus, considering only the change of the mean in the budgeting step cannot guarantee that the timing constraint is satisfied after sizing.

Most of existing works optimize the cost at a late stage: they assume that the distributions of variables are known. An early-stage design can improve design greatly since it has more flexibility. But the detailed information of the distributions is not known at the early stage. How to explore limited information (e.g. the mean and the bounds of the distributions) in an early-stage design is the focus of this paper. There are already some works done on this topic [12–15], but all of them are on the analysis or modeling, not the optimization.

The contribution of our work includes:

- Our new formulation considers the budgeting of both the mean and the variance of the delay, so the timing violation is greatly reduced.
- The underlying distribution is allowed to be any symmetric distribution. So our technique can be used at both an early stage and a late stage.
- The timing budgeting problem is transformed to a linear programming problem through a pessimism controllable robust optimization technique, and is efficiently solved.
- Our block-level timing budgeting reduces the timing pessimism compared with gate-level timing budgeting.

\*This work was supported by NSF under CCR-0238484.

The rest of this paper is organized as follows. Section 2 briefly discusses how timing budgeting should be conducted. Section 3 presents our theory to handle timing budgeting under uncertainty. Section 4 proposes the application of our theory in gate sizing, and also proposes the block-level timing budgeting technique to reduce timing pessimism. Section 5 shows the experimental results. Finally, the conclusions are drawn in Section 6.

## 2 Timing Budgeting for Optimization

The general timing budgeting problem without uncertainty is

$$\text{Maximize} \quad \sum_{(i,j) \in E} c_{ij} b_{ij} \quad (1)$$

$$\text{s.t.} \quad t_j - t_i \geq d_{ij} + b_{ij} \quad \forall (i,j) \in E \quad (2)$$

$$0 \leq b_{ij} \leq \delta \quad \forall (i,j) \in E \quad (3)$$

where  $d_{ij}$  is the existing delay on edge  $(i,j)$ ,  $b_{ij}$  is the delay budget assigned on edge  $(i,j)$ ,  $\delta$  is a small positive real number, and  $c_{ij}$  is the sensitivity of the cost to the change of the delay. For example, in the leakage power minimization problem,

$$c_{ij} = \frac{\Delta P}{\Delta D},$$

which represents the leakage power reduction for one unit of delay change. The power-delay curve is not linear in general, so in order to make the linear approximation valid, we need to restrict the range of the delay change within a small range, so we have  $b_{ij} \leq \delta$ .  $b_{ij} \geq 0$  implies the delays of gates cannot decrease, so the current solution is always a feasible solution in the next iteration. If  $b_{ij}$  can be negative, because of the discrete nature of gate delays, we may not find a feasible solution in the budgeting implementation step.

Suppose the maximal delay from primary inputs to primary outputs should be not greater than  $D_{constr}$ . We introduce two dummy nodes  $s$  and  $t$ :  $s$  is connected to all the primary inputs, and  $t$  is connected from all the primary outputs. We also introduce an edge from  $t$  to  $s$  with delay  $-D_{constr}$ . Then

$$t_s - t_t \geq -D_{constr}$$

is put into the formulation to enforce the timing constraint.

As shown in Fig. 1, after timing budgeting, we need to size up/down gates or assign different threshold voltages to gates to implement the computed budgets. This is just a local optimization problem, and thus can be efficiently solved.

When process variations are considered, sensitivities ( $c_{ij}$ ) and delays are no longer deterministic. Let  $\hat{D}$  represent an element (e.g., a constant or a function)  $D$  with uncertainty. The timing budgeting problem under uncertainty was formulated as:

$$\text{P1: Maximize} \quad \sum_{(i,j) \in E} \hat{c}_{ij} b_{ij} \quad (4)$$

$$\text{s.t.} \quad t_j - t_i \geq \hat{d}_{ij} + b_{ij} \quad \forall (i,j) \in E \quad (5)$$

$$0 \leq b_{ij} \leq \delta \quad \forall (i,j) \in E \quad (6)$$

Note that budgets ( $b_{ij}$ ) are deterministic here.

The computation of  $\hat{c}_{ij}$  is as follows. For each gate in the circuit, we replace it by another gate of the same type from the library, and compute the statistical cost difference  $\Delta \hat{P}$  of these two gates, and also the difference  $\Delta D$  between the mean delays of these two gates. Then we compute

$$\frac{\Delta \hat{P}}{\Delta D},$$

and finally select the one with maximal mean value as the  $\hat{c}_{ij}$ . Note that the delay difference is deterministic. The reason is

that the  $\hat{c}_{ij}$  is the cost sensitivity to the *deterministic* shift ( $b_{ij}$ ) of the mean value.

## 3 Budgeting by Robust Optimization

The formulation P1 does not consider the change of variance when the mean of the delay is shifted by  $b_{ij}$ , so as mentioned before, it may introduce the violation of timing constraints.

We propose the following general formulation that can handle the changes of the variances of delays.

$$\text{Maximize} \quad \sum_{(i,j) \in E} \hat{f}_{ij}(b_{ij}) \quad (7)$$

$$\text{s.t.} \quad t_j - t_i \geq \hat{d}_{ij} + \hat{g}_{ij}(b_{ij}) \quad \forall (i,j) \in E \quad (8)$$

$$0 \leq b_{ij} \leq \delta \quad \forall (i,j) \in E \quad (9)$$

where functions  $\hat{f}_{ij}(b_{ij})$  and  $\hat{g}_{ij}(b_{ij})$  give the reduction of cost and the change of the edge delay when the mean of the delay on edge  $(i,j)$  is shifted by  $b_{ij}$ , respectively.

We assume that  $\hat{f}_{ij}(b_{ij})$  is a linear function of  $b_{ij}$ , and

$$\hat{f}_{ij}(b_{ij}) = \hat{c}_{ij} b_{ij}.$$

For the function  $\hat{g}_{ij}$ , [4] shows that the sensitivity of the delay of a gate to the variance of a parameter can be approximated as the product of a constant and the mean value of the delay, i.e., suppose the delay is represented in the linear form:

$$\hat{d}_{ij} = d_{ij}^{(0)} + \sum_{1 \leq k \leq n} d_{ij}^{(k)} \hat{\Delta}_k,$$

where  $d_{ij}^{(0)}$  is the mean of  $d_{ij}$ ,  $\hat{\Delta}_k$  is the variance of the  $k$ th parameter, and  $d_{ij}^{(k)}$  is the sensitivity of  $\hat{d}_{ij}$  to  $\hat{\Delta}_k$ , then

$$d_{ij}^{(k)} = \alpha_{ij}^{(k)} d_{ij}^{(0)},$$

where  $\alpha_{ij}^{(k)}$  is a constant when  $d_{ij}^{(0)}$  changes. Based on this, we get

$$\hat{g}_{ij}(b_{ij}) = \left[ 1 + \sum_k (\alpha_{ij}^{(k)} \hat{\Delta}_k) \right] b_{ij}.$$

Then Eq.(8) becomes

$$t_j - t_i \geq (d_{ij}^{(0)} + b_{ij}) + \sum_{1 \leq k \leq n} \left[ \alpha_{ij}^{(k)} (d_{ij}^{(0)} + b_{ij}) \right] \hat{\Delta}_k,$$

from which we can see that the variance also changes when the budget  $b_{ij}$  on the mean value changes. Thus, the new formulation captures the change of the variance when the mean of the delay shifts.

In summary, the formulation is

$$\text{P2: Maximize} \quad \sum_{(i,j) \in E} \hat{c}_{ij} b_{ij}$$

s.t.  $\forall (i,j) \in E$ :

$$t_j - t_i \geq (d_{ij}^{(0)} + b_{ij}) + \sum_{1 \leq k \leq n} \left[ \alpha_{ij}^{(k)} (d_{ij}^{(0)} + b_{ij}) \right] \hat{\Delta}_k \quad (10)$$

$$0 \leq b_{ij} \leq \delta \quad (11)$$

Robust optimization is a generic technique to handle this kind of mathematical programming problem under uncertainty. Bertsimas and Sim [16] proposed a robust optimization technique to transform a linear programming problem under uncertainty to a linear programming problem without uncertainty. We use this technique to transform our timing budgeting problem to a linear programming problem.

Without loss of generality, we assume that  $\hat{\Delta}_k \in [-1, 1]$ . In reality, it is unlikely that all of the parameters will change simultaneously. A real number  $\Gamma_{ij} \in [0, n]$  is used to control the robustness of the constraint Eq.(10). If only  $\lfloor \Gamma_{ij} \rfloor$  of parameters are allowed to change, and one parameter changes by at most  $(\Gamma_{ij} - \lfloor \Gamma_{ij} \rfloor)$ , the following robust optimization technique guarantees the feasibility of the solution in the problem P2.

$$\begin{aligned} \text{P3: Maximize } & \sum \hat{c}_{ij} b_{ij} \\ & t_j - t_i \geq (d_{ij}^{(0)} + b_{ij}) + \\ & \quad \max_{\{S_{ij} \cup \{t\} | S_{ij} \subseteq \{1..n\}, |S_{ij}| = \lfloor \Gamma_{ij} \rfloor, t \in \{1..n\} \setminus S_{ij}\}} \\ & \quad \left\{ \sum_{k \in S_{ij}} y_k + (\Gamma_{ij} - \lfloor \Gamma_{ij} \rfloor) y_t \right\} \\ & y_k \geq |\alpha_{ij}^{(k)} (d_{ij}^{(0)} + b_{ij})| \\ & 0 \leq b_{ij} \leq \delta \end{aligned}$$

After transformation, the following formulation is equivalent to P3.

$$\begin{aligned} \text{Maximize } & \sum \hat{c}_{ij} b_{ij} \\ & t_j - t_i \geq d_{ij}^{(0)} + b_{ij} + \Gamma_{ij} z_{ij} + \sum_{1 \leq k \leq n} q_{ij}^{(k)} \\ & z_{ij} + q_{ij}^{(k)} \geq |\alpha_{ij}^{(k)} (d_{ij}^{(0)} + b_{ij})| \\ & z_{ij} \geq 0 \\ & q_{ij}^{(k)} \geq 0 \\ & 0 \leq b_{ij} \leq \delta \end{aligned}$$

In this formulation, all the constraints are linear. Similarly, we transform the objective function to a linear function without uncertainty. The final formulation is

$$\begin{aligned} \text{P4: Maximize } & Y \\ & Y \leq c_{ij}^{(0)} b_{ij} - \Gamma_0 z_0 - \sum_{1 \leq k \leq n} q_0^{(k)} \\ & t_j - t_i \geq d_{ij}^{(0)} + b_{ij} + \Gamma_{ij} z_{ij} + \sum_{1 \leq k \leq n} q_{ij}^{(k)} \\ & z_0 + q_0^{(k)} \geq \left| \sum_{(i,j) \in E} c_{ij}^{(k)} b_{ij} \right| \\ & z_{ij} + q_{ij}^{(k)} \geq |\alpha_{ij}^{(k)} (d_{ij}^{(0)} + b_{ij})| \\ & z_0 \geq 0 \\ & q_0^{(k)} \geq 0 \\ & z_{ij} \geq 0 \\ & q_{ij}^{(k)} \geq 0 \\ & 0 \leq b_{ij} \leq \delta \end{aligned}$$

which is a linear programming problem. Thus, we can efficiently solve this.

If the problem does not satisfy the constraint on the maximal number of parameters allowed to change, [16] proved that if the distributions of those variables under uncertainty are symmetric, the probability that the timing constraint on edge  $(i, j)$  is violated is upper-bounded by

$$B(n, \Gamma_{ij}),$$

where

$$B(n, \Gamma_{ij}) = \frac{1}{2^n} \left\{ (1 - \mu) \binom{n}{\lfloor v \rfloor} + \sum_{l=\lfloor v \rfloor+1}^n \binom{n}{l} \right\} \quad (12)$$

where  $v = \frac{\Gamma_{ij} + n}{2}$  and  $\mu = v - \lfloor v \rfloor$ . So if

$$\forall (i, j) \in E, B(n, \Gamma_{ij}) \leq 1 - \eta,$$

where  $\eta$  is the yield constraint on the path delay, the timing yield is satisfied. Thus, our objective is to select  $\Gamma_{ij}$  such that  $B(n, \Gamma_{ij}) \leq 1 - \eta$ . We can use a look-up-table (LUT) to store the values of  $B(n, \Gamma_{ij})$  for a set of  $\Gamma_{ij}$ s, then the determination of  $\Gamma_{ij}$  is straight forward. A good property of this bound is that it does not depend on the input data (e.g., the distribution of  $d_{ij}$ ), so we can use it for arbitrary symmetric distributions.

#### 4 Application to Gate Sizing

In order to verify the effectiveness of our budgeting technique, we apply it to the statistical gate sizing problem. The problem is described as follows.

**Problem 1** *Given a combinational circuit and a library of gates with different sizes, optimize the circuit such that the worst total cost (e.g., leakage power) is minimized while the timing yield is at least a given value.*

This problem can be formally formulated as:

$$\begin{aligned} \text{Minimize } & \sum_{(i,j) \in E} \hat{C}(\hat{d}_{ij}) \\ \text{s.t. } & Pr(\max_{path} \hat{D}_{path} \leq D_{constr}) \geq \eta \\ & \hat{D}_{path} = \sum_{(i,j) \in path} \hat{d}_{ij} \end{aligned}$$

where function  $\hat{C}(\hat{d}_{ij})$  gives the cost when the delay on edge  $(i, j)$  is  $\hat{d}_{ij}$ ,  $\hat{D}_{path}$  is the maximal delay of the *path*,  $D_{constr}$  is the delay constraint, and  $\eta$  is the timing yield.

The global timing constraint on all the paths is transformed to local timing constraint on each path:

$$\begin{aligned} \text{Minimize } & \sum_{(i,j) \in E} \hat{C}(\hat{d}_{ij}) \\ \text{s.t. } & Pr(\hat{D}_{path} \leq D_{constr}) \geq \eta \quad \forall path \text{ from PIs to POs} \\ & \hat{D}_{path} = \sum_{(i,j) \in path} \hat{d}_{ij} \end{aligned}$$

Note that mathematically it is not guaranteed that this transformation is conservative, but as shown in [17], it is conservative in practice because of high correlations between path delays.

Since the number of paths is exponential with respect to the number of nodes and edges, we need to transform the timing constraint on each path to more local constraints. Previous research [6] distributed the constraint to each gate. In that case, there is no difference between the worst-case approach and this gate level approach in the timing part since they both use  $\mu + \Phi^{-1}(\eta)\sigma$  as the delay of a gate. We propose the block-level timing budgeting that can reduce the pessimism on timing.

#### 4.1 Block-level timing budgeting

We partition the circuit into many blocks; enumerate all the sub-paths in each block; write one timing constraint for each sub-path. Fig. 3 shows an example circuit. The dotted block has four gates, the five sub-paths within the block are enumerated, and each sub-path has one timing constraint:

$$\begin{aligned} t_4 - t_1 &\geq d_{1,x} + d_{x,y} + d_{y,4} + b_{1,x} + b_{x,y} + b_{y,4} \\ t_4 - t_1 &\geq d_{1,x} + d_{x,z} + d_{z,4} + b_{1,x} + b_{x,y} + b_{z,4} \\ t_4 - t_2 &\geq d_{2,x} + d_{x,y} + d_{y,4} + b_{1,x} + b_{x,y} + b_{y,4} \\ t_4 - t_2 &\geq d_{2,x} + d_{x,z} + d_{z,4} + b_{1,x} + b_{x,y} + b_{z,4} \\ t_4 - t_3 &\geq d_{3,z} + d_{z,4} + b_{3,z} + b_{z,4} \end{aligned}$$

where  $d_{x,y}$  and  $b_{x,y}$  are the delay and the delay budget on edge  $(x, y)$ , respectively.

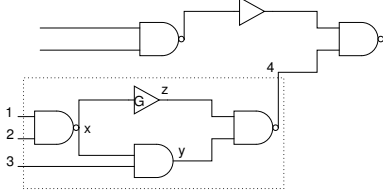


Figure 3: Block-level timing budgeting.

As shown in Fig. 4, two consecutive gates have delays  $1 + \hat{x}$  and  $1 - \hat{x}$ , respectively, where  $\hat{x}$  has a standard Gaussian distribution. The gate-level timing budgeting has the following two constraints:

$$t_j - t_i \geq 1 + \hat{x} + b_{ij},$$

and

$$t_k - t_j \geq 1 - \hat{x} + b_{jk}.$$

Thus the minimal delay from  $i$  to  $k$  is 8 ( $2(1 + 3\sigma) = 8$ ) through robust transformation. But the block-level budgeting combines these two gates together and thus has only one constraint:

$$t_k - t_i \geq 2 + b_{ij} + b_{jk}.$$

Thus the minimal delay from  $i$  to  $k$  is only 2. This example is just a demonstration: the sensitivities may have different signs for consecutive gates. When the sensitivities have the same signs, it can also be proved that combining gates together can reduce the timing pessimism.

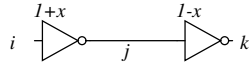


Figure 4: Block-level timing budgeting can reduce timing pessimism.

Let  $p(u \rightsquigarrow v)$  represent a path  $p$  from the input  $u$  to the output  $v$  of a block. In summary, the block-level timing budgeting problem is formulated as

$$\text{P5: Maximize } \sum_{(i,j) \in E} \hat{c}_{ij} b_{ij}$$

s.t.  $\forall (u \rightsquigarrow v) \in \{\text{paths in blocks}\} :$

$$\begin{aligned} t_v - t_u &\geq \sum_{(i,j) \in (u \rightsquigarrow v)} (d_{ij}^{(0)} + b_{ij}) + \\ &\sum_{1 \leq k \leq n} \left\{ \sum_{(i,j) \in (u \rightsquigarrow v)} [\alpha_{ij}^{(k)} (d_{ij}^{(0)} + b_{ij})] \right\} \hat{\Delta}_k \\ 0 &\leq b_{ij} \leq \delta \end{aligned}$$

Through the previously introduced robust optimization technique, P5 is converted to a linear programming problem:

$$\text{P6: Maximize } Y \quad (13)$$

$$\text{s.t. } Y \leq c_{ij}^{(0)} b_{ij} - \Gamma_0 z_0 - \sum_{1 \leq k \leq n} q_0^{(k)} \quad (14)$$

$$t_v - t_u \geq \sum_{(i,j) \in p(u \rightsquigarrow v)} (d_{ij}^{(0)} + b_{ij}) + \Gamma_p z_p + \sum_{1 \leq k \leq n} q_p^{(k)} \quad (15)$$

$$z_0 + q_0^{(k)} \geq \left| \sum_{(i,j) \in E} c_{ij}^{(k)} b_{ij} \right| \quad (16)$$

$$z_p + q_p^{(k)} \geq \left| \sum_{i,j \in p(u \rightsquigarrow v)} \alpha_{ij}^{(k)} (d_{ij}^{(0)} + b_{ij}) \right| \quad (17)$$

$$z_0 \geq 0 \quad (18)$$

$$q_0^{(k)} \geq 0 \quad (19)$$

$$z_p \geq 0 \quad (20)$$

$$q_p^{(k)} \geq 0 \quad (21)$$

$$0 \leq b_{ij} \leq \delta \quad (22)$$

Now we are considering two special cases.

#### 4.2 Deterministic budgets

When  $\hat{g}_{ij}(b_{ij}) = b_{ij}$ , the timing constraints in the timing budgeting problem can be converted to

$$t_v - t_u \geq \sum_{(i,j) \in p(u \rightsquigarrow v)} (d_{ij}^{(0)} + b_{ij}) + z_p \Gamma_p + \sum_{1 \leq k \leq n} q_p^{(k)}$$

$$z_p + q_p^{(k)} \geq \left| \sum_{(i,j) \in p(u \rightsquigarrow v)} d_{ij}^{(k)} \right|$$

$$z_p \geq 0$$

$$q_p^{(k)} \geq 0$$

$$0 \leq b_{ij} \leq \delta$$

Its KKT optimality conditions include the following conditions:

$$\sum_{\text{path } p \text{ from } u} x^{(p)} = \sum_{\text{path } q \text{ to } v} x^{(q)} \quad \forall \text{ I/O } u \text{ of blocks} \quad (23)$$

$$x^{(p)} \geq 0 \quad \forall \text{ path } p \quad (24)$$

$$\sum_k \beta_p^{(k)} \leq x^{(p)} \Gamma_p \quad \forall \text{ path } p \quad (25)$$

$$0 \leq \beta_p^{(k)} \leq x^{(p)} \quad \forall \text{ path } p \quad (26)$$

The  $x^{(p)}$  behaves like a network flow. Actually, it can be proved that the dual of the gate-level timing budgeting problem is a convex-cost flow problem.

The objective function of the dual problem has a term:

$$- \sum_p \left[ \sum_k \beta_p^{(k)} \left| \sum_{(i,j) \in p} d_{ij}^{(k)} \right| \right].$$

Without loss of generality, we assume that

$$\forall 1 \leq k \leq n-1 : \left| \sum_{(i,j) \in p} d_{ij}^{(k)} \right| \geq \left| \sum_{(i,j) \in p} d_{ij}^{(k+1)} \right|, \quad (27)$$

then for the optimal  $\beta_p^{(k)}$ ,

$$\beta_p^{(k)} = \begin{cases} x^{(p)} & \text{if } k \leq \Gamma_p \\ (\Gamma_p - \lfloor \Gamma_p \rfloor)x^{(p)} & \text{if } \Gamma_p \neq \lfloor \Gamma_p \rfloor \text{ and } k = \lfloor \Gamma_p \rfloor \\ 0 & \text{otherwise} \end{cases}$$

According to complementary slackness conditions, if  $x^{(p)} \neq 0$ , we know

$$z_p + q_p^{(k)} = \left| \sum_{(i,j) \in P} d_{ij}^{(k)} \right| \quad \text{if } k \leq \lfloor \Gamma_p \rfloor \quad (28)$$

$$q_p^{(k)} = 0 \quad \text{if } k > \lfloor \Gamma_p \rfloor \quad (29)$$

$$q_p^{(k)} = 0 \quad \text{if } \Gamma_p \neq \lfloor \Gamma_p \rfloor, k = \lfloor \Gamma_p \rfloor \quad (30)$$

Thus,

$$z_p \Gamma_p + \sum_k q_p^{(k)} = \left[ \sum_{k \leq \lfloor \Gamma_p \rfloor} \left| \sum_{(i,j) \in P} d_{ij}^{(k)} \right| \right] + z_p (\Gamma_p - \lfloor \Gamma_p \rfloor)$$

In order to make  $b_{ij}$  have more space to change, we need to reduce  $z_p \Gamma_p + \sum_k q_p^{(k)}$  as much as possible, so we need to find a minimal  $z_p$ . If  $\lfloor \Gamma_p \rfloor = n$ ,  $z_p = 0$ . Otherwise, if  $\Gamma_p \neq \lfloor \Gamma_p \rfloor$ , according to Eq.(28) and Eq.(30),

$$z_p = \left| \sum_{(i,j) \in P} d_{ij}^{(\lfloor \Gamma_p \rfloor)} \right|$$

If  $\exists p : x^{(p)} = 0$ , there might exist more than one optimal solutions that have the same objective value. It is obvious that if

$$z_p \Gamma_p + \sum_k q_p^{(k)}$$

is minimal,  $b_{ij}$  has more space to change, so the objective value is at least not worse than the optimal value. So similarly, we have

$$z_p = \begin{cases} \left| \sum_{(i,j) \in P} d_{ij}^{(\lfloor \Gamma_p \rfloor)} \right| & \text{if } \lfloor \Gamma_p \rfloor < n \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

$$q_p^{(k)} = \begin{cases} \left| \sum_{(i,j) \in P} d_{ij}^{(k)} \right| - z_p & \text{if } k \leq \lfloor \Gamma_p \rfloor \\ 0 & \text{if } k > \lfloor \Gamma_p \rfloor \end{cases} \quad (32)$$

Based on these results, we can simplify the problem by eliminating those  $z_p$  and  $q_p^{(k)}$ .

If the budgets are statistical (i.e.,  $\hat{g}_{ij}(b_{ij}) \neq b_{ij}$ ), the right hand side of Eq.(17) depends on  $b_{ij}$ 's and  $\alpha_{ij}^{(k)}$ , and cannot be sorted as in Eq.(27), so the simplification cannot be used. We still need to directly solve the linear programming P6.

### 4.3 Known distribution

For a late-stage design, the distributions are already known. For example, the delays are often modeled to have Gaussian distributions.

If budgets are deterministic, and the delays have Gaussian distributions, our task is to find the minimal  $\Gamma_p$  such that

$$\sum_i z_p \Gamma_p + \sum_k q_p^{(k)} \geq \mu \left( \sum_{(i,j) \in P} \hat{d}_{ij} \right) + \Phi^{-1}(\eta) \sigma \left( \sum_{(i,j) \in P} \hat{d}_{ij} \right).$$

We have known  $z_p$  and  $q_p^{(k)}$  from Eq.(31) and Eq.(32), so  $\Gamma_p$  can be easily computed.

If budgets are not deterministic, and the delays have Gaussian distributions, the standard deviation of the delay is

$$\sqrt{\sum_k \left( \sum_{(i,j) \in P} \alpha_{ij}^{(k)} (d_{ij}^{(0)} + b_{ij}) \right)^2},$$

which involves the non-linear operation on variables  $b_{ij}$ 's. How to determine  $\Gamma_p$  becomes difficult. We use the original bound Eq.(12) that ignores the distribution information to determine  $\Gamma_p$ .

For the objective function, the sensitivities are often modeled to have Lognormal distributions. In reality, the budgeting computation step and the budgeting implementation step are not coupled very well because of the discreteness of gate delays. For example, the budgeting computation step assigns a budget  $3ps$  to a gate  $G$  with delay  $d_G$ , but we cannot find a gate in the library with exactly  $d_G + 3ps$  delay. Suppose we find a gate with delay  $d_G + 2ps$ , then actually  $1ps$  budget is not used, so there exist a discrepancy between the actual cost after the implementation step and the expected cost computed in the budgeting step. So the accurate model of the objective function may not benefit the final result greatly. Thus, we always set  $\Gamma_0 = n$  in P6.

## 5 Experimental results

Table 1: Comparison between statistical approaches that use deterministic budgets and statistical budgets on worst case delay

name	circuits		DET-BUDGET	ChiruTimer
	#gates	$D_{constr}$ (ns)	delay (ns)	delay (ns)
c1355	384	5.00	5.23	5.00
c1908	430	4.50	4.57	4.49
c2670	563	5.00	5.12	4.99
c3540	1075	8.00	8.18	8.00
c5315	1453	6.00	6.17	6.00
c6288	2913	17.00	17.69	17.00
c7552	1624	6.00	6.19	6.00

Our approach is implemented in C++ in our timer called ChiruTimer. We use MOSEK linear programming solver [18] as a sub-routine to solve the linear programming problem. The number of parameters is 10. We assume that parameters have at most 30% deviation from their means. The means and the bounds of the distributions of costs and delays are known.

Each block in the block-level budgeting has a gate and all its immediate fan-out gates. The yield constraint on timing is 100%. We tested our approach on ISCAS85 benchmarks that are synthesized to a UMC 90nm technology library. All the experiments were run on a Linux Redhat PC with 2.4GHz CPU and 2.0GB memory.

### 5.1 Timing yield

First, we test the effectiveness of statistical budgets (i.e.,  $\hat{g}_{ij}(b_{ij}) \neq b_{ij}$ ) on improving timing yield. Table 1 shows the comparison results between two statistical gate sizing approaches: the DET-BUDGET approach uses deterministic budgets (i.e.,  $\hat{g}_{ij}(b_{ij}) = b_{ij}$ ), and ChiruTimer uses statistical budgets. The ‘‘delay (ns)’’ columns show the worst maximal delay from primary inputs to primary outputs. The results indicate that all the solutions from DET-BUDGET violate the timing constraints. For example, the solution for c6288 has a timing violation of 690 ps. ChiruTimer does not have any timing violations.

Table 2: Comparison between worst case deterministic approach and ChiruTimer on worst case cost

circuits		Worst Deterministic		ChiruTimer		
name	#gates	leakage (uw)	time (s)	leakage (uw)	time (s)	savings (%)
c1355	384	138.91	14.10	128.17	57.85	7.73
c1908	430	131.66	16.90	114.36	57.61	13.14
c2670	563	90.86	26.88	69.27	71.78	23.76
c3540	1075	188.44	60.08	148.94	296.54	20.96
c5315	1453	220.73	88.57	164.22	326.19	25.60
c6288	2913	684.46	163.20	592.04	1512.54	13.50
c7552	1624	311.38	70.95	256.02	429.51	17.78
Avg						17.50

## 5.2 Cost improvement

Second, we test our approach on the cost improvement. Table 2 shows the comparison results between worst case deterministic approach and ChiruTimer. The worst-case deterministic approach uses worst-case delays and sensitivities. Its formulation of budgeting step is

$$\text{Maximize } \sum_{(i,j) \in E} (c_{ij}^{(0)} - \sum_k |c_{ij}^{(k)}|) b_{ij} \quad (33)$$

$$\text{s.t. } t_j - t_i \geq d_{ij}^{(0)} + \sum_k |d_{ij}^{(k)}| + (1 + \sum_k |\alpha_{ij}^{(k)}|) b_{ij} \quad (34)$$

$$0 \leq b_{ij} \leq \delta \quad (35)$$

It guarantees that this worst-case deterministic approach gets solutions without timing violation since the coefficients of  $b_{ij}$  in the constraints Eq.(34) are their worst values. The results indicate that ChiruTimer achieves 17.50% savings on the worst leakage power on average. These savings come from two parts: the first part is that the cost sensitivities are statistical, so the correlations between cost sensitivities are considered; the second part is the block-level timing budgeting, so the correlations between delays are partially considered. The blocks in ChiruTimer are small: a gate and its immediate fan-out gates form a block. With bigger blocks, more cost savings can be expected. For the running time, ChiruTimer finished most of these cases in several minutes.

## 6 Conclusions

We propose a novel formulation of the timing budgeting problem to consider the changes of both the means and the variances of delays, and transform the problem to a linear programming problem using a robust optimization technique.

Our approach can be used not only in late-stage design where the detailed distribution information is known, but also in early-stage design since our approach does not assume any specific underlying distributions. In addition, we propose a block-level timing budgeting to capture the correlations between devices and reduce timing pessimism. The experimental results demonstrate that our approach not only increases the timing yield, but also improves the leakage power by 17.50% on average.

## References

- [1] H. Chang and S. Sapatnekar. Statistical timing analysis considering spatial correlations using a single PERT-like traversal. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 621–625, San Jose, CA, November 2003.
- [2] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan. First-order incremental block-based statistical timing analysis. In *Proc. of the Design Automation Conf.*, pages 331–336, San Diego, June 2004.
- [3] 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 *Proc. Intl. Conf. on Computer-Aided Design*, pages 1023–1028, 2005.
- [4] M. Guthaus, N. Venkateswaran, C. Visweswariah, and Z. Zolotov. Gate sizing using incremental parameterized statistical timing analysis. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 1029–1036, 2005.
- [5] D. Sinha, N. Shenoy, and H. Zhou. Statistical gate sizing for timing yield optimization. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 1037–1041, San Jose, CA, 2005.
- [6] M. Mani, A. Devgan, and M. Orshansky. An efficient algorithm for statistical minimization of total power under timing yield constraints. In *Proc. of the Design Automation Conf.*, pages 309–314, Anaheim, CA, June 2005.
- [7] A. Davoodi and A. Srivastava. Variability driven gate sizing for binning yield optimization. In *Proc. of the Design Automation Conf.*, pages 959–964, San Francisco, CA, USA, July 2006.
- [8] R. Chen and H. Zhou. Fast min-cost buffer insertion under process variations. In *Proc. of the Design Automation Conf.*, pages 338–343, 2007.
- [9] V. Sundararajan, S. S. Sapatnekar, and K. K. Parhi. Fast and exact transistor sizing based on iterative relaxation. *IEEE Transactions on Computer Aided Design*, 21(5):568–581, 2002.
- [10] R. Chen and H. Zhou. Efficient algorithms for buffer insertion in general circuits based on network flow. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 322–326, 2005.
- [11] S. Ghiasi, E. Bozorgzadeh, P.-K. Huang, R. Jafari, and M. Sarrafzadeh. A unified theory of timing budget management. *IEEE Transactions on Computer Aided Design*, 25(11):2364–2375, November 2006.
- [12] J. D. Ma, C. F. Fang, R. A. Rutenbar, X. Xie, and D. S. Boning. Interval-valued statistical modeling of oxide chemical-mechanical polishing. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 141–148, San Jose, CA, 2005.
- [13] A. Singhee, C. F. Fang, J. D. Ma, and R. A. Rutenbar. Probabilistic interval-valued computation: toward a practical surrogate for statistics inside cad tools. In *Proc. of the Design Automation Conf.*, pages 167–172, San Francisco, CA, USA, July 2006.
- [14] W.-S. Wang, V. Kreinovich, and M. Orshansky. Statistical timing based on incomplete probabilistic descriptions of parameter uncertainty. In *Proc. of the Design Automation Conf.*, pages 161–166, San Francisco, CA, USA, July 2006.
- [15] W.-S. Wang and M. Orshansky. Robust estimation of parametric yield under limited descriptions of uncertainty. In *Proc. Intl. Conf. on Computer-Aided Design*, pages 884–890, San Jose, California, November 2006.
- [16] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming B*, 98:49–71, 2003.
- [17] M. Mani and M. Orshansky. A new statistical optimization algorithm for gate sizing. In *Proc. Intl. Conf. on Computer Design*, pages 272–277, San Jose, CA, October 2004.
- [18] MOSEK. Mosek aps optimization software. <http://www.mosek.com>.