

Risk Aversion Min-Period Retiming Under Process Variations*

Jia Wang

Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, IL, USA

Hai Zhou

Fudan University, China
Northwestern University, USA

Abstract—Recent advances in statistical timing analysis (SSTA) achieve great success in computing arrival times under variations by extending sum and maximum operations to random variables. It remains a challenge problem to apply such results in order to address the variability in circuit optimizations. In this paper, we study the statistical retiming problem, where retiming is a powerful sequential transformation that relocates flip-flops in a circuit without changing its functionality. We formulate the risk aversion min-period retiming problem under process variations based on conventional two-stage stochastic program with fixed recourse and a risk aversion objective of the clock period. We prove that the proposed problem is an integer convex program, show that the subgradient of the objective function can be derived from the combinational paths with the maximum path delay, and present a heuristic incremental algorithm to solve the proposed problem. Our approach can handle arbitrary gate delay model under process variations through sampling from a black-box and the effectiveness is confirmed by the experimental results. Further more, we point out how the current state-of-the-art SSTA techniques could be improved for future optimization algorithms when analytical models are available.

I. INTRODUCTION

With aggressive scaling down of VLSI feature sizes, process variations have become a critical issue in VLSI fabrication that the designers must face. Because of the increasing variations, chip characteristics, e.g. the clock period and the power consumption, fall into larger intervals instead of being single values or within narrow ranges. As statistical analysis approaches enable the designer to analysis the variations, statistical optimization algorithms will finally equip the designers with the necessary tools to control such stochastic effects in order to improve chip yield and system reliability.

A review of the recent advances in statistical timing analysis can be found in the paper [1]. In summary, state-of-the-art SSTA algorithms can achieve a good balance in terms of accuracy and efficiency to compute the arrival times and the clock period under variations by extending the sum and maximum operations to random variables. To advance SSTA in order to handle other aspects of the circuits and to apply SSTA for efficient statistical optimization remain the challenge problems.

Conventional circuit optimizations have been extended to address the issue of variability through statistical optimization. For example, for the gate sizing problems, numerous approaches have been proposed for statistical sizing optimizations. In [2], Lagrangian relaxation based sizing technique was extended through introducing a safety margin to the circuit timing according to path delay variations. In [3], [4], [5], sensitivity guided iterative improvement heuristics were developed. Convex formulations with provable global optimality were proposed in [6], [7] to optimize the circuit under the worst cases considering variations, and in [8] through conventional two-stage stochastic programs with fixed recourse [9] to optimize the binning yield. With those successes, it is natural to ask if such approaches can provide insights into future SSTA researches and can be extended to other deterministic optimizations.

Among the many deterministic optimization techniques, retiming [10] is one of the most powerful sequential transformations. Intuitively, one can apply retiming to improve the timing yield of a circuit under process variations since relocating the FFs could

balance the combinational paths. Such idea was previously explored by Wang and Zhou [11], where a heuristic algorithm was proposed by combining SSTA with a deterministic min-period retiming algorithm [12]. However, there is little theoretical guarantee that such heuristic would result in a good retiming solution. In this paper, we study the statistical retiming problem with a more sound theoretical basis and propose a new heuristic algorithm to optimize the circuit for better clock period distribution under process variations. Our contributions in this paper include:

- 1) We formulate the risk aversion min-period retiming problem for statistical retiming optimization. Our formulation is based on conventional two-stage stochastic programs with fixed recourse. A coherent measure of risk called conditional value-at-risk [13] is used as the objective function to be minimized. We prove that the proposed problem is an integer convex program by presenting a continuous convex relaxation of it.
- 2) We derive an analytical formula for the subgradient of the objective function based on the continuous relaxation. Compared to the previous works [4], [8] where the subgradients are computed through perturbing the circuits and evaluating the changes statistically for multiple times, our approach is much more efficient. We can use any random gate delay model through sampling from a black box model representing the underlying variation model. Moreover, we show how current SSTA techniques can be improved to further speed-up the subgradient computation and thus the statistical retiming optimizations.
- 3) We extend the concept of timing critical paths, which is essential for deterministic retiming optimizations [10], [12], [14], to a statistical sense. We propose a practical simplification of the concept such that it can be efficiently integrated into our algorithm for the risk aversion min-period retiming problem.
- 4) We propose the Incremental Risk Aversion Retiming algorithm to solve the risk aversion min-period retiming problem heuristically, guided by the subgradient and the statistical timing critical paths.

The rest of this paper is organized as follows. The retiming problems, the two-stage stochastic programs, and the coherent measure of risk are introduced in Section II. We formulate the risk aversion min-period retiming problem in Section III and show that it is an integer convex program in Section IV. We propose our Incremental Risk Aversion Retiming algorithm in Section V. After experimental results are given in Section VI, Section VII concludes the paper.

II. PRELIMINARIES

A. Deterministic Retiming Problems

For retiming, a synchronous sequential circuit is modeled by a directed graph $G = (V, E)$ as in Leiserson and Saxe [10]. The vertices V represent combinational gates and the edges E represent signals between vertices. The gate delays are given as the nonnegative vertex weights $d : V \rightarrow \mathbb{R}^*$. The numbers of FFs on the signals are given as the nonnegative edge weights $w : E \rightarrow \mathbb{N}$.

To guarantee that the circuit functionality will be preserved after FF relocation, a retiming is given by a vertex labeling $r : V \rightarrow \mathbb{Z}$,

*The research was conducted at Northwestern University and supported in part by NSF under CNS-0613967.

which represents the number of FFs moved backward over each gate from its fanouts to its fanins. The FF number on the edge (u, v) after retiming is $w_r(u, v) = w(u, v) + r(v) - r(u)$. The retiming r is *valid* iff the FF number of every edge remains nonnegative,

$$w_r(u, v) \geq 0, \forall (u, v) \in E. \quad (1)$$

For a given valid retiming r , the retimed circuit works under a given clock period ϕ iff the maximum combinational path delay in the circuit is at most ϕ . In such case, the retiming r is called *feasible* for ϕ . To compute the maximum path delay, arrival times $t : V \rightarrow \mathbb{R}$ are introduced at the outputs of all the gates. The following constraints should be satisfied.

$$\begin{aligned} w_r(u, v) = 0 &\Rightarrow t(v) \geq d(v) + t(u), \forall (u, v) \in E, \\ d(v) &\leq t(v) \leq \phi, \forall v \in V. \end{aligned} \quad (2)$$

Conventionally, the min-period retiming problem asks for a minimum clock period such that there exists a feasible retiming for it, and the min-area retiming problem asks for a feasible retiming for a given clock period to minimize the total FF area. To solve both problems, it would be helpful to investigate the timing critical paths, each of which is a directed path connecting two vertices satisfying that the total number of FFs along the path is the minimum among all the paths with the same endpoints, and the total delay along the path exceeds the desired clock period. For any pair of vertices, if there exists a timing critical path connecting them, a feasible retiming will require at least 1 FF along the path. However, because it is usually expensive and sometimes prohibitive to generate all such critical timing constraints, practically efficient algorithms [12], [14] are able to identify those critical timing constraints from Eq. (2) only when they are required and to organize them into proper data structures in order to guide the optimization and to assert optimality with a low storage overhead.

B. Two-Stage Stochastic Program with Fixed Recourse and Coherent Measure of Risk

A decision problem whose output depends not only on the decision itself but also some uncertain parameters not available at the time of decision making is usually formulated as a two-stage stochastic program with fixed recourse [9], [15], [16], [17]. In such programs, the uncertain parameters unknown at the time of decision are modeled as random variables. The program is separated into two stages as suggested by its name. In the first stage, a decision is made and will incur an initial cost. In the second stage, the uncertain parameters are realized and a second stage cost is determined from both the decision and the realized uncertain parameters through a known deterministic program, i.e. the fixed recourse. The objective of the program is to make a decision in the first stage to minimize the ‘‘total cost’’ of the two stages – as the outcome is random, such cost may have many possible interpretations.

Let X be the random variable representing the outcome of the two-stage stochastic program. An interpretation can be formalized by introducing a measure of risk $M[X]$ that maps X into a real number. Usually a random variable mapped to a smaller value is better than the ones mapped to larger values. For example, a family of the most popular measures are the following ones involving the mean and the standard deviations of the random variable X , $M_\gamma[X] \triangleq E[X] + \gamma\sqrt{E[(X - E[X])^2]}$. On the other hand, if X represents the random clock period of a circuit under process variations, given a target clock period ϕ , one can measure X by the timing yield, i.e., $\text{Yield}_\phi[X] \triangleq P(X \leq \phi)$. However, as pointed out by Rockafellar [13], the above measures are not favorable objectives for optimizations because they are not *coherent*. A coherent measure of risk should satisfy a few conditions as follows.

Definition 1 (Coherent Measure of Risk [13]): A measure of risk M is coherent in the basic sense if,

1. $M[C] = C$ for all constants C .
2. $M[(1 - \lambda)X + \lambda Y] \leq (1 - \lambda)M[X] + \lambda M[Y]$ for $\lambda \in [0, 1]$.
3. $M[X] \leq M[Y]$ if $P(X \leq Y) = 1$.
4. $M[X] \leq 0$ when there exists a infinite sequence of random variables X_k such that $\lim_{k \rightarrow +\infty} E[(X_k - X)^2] = 0$ and $M[X_k] \leq 0$.
5. $M[\lambda X] = \lambda M[X]$ for $\lambda > 0$.

Intuitively, the first condition indicates that if a deterministic value is treated as a random variable taking a single value, the measure should interpret it by the deterministic value; the second condition requires that the measure to be convex with respect to the random variables; the third condition ensures the measure to be monotonic, i.e., if one random variable is no smaller than the other with probability 1, the measure of the former should be no smaller than that of the latter; the fourth condition guarantees that if a random variable can be approximated by some other random variables, one will accept it when all the approximations are acceptable; and the fifth condition implies that the measure is insensitive to scaling.

Coherent measures of risk do exist. For example, the expectation $E[X]$ of the random variable X is a coherent measure of risk, though it is feeble and cannot capture the risk associated with X . A more interesting coherent measure of risk, as proposed by Rockafellar [13], is the *conditional value-at-risk* that measures the risk in a random variable beyond a risk aversion level α . For a risk aversion level α , a measure of risk $\text{VaR}_\alpha[X]$ called *value-at-risk* is first defined as the value satisfying that $P(X \leq \text{VaR}_\alpha[X]) = \alpha$. Intuitively, the value-at-risk measure can be treated as the inverse of the timing yield measure: while the timing yield measure computes the risk aversion level (the timing yield) from a given value (the target clock period), the value-at-risk measure computes the value at a given risk aversion level. The value-at-risk measure is not coherent, the conditional value-at-risk measure, which is coherent, is defined as follows based on value-at-risk: $\text{CVaR}_\alpha[X] \triangleq E[X | X > \text{VaR}_\alpha[X]]$. For any $x \in \mathbb{R}$, let x^+ denote $\max\{0, x\}$. One can prove that [13],

$$\text{CVaR}_\alpha[X] = \text{VaR}_\alpha[X] + \frac{1}{1 - \alpha} E[(X - \text{VaR}_\alpha[X])^+]. \quad (3)$$

Therefore, if the conditional value-at-risk should be minimized, it has the advantage to optimize both the value-at-risk and the tail beyond the value-at-risk.

III. PROBLEM FORMULATION

Under process variations, the delays of the gates in the circuit are no longer deterministic but random variables. Let Ω be the probabilistic space representing process variations. For a particular variation $\omega \in \Omega$, assume that the random gate delays are realized as the deterministic nonnegative vertex weights $d_\omega : V \rightarrow \mathbb{R}^*$. For a valid retiming r , let the minimum clock period for the retimed circuit under the variation ω be $\phi_\omega(r)$. According to Section II-B, we formulate the following *risk aversion min-period retiming* problem as a two-stage stochastic program with fixed recourse.

Problem 1 (Risk Aversion Min-Period Retiming): Given a risk aversion level of α , find an integer-valued vertex labeling r for the following program:

Minimize $\text{CVaR}_\alpha[\phi_\omega(r)]$ s.t. $w_r(u, v) \geq 0, \forall (u, v) \in E$,
where for every variation ω belonging to the probabilistic space Ω , $\phi_\omega(r)$ is the minimum objective of the following program,

Minimize ϕ s.t.

$$\begin{aligned} w_r(u, v) = 0 &\Rightarrow t(v) \geq d_\omega(v) + t(u), \forall (u, v) \in E, \\ d_\omega(v) &\leq t(v) \leq \phi, \forall v \in V. \end{aligned}$$

It is clear that in the first stage of the risk aversion min-period retiming problem, a valid retiming will be chosen with an initial cost of 0. In the second stage, when the random gate delays are realized as d_ω , the minimum clock period is computed through the fixed recourse

by solving the second stage program. The second stage cost is the coherent risk aversion measure of the minimum clock period.

IV. A CONVEX RELAXATION

Note that the proposed risk aversion min-period retiming problem is difficult, not only because r should be integer-valued, but also because the second stage program is not a mathematical program. To overcome such difficulty, we propose to relax the program before attempting to solve it.

A. Continuous Relaxation Formulation

Consider an arbitrary simple path p from u to v , i.e. a path without cycles, in the circuit graph G . Let the total number of FFs along the path be $w(p)$. Let the total path delay be $d_\omega(p)$ for a particular $\omega \in \Omega$. Then for a valid retiming r , the minimum clock period should satisfy that $d_\omega(p) \leq \phi_\omega(r)(w_r(p) + 1)$, where $w_r(p) = w(p) + r(v) - r(u)$ is the total number of FFs along the path p in the retimed circuit. Moreover, because r is valid, we should have that $w_r(p) \geq 0$. On the other hand, there must exist a combinational path p^* in the retimed circuit with the maximum combinational path delay. For such path p^* , it must satisfy that $w_r(p^*) = 0$ and $d_\omega(p^*) = \phi_\omega(r)$. Therefore, we have the following lemma that transforms the second stage program into a mathematical program by enumerating paths.

Lemma 1: For a valid retiming r , the minimum clock period for a particular $\omega \in \Omega$ can be computed as that,

$$\phi_\omega(r) = \max_{\text{simple path } p \text{ in } G} \frac{d_\omega(p)}{w_r(p) + 1}. \quad (4)$$

Note that although the second stage program as formulated in Lemma 1 is a mathematical program, its size is exponential in terms of the size of the circuit graph G , while the size of the second stage program as formulated in Problem 1 is linear. As the mathematical program formulation will be only applied to theoretical analysis, its size will not be a concern for practical implementations.

Based on Eq. (4), we can relax the requirement that r should be integer-valued by extending $\phi_\omega(r)$ to real-valued r . First, we define a real-valued r to be *valid* iff $w_r(u, v) \geq 0$ holds for every edge $(u, v) \in E$. Then for any simple path p from u to v in G , it remains true that $w_r(p) \geq 0$. Therefore, for any valid real-valued r , we can define $\phi_\omega(r)$ using the same equation as Eq. (4). In summary, we have the following continuous relaxation of the risk aversion min-period retiming problem.

Problem 2: Given a risk aversion level of α , find a real-valued vertex labeling r for the following program:

Minimize $\text{CVaR}_\alpha[\phi_\omega(r)]$ s.t. $w_r(u, v) \geq 0, \forall (u, v) \in E$, where for every variation ω belonging to the probabilistic space Ω ,

$$\phi_\omega(r) = \max_{\text{simple path } p \text{ in } G} \frac{d_\omega(p)}{w_r(p) + 1}.$$

B. Convexity of Formulation

A very important property of Problem 2 is that it is a convex program. The proof is as follows.

Let r' and r both be valid and real-valued. For a particular $\omega \in \Omega$, assume that for the simple path p_ω from u_ω to v_ω , we have that $\phi_\omega(r) = \frac{d_\omega(p_\omega)}{w_{r'}(p_\omega) + 1}$. Let the vertex labeling $s_\omega : V \rightarrow \{-1, 0, 1\}$ be that $s_\omega(u_\omega) = 1$, $s_\omega(v_\omega) = -1$, and $s_\omega(x) = 0$ for any other $x \in V$. On the other hand, we should have $\phi_\omega(r') \geq \frac{d_\omega(p_\omega)}{w_{r'}(p_\omega) + 1}$. Therefore,

$$\begin{aligned} \phi_\omega(r') - \phi_\omega(r) &\geq \frac{d_\omega(p_\omega)}{w_{r'}(p_\omega) + 1} - \frac{d_\omega(p_\omega)}{w_r(p_\omega) + 1} \\ &= \phi_\omega(r) \left(\frac{w_r(p_\omega) + 1}{w_{r'}(p_\omega) + 1} - 1 \right) = \phi_\omega(r) \frac{w_r(p_\omega) - w_{r'}(p_\omega)}{w_{r'}(p_\omega) + 1} \\ &= \phi_\omega(r) \left(\frac{w_r(p_\omega) - w_{r'}(p_\omega)}{w_r(p_\omega) + 1} + \frac{(w_r(p_\omega) - w_{r'}(p_\omega))^2}{w_{r'}(p_\omega) + 1} \right) \\ &\geq \phi_\omega(r) \frac{w_r(p_\omega) - w_{r'}(p_\omega)}{w_r(p_\omega) + 1} = \sum_{u \in V} \frac{\phi_\omega(r) s_\omega(u)}{w_r(p_\omega) + 1} (r'(u) - r(u)). \end{aligned} \quad (5)$$

Given a risk aversion level α , let $I_\omega(r)$ be 1 if $\phi_\omega(r) \geq \text{VaR}_\alpha[\phi_\omega(r)]$ and 0 otherwise. For ease of presentation, denote $\text{VaR}_\alpha[\phi_\omega(r)]$ by A and $\text{VaR}_\alpha[\phi_\omega(r')]$ by A' . Then, we have that,

$$\mathbb{E}[I_\omega(r)] = P(\phi_\omega(r) \geq A) = 1 - \alpha, \quad (6)$$

$$(\phi_\omega(r) - A)^+ = (\phi_\omega(r) - A)I_\omega(r), \quad (7)$$

$$(\phi_\omega(r') - A')^+ \geq (\phi_\omega(r') - A')I_\omega(r). \quad (8)$$

Thus,

$$\begin{aligned} &\text{CVaR}_\alpha[\phi_\omega(r')] - \text{CVaR}_\alpha[\phi_\omega(r)] \\ &= \left(A' + \frac{\mathbb{E}[(\phi_\omega(r') - A')^+]}{1 - \alpha} \right) - \left(A + \frac{\mathbb{E}[(\phi_\omega(r) - A)^+]}{1 - \alpha} \right) \\ &\geq (A' - A) + \frac{\mathbb{E}[(\phi_\omega(r') - A') - (\phi_\omega(r) - A)]I_\omega(r)}{1 - \alpha} \\ &= (A' - A) \left(1 - \frac{\mathbb{E}[I_\omega(r)]}{1 - \alpha} \right) + \frac{\mathbb{E}[(\phi_\omega(r') - \phi_\omega(r))I_\omega(r)]}{1 - \alpha} \\ &= \frac{\mathbb{E}[(\phi_\omega(r') - \phi_\omega(r))I_\omega(r)]}{1 - \alpha} \\ &\geq \frac{1}{1 - \alpha} \mathbb{E}[I_\omega(r)] \sum_{u \in V} \frac{\phi_\omega(r) s_\omega(u)}{w_r(p_\omega) + 1} (r'(u) - r(u)) \\ &= \sum_{u \in V} \frac{r'(u) - r(u)}{1 - \alpha} \mathbb{E}[I_\omega(r) \frac{\phi_\omega(r) s_\omega(u)}{w_r(p_\omega) + 1}] \end{aligned}$$

Therefore, the following lemma must hold.

Lemma 2: $\text{CVaR}_\alpha[\phi_\omega(r)]$ is a convex function of r for all real-valued valid r . For a particular valid r , define $g_r : V \rightarrow \mathbb{R}$ as that,

$$g_r(u) \triangleq \frac{1}{1 - \alpha} \mathbb{E}[I_\omega(r) \frac{\phi_\omega(r) s_\omega(u)}{w_r(p_\omega) + 1}]. \quad (9)$$

Then g_r is a subgradient of $\text{CVaR}_\alpha[\phi_\omega(r)]$.

As the set of all the valid real-valued r is convex, we have the following theorem according to Lemma 2.

Theorem 1: Problem 2 is a convex program. The optimal solution to the risk aversion min-period retiming problem is an integer optimal solution to Problem 2.

V. INCREMENTAL ALGORITHM FOR RISK AVERSION MIN-PERIOD RETIMING

We have shown in Section IV that the risk aversion min-period retiming is an integer convex program and derived a subgradient of the objective function. Intuitively, such subgradient can be used to guide iterative heuristic searches. In this section, we will first show the method to compute the subgradient in practice and then present a heuristic algorithm based on the idea of incremental retiming.

A. Computing Subgradient from Black Box Model

According to Lemma 2, the subgradient of the objective function can be computed as Eq. (9). Obviously, how to compute such subgradient in practice depends on how the probability space Ω and the random gate delays are specified. There are two typical models, where in the first model, the joint distribution of the gate delays is explicitly given, and in the second model, one can only obtain knowledge of the distribution by drawing independent samples from a black box. In this paper, we are interested in the latter black box model because the black box model is independent of the underlying distribution and thus our proposed algorithm can handle arbitrary variation models. Moreover, since the subgradient will be computed from each sample drawn from the black box model, established deterministic analysis frameworks can be reused. One may be concerned about the efficiency of the algorithms relying on the black box model because of the multiple samplings. However, since the subgradient is used to guide the optimization, absolute accuracy is not necessary and a limited number of samples will be suffice for effective optimizations. Note that in case of the former model where the distribution is explicitly given, it would be helpful if current SSTA techniques can be extended to compute the subgradient

according to Eq. (9) efficiently and accurately. Such extensions are out of the scope of this paper and are left as one of the future directions of SSTA.

Subroutine ComputeSubgrad	
Inputs	
G : the circuit graph. r : a valid retiming. α : the risk aversion level. N : the number of samples.	
Outputs	
An approximation \hat{g}_r of the subgradient g_r .	
1	Draw the samples ω_i for $i = 1, 2, \dots, N$.
2	For $i = 1$ to N :
3	Compute the minimum clock period ϕ_{ω_i} and the arrival times t by Eq. (2); use $q(v)$ to record the source of the critical path to the vertex v .
4	Identify the sink v_i of a critical path whose delay is ϕ_{ω_i} . The source of the path $u_i \leftarrow q(v_i)$.
5	$A \leftarrow$ the maximum value such that $ \{i : \phi_{\omega_i} < A\} \leq \alpha N$.
6	$\hat{g}_r(v) \leftarrow 0, \forall v \in V$.
7	For $i = 1$ to N :
8	If $\phi_{\omega_i} \geq A$:
9	$\hat{g}_r(u_i) \leftarrow \hat{g}_r(u_i) + \frac{\phi_{\omega_i}}{N(1-\alpha)}, \hat{g}_r(v_i) \leftarrow \hat{g}_r(v_i) - \frac{\phi_{\omega_i}}{N(1-\alpha)}$.

Fig. 1. The ComputeSubgrad subroutine.

Since the risk aversion min-period retiming problem requires an integer solution of Problem 2, we maintain an integer solution through our algorithm. Therefore, only the subgradients at integer solutions should be computed. Let r be a valid retiming. Eq. (9) suggests that the subgradient can be approximated by taking the average of the corresponding values from the individual samples. Suppose that N samples, $\omega_i, i = 1, 2, \dots, N$, are independently drawn from the black box. We design the ComputeSubgrad subroutine as shown in Fig. 1 to obtain an approximation \hat{g}_r of g_r by averaging the samples. In this algorithm, after the samples are drawn, we perform timing analysis on line 3 for each sample to determine the minimum clock period and the arrival times according to Eq. (2). As the same time, we maintain a vertex labeling $q(v)$ to record the source of the critical path to the vertex v . Then, the combinational path with the maximum path delay is identified implicitly on line 4 by its endpoints u_i and v_i . Note that the path delay should be ϕ_{ω_i} and there is no FF along the path in the retimed circuit. An approximation of $\text{VaR}_\alpha[\phi_\omega]$ is obtained on line 5. Finally, in the loop on line 7, we compute an approximation \hat{g}_r of the subgradient g_r according to Eq. (9).

Note that many previous statistical optimization works, e.g. [4], [8], employed a different approach to approximate such subgradient. For a decision variable, the previous approaches will first perturb the variable and then approximate the subgradient of this variable by the change of the objective function under such perturbation. Such approach incurs large runtime overhead because, first, although for some decision variables, perturbation will not change the objective function and thus they can be excluded from the above computation, the number of the decision variables that the above computation must be applied to will increase as the circuit size increases; second, evaluating the objective function usually requires expensive SSTA algorithms. On the other hand, our analytical formula for the subgradient, as in Eq. (9), allows us to compute the subgradient comparably efficiently via sampling from a black box model. Moreover, as mentioned before, the efficiency of our approach can be further improved with future relevant SSTA researches.

B. Statistical Timing Critical Paths

An intuitive idea for optimization is to iteratively improve a valid retiming r following the subgradient obtained via the ComputeSubgrad subroutine by solving the following problem for r' .

$$\text{Minimize } \sum_{v \in V} \hat{g}_r(v)(r'(v) - r(v)) \text{ s.t.} \quad (10)$$

$w_{r'}(u, v) \geq 0, \forall (u, v) \in E$, and $0 \leq r'(v) - r(v) \leq 1, \forall v \in V$. In this problem, the objective function is a first-order approximation of $\text{CVaR}_\alpha[\phi_\omega(r')]$. As this first-order approximation would become

inaccurate when r' is faraway from r , we require the difference between r' and r to be at most 1. Because the constraints are a system of difference inequalities, this problem can be solved by network-flow techniques and it is not necessary to round a non-integer solution to an integer one for a valid retiming since there always exists an integer-valued optimal solution.

However, this intuitive idea does not perform well in practice. The reason is that even changing $r(v)$ by 1 for some vertex v will result in huge changes in the minimum clock period. Cutting plane techniques, similar to the statistical gate sizing work [8], can be applied to form a more accurate approximation. However, such techniques no longer guarantee the existence of an integer-valued optimal solution and may require a heuristic to round a non-integer optimal solution. Therefore, they cannot be applied directly to our retiming problem.

We propose to overcome such difficulty by introducing the concept of *statistical timing critical paths*. These paths can be treated as a natural extension of the deterministic timing critical paths as mentioned in Section II-A to the statistical sense. Let r be the current valid retiming. Consider a simple path p in G . For any variation $\omega \in \Omega$, let $d_\omega(p)$ be the path delay. If $\text{CVaR}_\alpha[d_\omega(p)] > \text{CVaR}_\alpha[\phi_\omega(r)]$, then there is at least one FF along the path p in the retimed circuit; otherwise, we should have $\text{CVaR}_\alpha[d_\omega(p)] \leq \text{CVaR}_\alpha[\phi_\omega(r)]$ since $d_\omega(p) \leq \phi_\omega(r)$ holds for the combinational path p . Similarly, the same condition will hold for any valid retiming r' satisfying $\text{CVaR}_\alpha[\phi_\omega(r')] \leq \text{CVaR}_\alpha[\phi_\omega(r)]$, as stated in the following lemma.

Lemma 3: Given a constant C , we define a simple path p to be a statistical timing critical path if $\text{CVaR}_\alpha[d_\omega(p)] > C$. For any valid retiming r satisfying $\text{CVaR}_\alpha[\phi_\omega(r)] \leq C$, we must have that $w_r(p) \geq 1$.

Based on Lemma 3, we can augment the formulation in Eq. (10) by the following constraints without affecting the optimality.

$$w_{r'}(p) \geq 1, \forall p \text{ satisfying } \text{CVaR}_\alpha[d_\omega(p)] > \text{CVaR}_\alpha[\phi_\omega(r)]. \quad (11)$$

Note that the constraints in Eq. (11) have the same structure as those in Eq. (10), i.e., they are a system of difference inequalities. Therefore, the existence of an integer-valued optimal solution is still guaranteed.

C. Incremental Risk Aversion Retiming Algorithm

One difficulty of the constraints in Eq. (11) is that since the risk measure should be computed for many simple paths, it could be inefficient in practice. We propose to simplify the computation in our implementation by identifying similar paths through deterministic timing analysis. Let $\bar{d}(v) = \mathbb{E}[d_\omega(v)]$ be the nominal delay for each gate v . For a simple path p , let $\bar{d}(p)$ be the nominal path delay with respect to the nominal gate delays \bar{d} . For a given valid retiming r , let $\bar{\phi}(r)$ be the nominal minimum clock period, i.e., the minimum clock period with respect to \bar{d} . Then, we assume a simple path to be a statistical timing critical path if $\bar{d}(p) > \beta \bar{\phi}(r)$, where $\beta \geq 1$ is a parameter specified by the designer. In summary, given a valid retiming r , we propose to solve the following incremental retiming problem to obtain another valid retiming r' in order to improve the conditional value-at-risk measure of risk.

$$\text{Problem 3: Minimize } \sum_{v \in V} \hat{g}_r(v)(r'(v) - r(v)) \text{ s.t.} \\ w_{r'}(u, v) \geq 0, \forall (u, v) \in E, \text{ and } 0 \leq r'(v) - r(v) \leq 1, \forall v \in V, \\ w_{r'}(p) \geq 1, \forall \text{ simple path } p \text{ satisfying } \bar{d}(p) > \beta \bar{\phi}(r).$$

In Problem 3, since path enumeration is required to construct the constraints, the number of the constraints can be quadratic in terms of the number of the vertices, i.e. $\Theta(|V|^2)$. This may impose huge storage and runtime overhead if we are going to solve Problem 3 directly. However, we can treat Problem 3 as a special min-area retiming problem and apply a recently discovered incremental min-area retiming algorithm iMinArea [14] to solve it. The iMinArea

Subroutine IncreRetime	
Inputs	
G : the circuit graph. \bar{d} : the nominal gate delay. \bar{r} : a valid retiming. \hat{g}_r : the approximation of the subgradient. β : a designer specified parameter.	
Outputs	
The optimal solution r' of Problem 3.	
1	Compute $\bar{\phi}(r)$ as the nominal minimum clock period of r . $\phi \leftarrow \beta\bar{\phi}(r)$.
2	Initialize F to be a regular forest with no edge with respect to $-\hat{g}_r$.
3	Loop:
4	$I \leftarrow$ vertices of the positive trees in F .
5	If $I = \emptyset$: $r' \leftarrow r$. Return.
6	If $w_r(u, v) = 0$ for an edge (u, v) leaving I :
7	Update F with (u, v) . Continue the loop.
8	Construct a retiming r_I by moving 1 FF from the fanouts of I to their fanins in r .
9	Compute the arrival times t and the sources of the critical path q for each vertex v in r_I .
10	If $t(v) > \phi$ in r_I for some vertex v :
11	Update F with $(q(v), v)$. Continue the loop.
12	$r' \leftarrow r_I$. Return.

Fig. 2. The IncreRetime subroutine adapted from the iMinArea algorithm.

algorithm requires only $O(|V|)$ storage on top of the circuit graph G and is efficient in practice. Let $\hat{g}_r(v)$ represents the increase of FF area when 1 FF is moved from the fanouts of v to its fanins. It is straight-forward that the given valid retiming r is feasible for the clock period $\beta\bar{\phi}(r)$ with respect to the nominal gate delays. Then, Problem 3 actually asks for a set of vertices I such that the retiming r' , which is obtained by moving 1 FF from the fanouts of I to its fanins, is a feasible retiming for the clock period $\beta\bar{\phi}(r)$ with the minimum FF area. Because only 1 FF is allowed to move, it is not necessary to run the iMinArea algorithm until it finishes. We adapt the iMinArea algorithm in our IncreRetime subroutine as shown in Fig. 2 to solve Problem 3. The details of the iMinArea algorithm can be found in the work [14]. The following lemma states the correctness of the IncreRetime subroutine.

Lemma 4: The IncreRetime subroutine terminates and when it terminate, it returns an optimal solution of Problem 3.

Algorithm Incremental Risk Aversion Retiming	
Inputs	
G : the circuit graph. r : an initial valid retiming. α : the risk aversion level. R : the maximum number of iterations.	
Outputs	
The retiming r^* with the best $\text{CVaR}_\alpha[\phi_\omega(r^*)]$.	
1	$r^* \leftarrow r$.
2	For $k = 1$ to R :
3	Compute $\text{CVaR}_\alpha[\phi_\omega(r)]$ by sampling.
4	If $\text{CVaR}_\alpha[\phi_\omega(r)] < \text{CVaR}_\alpha[\phi_\omega(r^*)]$: $r^* \leftarrow r$.
5	Compute \hat{g}_r by ComputeSubgrad.
6	Solve Problem 3 for r' by IncreRetime.
7	If $r' = r$: Stop.
8	Else : $r \leftarrow r'$.

Fig. 3. The Incremental Risk Aversion Retiming algorithm.

Based on the above discussions, we design the *Incremental Risk Aversion Retiming* algorithm as shown in Fig. 3 to solve the risk aversion min-period retiming problem. In this algorithm, from a given initial valid retiming, we iteratively improve the current solution by first computing a subgradient on line 5 and then moving to the next solution on line 6 via solving Problem 3. The iteration will stop when a current retiming cannot be improved as found on line 7, or a maximum number R of iterations have been performed. The retiming solution with the best conditional value-at-risk measure of risk will be picked at the end of the algorithm.

VI. EXPERIMENTS

We obtain the code of the deterministic incremental min-period retiming algorithm [12] and build a risk-aware deterministic approach for comparison with our Incremental Risk Aversion Retiming algorithm. In this approach, we first assign each gate a deterministic delay derived from the gate delay distribution and a parameter γ specified

TABLE II
RESULTS IN TERMS OF TIMING YIELD.

name	Deterministic Approach				Ours
	init	$\gamma = 0$	$\gamma = 1$	$\gamma = 3$	
s27	88.3%	88.3%	90.0%	90.0%	90.0%
s208.1	62.9%	89.6%	89.6%	89.6%	90.0%
s298	48.5%	89.2%	89.2%	88.7%	90.0%
s382	9.6%	88.0%	88.0%	88.0%	90.0%
s386	80.6%	90.0%	90.0%	90.0%	90.0%
s344	17.6%	84.8%	87.8%	87.8%	90.0%
s349	17.7%	86.2%	88.6%	88.6%	90.0%
s400	10.9%	84.8%	84.8%	85.8%	90.0%
s420.1	13.1%	89.4%	89.4%	89.4%	90.0%
s444	5.8%	86.3%	87.5%	84.9%	90.0%
s510	86.8%	90.0%	89.5%	89.5%	90.0%
s526	17.6%	82.3%	83.9%	83.9%	90.0%
s641	89.1%	89.1%	89.1%	89.1%	90.0%
s713	89.9%	89.9%	89.9%	89.9%	90.0%
s820	88.7%	89.8%	89.8%	88.7%	90.0%
s832	86.8%	90.0%	90.0%	90.0%	90.0%
s838.1	5.7%	87.9%	89.3%	84.5%	90.0%
s953	66.6%	89.5%	90.0%	90.0%	90.0%
s1196	88.2%	89.8%	89.8%	89.8%	90.0%
s1238	90.0%	90.0%	90.0%	90.0%	90.0%
s1423	22.3%	88.4%	88.4%	88.4%	90.0%
s1488	63.0%	90.0%	90.0%	90.0%	90.0%
s1494	71.2%	90.0%	90.0%	90.0%	90.0%
s5378	82.5%	82.5%	82.5%	82.5%	90.0%
s9234.1	90.6%	90.6%	90.0%	90.0%	90.0%
s13207.1	82.8%	88.9%	88.9%	88.9%	90.0%
s15850.1	46.5%	87.2%	87.5%	87.5%	90.0%
s35932	74.2%	91.5%	89.5%	84.7%	90.0%
s38417	0.2%	83.1%	84.3%	82.2%	90.0%
s38584.1	85.3%	90.0%	90.0%	90.0%	90.0%

by the designer. Then we run Zhou’s algorithm [12] for a min-period retiming to obtain a solution. For a gate v , the deterministic gate delay is that $E[d_\omega(v)] + \gamma\sqrt{E[(d_\omega(v) - E[d_\omega(v)])^2]}$, i.e., a weighted summation of the nominal delay and the standard deviation. Note that this deterministic approach is similar to the “Alternative Algorithm” proposed in the work [11].

We implement our Incremental Risk Aversion Retiming algorithm in C++. All the codes are compiled by GCC version 3.4 and run on a Linux workstation with dual 927MHz Intel Pentium III processors and 512MB memory.

We derive our experimental benchmarks from the conventional ISCAS89 sequential circuits. To establish a gate delay model for process variations, we assume a joint Gaussian distribution of the gate delay. The parameters of the distribution are determined as follows. First, we assign each gate a nominal delay proportional to the number of its fanouts and a standard deviation that is within 20% to 30% of the nominal value. Then, assuming that each gate has a dimension of 1×1 , we perform a wire-length driven placement of the circuits using the placement tool mPL6 [18]. After placement, the chip area is divided into a 4×4 grid. Two gate delays are assumed to be perfectly correlated if they are within a same grid block, i.e., the covariance is 1. Otherwise, the covariance of two gate delays is assigned to be inversely proportional to the distance of the centers of the grid blocks that the two gates belong to.

We assume a risk aversion level of $\alpha = 0.9$. For each benchmark, we first perform three deterministic optimizations with the parameter $\gamma = 0, 1$, and 3 and obtain three solutions. Then we run our Incremental Risk Aversion Retiming algorithm with the initial retiming being the solution obtained from the above deterministic optimizations with $\gamma = 1$. Our algorithm is allowed to run for at most 50 iterations before one solution is obtained. The other parameters are $N = 500$ and $\beta = 1.01$. The conditional value-at-risk measure of the clock period for each solution is evaluated by performing Monte Carlo analysis for 10000 samples to ensure accuracy. The results are reported in Table I as follows. The statistics of the circuits are reported in the columns “ $|V|$ ” and “ $|E|$ ”. Under the

TABLE I

RESULTS COMPARISON BETWEEN DETERMINISTIC APPROACH AND OUR ALGORITHM.

name	Statistics		CVaR of Deterministic Approach				Ours				
	V	E	init	$\gamma = 0$	$\gamma = 1$	$\gamma = 3$	best	CVaR	impr.	# R	t(s)
s27	11	19	13.29	13.29	13.12	13.12	13.12	13.12	0.00%	1	0.0
s208.1	105	182	26.78	23.22	23.22	23.22	23.22	23.15	0.28%	2	0.1
s298	120	250	35.03	28.31	28.31	28.50	28.31	28.16	0.55%	50	1.2
s382	159	312	48.97	32.27	32.27	32.27	32.27	31.85	1.32%	5	0.2
s386	160	354	57.08	53.59	53.59	53.59	53.59	53.59	0.00%	50	1.6
s344	161	280	47.15	33.28	32.69	32.69	32.69	32.24	1.38%	50	1.5
s349	162	284	46.97	33.08	32.62	32.62	32.62	32.40	0.68%	50	1.5
s400	165	326	50.23	34.23	34.23	34.01	34.01	33.24	2.26%	7	0.2
s420.1	219	384	38.69	27.17	27.17	27.17	27.17	27.03	0.53%	3	0.2
s444	182	358	51.87	34.34	34.02	34.73	34.02	33.55	1.38%	10	0.4
s510	212	431	50.54	49.33	49.53	49.53	49.33	49.33	0.00%	50	2.1
s526	194	451	49.61	35.15	34.87	34.87	34.87	33.48	3.99%	50	2.0
s641	380	563	154.35	154.35	154.35	154.35	154.35	153.40	0.62%	8	0.6
s713	394	614	167.47	167.47	167.47	167.47	167.47	167.32	0.09%	5	0.4
s820	290	776	133.84	133.08	133.08	133.84	133.08	132.94	0.10%	50	3.1
s832	288	788	134.14	130.73	130.73	130.73	130.73	130.73	0.00%	50	3.1
s838.1	447	788	63.28	41.79	41.39	42.45	41.39	41.16	0.56%	50	4.1
s953	396	766	61.12	52.17	51.98	51.98	51.98	51.98	0.00%	50	3.8
s1196	530	1023	66.75	65.95	65.95	65.95	65.95	65.88	0.11%	2	0.3
s1238	509	1055	70.67	70.67	70.67	70.67	70.67	70.66	0.01%	2	0.3
s1423	658	1169	211.37	160.50	160.50	160.50	160.50	158.91	0.99%	12	1.5
s1488	654	1406	190.86	161.60	161.60	161.60	161.60	161.60	0.00%	50	6.7
s1494	648	1412	196.48	174.32	174.32	174.32	174.32	174.32	0.00%	50	6.6
s5378	2780	4261	66.64	66.64	66.64	66.64	66.64	64.05	3.89%	50	29.4
s9234.1	5598	4604	114.28	114.28	115.76	115.76	114.28	113.63	0.57%	50	50.6
s13207.1	7952	11082	193.40	123.08	122.62	123.10	122.62	118.85	3.08%	50	120.8
s15850.1	9773	13566	243.37	120.56	119.73	119.51	119.51	109.88	8.06%	50	156.0
s35932	16066	28589	187.44	170.62	172.38	177.34	170.62	171.84	-0.71%	50	265.3
s38417	22180	31127	173.83	97.46	96.62	97.18	96.62	93.69	3.03%	50	457.3
s38584.1	19254	33060	267.32	245.73	247.31	246.22	245.73	242.41	1.35%	50	445.3
runtimes \leq 1 second											

column “Deterministic Approach”, we report the conditional value-at-risk measure of the clock period for the original circuit before retiming in column “init” and report those of the three solutions obtained by the deterministic optimizations in the columns “ $\gamma = 0$ ”, “ $\gamma = 1$ ”, and “ $\gamma = 3$ ”. The column “best” shows the best one from the previous 4 columns, which is the best solution that one can get through the deterministic approach. The runtimes of the deterministic approach are all within 1 second and are thus excluded from being reported here. The results from our algorithm is reported under the column “Ours”. The conditional value-at-risk measure of the clock period is reported in the column “CVaR”. The improvement in percentage compared to the one in the column “best” is reported in the column “impr.”. The number of the iterations performed is reported in the column “# R” and the runtime in seconds is reported in the column “t(s)”. Note that for most of the benchmark, computing the subgradient uses more than 90% of the runtime. It can be seen from the table that our algorithm improves the solution quality for almost every benchmark circuit for up to 8% within fair amount of runtimes.

In addition, we compare the solutions in terms of the timing yield and report the results in Table II. The target clock periods are determined such that the solution obtained by our algorithm will have a timing yield of 90%. This table shows that the timing yield can be effectively improved by optimizing the conditional value-at-risk measure.

VII. CONCLUSIONS

In this paper, we formulated the risk aversion min-period retiming problem to optimize the clock period of a circuit under process variations. The formulation is based on conventional two-stage stochastic program with fixed recourse with a risk aversion objective. We proved that the proposed problem is an integer convex program. We gave an analytical formula for the subgradient of the objective function and proposed to compute an approximation of the subgradient by sampling from a black box. We presented a heuristic incremental

algorithm to solve the proposed problem. and the effectiveness of our proposed approach is confirmed by the experimental results.

REFERENCES

- [1] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer. *Statistical Timing Analysis: From Basic Principles to State of the Art*. IEEE TCAD, 27(4):589–607, Apr. 2008.
- [2] S. H. Choi, B. C. Paul, and K. Roy. *Novel Sizing Algorithm for Yield Improvement under Process Variation in Nanometer Technology*. In DAC, pages 454–459, 2004.
- [3] M. R. Guthaus, N. Venkateswarant, C. Visweswariah, and V. Zolotov. *Gate Sizing Using Incremental Parameterized Statistical Timing Analysis*. In ICCAD, pages 1029–1036, 2005.
- [4] D. Sinha, N. Shenoy, and H. Zhou. *Statistical Timing Yield Optimization by Gate Sizing*. IEEE TVLSI, 14(10):1140–1146, Oct. 2006.
- [5] A. Srivastava, K. Chopra, S. Shah, D. Sylvester, and D. Blaauw. *A Novel Approach to Perform Gate-Level Yield Analysis and Optimization Considering Correlated Variations in Power and Performance*. IEEE TCAD, 27(2):272–285, Feb. 2008.
- [6] M. Mani, A. Devgan, M. Orshansky, and Y. Zhan. *A Statistical Algorithm for Power- and Timing-Limited Parametric Yield Optimization of Large Integrated Circuits*. IEEE TCAD, 26(10):1790–1802, Oct. 2007.
- [7] J. Singh, Z. -Q. Luo, S. Sapatnekar. *A Geometric Programming-Based Worst-Case Gate Sizing Method Incorporating Spatial Correlation*. IEEE TCAD, 27(2):295–308, Feb. 2008.
- [8] A. Davoodi and A. Srivastava. *Variability Driven Gate Sizing for Binning Yield Optimization*. In DAC, pages 959–964, 2006.
- [9] R. J. -B. Wets. *Stochastic Programs with Fixed Recourse: The Equivalent Deterministic Program*. SIAM Review, 16(3):309–339, Jul. 1974.
- [10] C. E. Leiserson and J. B. Saxe. *Retiming Synchronous Circuitry*. Algorithmica, 6(1):5–35, 1991.
- [11] J. Wang and H. Zhou. *Minimal Period Retiming under Process Variations*. In GLSVLSI, pages 131–135, 2004.
- [12] H. Zhou. *Deriving a New Efficient Algorithm for Min-Period Retiming*. In ASP-DAC, pages 990–993, 2005.
- [13] R. T. Rockafellar. *Coherent Approaches to Risk in Optimization under Uncertainty*. INFORMS Tutorials in Operations Research, pages 38–61, 2007.
- [14] J. Wang and H. Zhou. *An Efficient Incremental Algorithm for Min-Area Retiming*. In DAC, pages 528–533, 2008.
- [15] D. B. Shmoys and C. Swamy. *Stochastic Optimization is (Almost) as easy as Deterministic Optimization*. IEEE Symposium on Foundations of Computer Science, pages 228–237, 2004.
- [16] N. Immerlica, D. Karger, M. Minkoff, and V. S. Mirrokni. *On the Costs and Benefits of Procrastination: Approximation Algorithms for Stochastic Combinatorial Optimization Problems*. ACM-SIAM Symposium on Discrete Algorithms, pages 691–700, 2004.
- [17] A. M. -C. So, J. Zhang, and Y. Ye. *Stochastic Combinatorial Optimization with Controllable Risk Aversion Level*. Lecture Notes in Computer Science, vol. 4110, pages 224–235, Aug. 2006.
- [18] T. F. Chan, J. Cong, J. R. Shinnerl, K. Sze, and M. Xie. *mPL6: Enhanced Multilevel Mixed-Size Placement*. In ISPD, pages 212–214, 2006.