Risk Aversion Min-Period Retiming Under Process Variations

Jia Wang
Illinois Institute of Technology
Chicago, Illinois, USA

Hai Zhou
Fudan University, China
Northwestern University, USA

January, 2009
Outline

Introduction

Problem Formulation

Algorithm Details

Experiments

Conclusions
Process Variations

- VLSI feature sizes keep shrinking with technology scaling.
- Process variations become significant.
  - Chips are realized randomly during manufacturing around nominal values, instead of precisely as expected by the chip designers.
- Affect manufacturing yield and system reliability.
- A critical issue of Design for Manufacturability (DFM)
  - Statistical circuit analysis: Monte Carlo, SSTA
  - Statistical circuit optimization
Statistical Circuit Optimization

- **Statistical optimization is difficult**
  - Optimize a circuit for many or infinity number of variation corners at the same time.

- **Gate sizing under process variations**
  - Majority of previous statistical circuit optimization works.
  - Sensitivity guided iterative improvement heuristics [Guthaus et al. 2005], [Sinha et al. 2006], [Srivastava et al. 2008]
  - Convex optimization for the worst cases [Mani et al. 2007], [Singh et al. 2008]
  - Two-stage stochastic program with fixed recourse [Davoodi et al. 2006]

- **Min-period retiming under process variations**
  - Iterative push-down heuristics [Wang et al. 2004]
  - No theoretical guarantee for optimality
Statistical Circuit Optimization

- Statistical optimization is difficult
  - Optimize a circuit for many or infinity number of variation corners at the same time.

- Gate sizing under process variations
  - Majority of previous statistical circuit optimization works.
  - Sensitivity guided iterative improvement heuristics
    - [Guthaus et al. 2005], [Sinha et al. 2006], [Srivastava et al. 2008]
  - Convex optimization for the worst cases
    - [Mani et al. 2007], [Singh et al. 2008]
  - Two-stage stochastic program with fixed recourse [Davoodi et al. 2006]

- Min-period retiming under process variations
  - Iterative push-down heuristics [Wang et al. 2004]
  - No theoretical guarantee for optimality
Statistical Circuit Optimization

- Statistical optimization is difficult
  - Optimize a circuit for many or infinity number of variation corners at the same time.
- Gate sizing under process variations
  - Majority of previous statistical circuit optimization works.
  - Sensitivity guided iterative improvement heuristics
    [Guthaus et al. 2005], [Sinha et al. 2006], [Srivastava et al. 2008]
  - Convex optimization for the worst cases
    [Mani et al. 2007], [Singh et al. 2008]
  - Two-stage stochastic program with fixed recourse [Davoodi et al. 2006]
- Min-period retiming under process variations
  - Iterative push-down heuristics [Wang et al. 2004]
  - No theoretical guarantee for optimality
Our Contribution

- Formulate the Risk Aversion Min-Period Retiming problem
  - Based on two-stage stochastic programs with fixed recourse and the conditional value-at-risk measure
- Prove that the proposed formulation is an integer convex program
- Derive an analytical formula for the subgradient of the objective function.
  - Simplify sensitivity computation
  - Handle arbitrary distribution by sampling
- Present an incremental algorithm to solve the proposed problem heuristically.
Deterministic Retiming

- Relocate flip-flops (FFs) w/o changing circuit functionality. [Leiserson and Saxe 83]
  - Powerful sequential transformation to reschedule both computation and communication.
- Circuit graph $G = (V, E)$
  - Gate delay: $d(v), \; v \in V$
  - # FFs on interconnects: $w(u, v), \; (u, v) \in E$
- Retiming is represented by an integer-valued vertex label.

$$r : V \rightarrow \mathbb{Z}$$

For gate $v$, $r(v)$ is the # FFs moved from all its fanouts to all its fanins.
Deterministic Retiming

- Relocate flip-flops (FFs) w/o changing circuit functionality. [Leiserson and Saxe 83]
  - Powerful sequential transformation to reschedule both computation and communication.

- Circuit graph $G = (V, E)$
  - Gate delay: $d(v), v \in V$
  - # FFs on interconnects: $w(u, v), (u, v) \in E$

- Retiming is represented by an integer-valued vertex label.

\[ r : V \rightarrow \mathbb{Z} \]

For gate $v$, $r(v)$ is the # FFs moved from all its fanouts to all its fanins.
Deterministic Retiming

- Relocate flip-flops (FFs) w/o changing circuit functionality. [Leiserson and Saxe 83]
  - Powerful sequential transformation to reschedule both computation and communication.

- Circuit graph $G = (V, E)$
  - Gate delay: $d(\nu), \nu \in V$
  - # FFs on interconnects: $w(\mu, \nu), (\mu, \nu) \in E$

- Retiming is represented by an integer-valued vertex label.

$$r : V \rightarrow \mathbb{Z}$$

For gate $\nu$, $r(\nu)$ is the # FFs moved from all its fanouts to all its fanins.
Deterministic Min-Period Retiming

- Validity constraints:
  \[ \forall (u, v) \in E : w(u, v) - r(u) + r(v) \geq 0 \]

- Timing constraints for the clock period \( \phi \):
  \[ \forall (u, v) \in E : w(u, v) = r(u) - r(v) \Rightarrow t(u) + d(v) \leq t(v) \]

  \[ \forall v \in V : d(v) \leq t(v) \leq \phi \]

- Min-period objective: find \( r \) to minimize \( \phi \)
Deterministic Min-Period Retiming

- **Validity constraints:**

\[ \forall (u, v) \in E : w(u, v) - r(u) + r(v) \geq 0 \]

- **Timing constraints for the clock period \( \phi \):**

\[ \forall (u, v) \in E : w(u, v) = r(u) - r(v) \Rightarrow t(u) + d(v) \leq t(v) \]

\[ \forall v \in V : d(v) \leq t(v) \leq \phi \]

- **Min-period objective:** find \( r \) to minimize \( \phi \)
Deterministic Min-Period Retiming

- **Validity constraints:**
  \[
  \forall (u, v) \in E : w(u, v) - r(u) + r(v) \geq 0
  \]

- **Timing constraints for the clock period** $\phi$:
  \[
  \forall (u, v) \in E : w(u, v) = r(u) - r(v) \Rightarrow t(u) + d(v) \leq t(v)
  \]

- **Min-period objective:** find $r$ to minimize $\phi$
  \[
  \forall v \in V : d(v) \leq t(v) \leq \phi
  \]
Decision making under uncertainty

- The output depends on decision variables and uncertain parameters not available at the time of decision.
- E.g. designer specifies circuit parameters at design time – the fabricated chips will be affected by process variations

First stage: determine the values of decision variables
- Incur an initial cost

Second stage: uncertain parameters are realized
- Incur a second stage cost through a fixed recourse, i.e.,
  - A known deterministic program of both the decision variables and the realized uncertain parameters

Objective: “minimize” the total cost
- The total cost is a random number.
- What is the meaning of “minimize”? 
Decision making under uncertainty
- The output depends on decision variables and uncertain parameters not available at the time of decision.
- E.g. designer specifies circuit parameters at design time – the fabricated chips will be affected by process variations

First stage: determine the values of decision variables
- Incur an initial cost

Second stage: uncertain parameters are realized
- Incur a second stage cost through a fixed recourse, i.e.,
- A known deterministic program of both the decision variables and the realized uncertain parameters

Objective: “minimize” the total cost
- The total cost is a random number.
- What is the meaning of “minimize”?
Decision making under uncertainty

- The output depends on decision variables and uncertain parameters not available at the time of decision.
- E.g. designer specifies circuit parameters at design time – the fabricated chips will be affected by process variations

First stage: determine the values of decision variables

- Incur an initial cost

Second stage: uncertain parameters are realized

- Incur a second stage cost through a fixed recourse, i.e.,
- A known deterministic program of both the decision variables and the realized uncertain parameters

Objective: “minimize” the total cost

- The total cost is a random number.
- What is the meaning of “minimize”? 
Decision making under uncertainty
- The output depends on decision variables and uncertain parameters not available at the time of decision.
- E.g. designer specifies circuit parameters at design time – the fabricated chips will be affected by process variations

First stage: determine the values of decision variables
- Incur an initial cost

Second stage: uncertain parameters are realized
- Incur a second stage cost through a fixed recourse, i.e.,
- A known deterministic program of both the decision variables and the realized uncertain parameters

Objective: “minimize” the total cost
- The total cost is a random number.
- What is the meaning of “minimize”? 
Coherent Measure of Risk

- Measure of risk: map a random variable into a real number
  - $E[X]$, expectation of $X$
  - $\text{Yield}_\phi[X] \triangleq P(X \leq \phi)$

- Coherent measure of risk [Rockafellar 2007]
  - Transfer the convexity of the second stage program to the objective function.
  - $E[X]$ is coherent, but not quite interesting.

- Conditional value-at-risk for a risk aversion level $\alpha$

\[
\text{CVaR}_\alpha[X] \triangleq E[X|X > \text{VaR}_\alpha[X]]
\]

where $\text{VaR}_\alpha[X]$ is the value satisfying $P(X \leq \text{VaR}_\alpha[X]) = \alpha$
Coherent Measure of Risk

- Measure of risk: map a random variable into a real number
  - $\mathbb{E}[X]$, expectation of $X$
  - $\text{Yield}_\phi[X] \triangleq P(X \leq \phi)$
- Coherent measure of risk [Rockafellar 2007]
  - Transfer the convexity of the second stage program to the objective function.
  - $\mathbb{E}[X]$ is coherent, but not quite interesting.
- Conditional value-at-risk for a risk aversion level $\alpha$

$$\text{CVaR}_\alpha[X] \triangleq \mathbb{E}[X|X > \text{VaR}_\alpha[X]]$$

where $\text{VaR}_\alpha[X]$ is the value satisfying $P(X \leq \text{VaR}_\alpha[X]) = \alpha$
Coherent Measure of Risk

- Measure of risk: map a random variable into a real number
  - $E[X]$, expectation of $X$
  - $\text{Yield}_\phi[X] \overset{\Delta}{=} P(X \leq \phi)$

- Coherent measure of risk [Rockafellar 2007]
  - Transfer the convexity of the second stage program to the objective function.
  - $E[X]$ is coherent, but not quite interesting.

- Conditional value-at-risk for a risk aversion level $\alpha$

\[
\text{CVaR}_\alpha[X] \overset{\Delta}{=} E[X|X > \text{VaR}_\alpha[X]]
\]

where $\text{VaR}_\alpha[X]$ is the value satisfying $P(X \leq \text{VaR}_\alpha[X]) = \alpha$
Outline

- Introduction
- Problem Formulation
- Algorithm Details
- Experiments
- Conclusions
Gate Delay under Process Variations

- Probabilistic space for process variations: $\Omega$
- For a particular variation $\omega \in \Omega$,
  - Random gate delays $d_\omega : V \rightarrow \mathbb{R}^*$
  - Minimum clock period for valid retiming $r$: $\phi_\omega(r)$
Risk Aversion Min-Period Retiming

Risk aversion min-period retiming for risk aversion level $\alpha$

Minimize $\text{CVaR}_\alpha[\phi_\omega(r)]$ s.t.

$w_r(u, v) \geq 0, \forall (u, v) \in E$, and $r(v) \in \mathbb{Z}, \forall v \in V$,

where $\phi_\omega(r)$ is the minimum objective of

Minimize $\phi$ s.t.

$w_r(u, v) = 0 \Rightarrow t(v) \geq d_\omega(v) + t(u), \forall (u, v) \in E$,

$d_\omega(v) \leq t(v) \leq \phi, \forall v \in V$. 
The second stage program is not mathematical programming.

Compute $\phi_\omega(r)$ by enumerating paths

\[
\exists \text{ simple path } p^* \text{ in } G, \phi_\omega(r) = d_\omega(p^*) \land w_r(p^*) = 0,
\]

\[
\forall \text{ simple path } p \text{ in } G, \phi_\omega(r) \geq \frac{d_\omega(p)}{w_r(p) + 1}.
\]

Continuous relaxation

Minimize $\text{CVaR}_\alpha[\phi_\omega(r)]$ s.t. $w_r(u, v) \geq 0, \forall (u, v) \in E,$

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

- The second stage program is not mathematical programming.
- Compute $\phi_\omega(r)$ by enumerating paths

$$\exists \text{ simple path } p^* \text{ in } G, \phi_\omega(r) = d_\omega(p^*) \land w_r(p^*) = 0,$$

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

- Continuous relaxation

Minimize $\text{CVaR}_\alpha[\phi_\omega(r)]$ s.t. $w_r(u, v) \geq 0$, $\forall(u, v) \in E$,

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

- The second stage program is not mathematical programming.
- Compute $\phi_\omega(r)$ by enumerating paths

$$\exists \text{ simple path } p^* \text{ in } G, \phi_\omega(r) = d_\omega(p^*) \land w_r(p^*) = 0,$$
$$\forall \text{ simple path } p \text{ in } G, \phi_\omega(r) \geq \frac{d_\omega(p)}{w_r(p) + 1}.$$

- Continuous relaxation

Minimize $\text{CVaR}_\alpha[\phi_\omega(r)]$ s.t. $w_r(u, v) \geq 0, \forall (u, v) \in E,$

where $\phi_\omega(r) = \max_{\text{simple path } p \text{ in } G} \frac{d_\omega(p)}{w_r(p) + 1}.$
Convexity of the Minimal Clock Period

- $p_\omega$: critical path for retiming $r$ from $u_\omega$ to $v_\omega$, i.e.

$$\phi_\omega(r) = \frac{d_\omega(p_\omega)}{w_r(p_\omega) + 1}$$

- Intuition: inserting/removing 1 FF to/from $p_\omega$ will decrease/increase $\phi_\omega(r)$ by at most/least $\frac{\phi_\omega(r)}{w_r(p_\omega)+1}$

- Let $s_\omega(u_\omega) = 1$, $s_\omega(v_\omega) = -1$, and $s_\omega(x) = 0$ for any other $x \in V$.

$$\phi_\omega(r') - \phi_\omega(r) \geq \sum_{u \in V} \frac{\phi_\omega(r)s_\omega(u)}{w_r(p_\omega) + 1} (r'(u) - r(u)).$$
Convexity of the Minimal Clock Period

- $p_\omega$: critical path for retiming $r$ from $u_\omega$ to $v_\omega$, i.e.

$$\phi_\omega(r) = \frac{d_\omega(p_\omega)}{w_r(p_\omega) + 1}$$

- Intuition: inserting/removing 1 FF to/from $p_\omega$ will decrease/increase $\phi_\omega(r)$ by at most/least

$$\phi_\omega(r) \left( \frac{r'(u) - r(u)}{w_r(p_\omega) + 1} \right).$$

- Let $s_\omega(u_\omega) = 1$, $s_\omega(v_\omega) = -1$, and $s_\omega(x) = 0$ for any other $x \in V$. 

$$\phi_\omega(r') - \phi_\omega(r) \geq \sum_{u \in V} \frac{\phi_\omega(r)s_\omega(u)}{w_r(p_\omega) + 1} (r'(u) - r(u)).$$
Convexity of the Minimal Clock Period

- $p_\omega$: critical path for retiming $r$ from $u_\omega$ to $v_\omega$, i.e.

$$\phi_\omega(r) = \frac{d_\omega(p_\omega)}{w_r(p_\omega) + 1}$$

- Intuition: inserting/removing 1 FF to/from $p_\omega$ will decrease/increase $\phi_\omega(r)$ by at most (least) $\frac{\phi_\omega(r)}{w_r(p_\omega) + 1}$

- Let $s_\omega(u_\omega) = 1$, $s_\omega(v_\omega) = -1$, and $s_\omega(x) = 0$ for any other $x \in V$.

$$\phi_\omega(r') - \phi_\omega(r) \geq \sum_{u \in V} \frac{\phi_\omega(r) s_\omega(u)}{w_r(p_\omega) + 1} (r'(u) - r(u)).$$
Convexity of the Continuous Relaxation

- $l_\omega(r)$: 1 if $\phi_\omega(r) \geq \text{VaR}_\alpha[\phi_\omega(r)]$ and 0 otherwise.
- Convexity of the objective function from that of $\phi_\omega(r)$

\[
\text{CVaR}_\alpha[\phi_\omega(r')] - \text{CVaR}_\alpha[\phi_\omega(r)] \geq \sum_{u \in V} \frac{r'(u) - r(u)}{1 - \alpha} \mathbb{E}[l_\omega(r) \frac{\phi_\omega(r)s_\omega(u)}{w_r(p_\omega) + 1}]
\]

- Convex constraints: $w_r(u, v) \geq 0, \forall (u, v) \in E$
- The continuous relaxation is a convex program.
- Risk aversion min-period retiming requires an integer optimal solution of the continuous relaxation.
Convexity of the Continuous Relaxation

- $I_\omega(r)$: 1 if $\phi_\omega(r) \geq \text{VaR}_\alpha[\phi_\omega(r)]$ and 0 otherwise.
- Convexity of the objective function from that of $\phi_\omega(r)$

$$\text{CVaR}_\alpha[\phi_\omega(r')] - \text{CVaR}_\alpha[\phi_\omega(r)] \geq \sum_{u \in V} \frac{r'(u) - r(u)}{1 - \alpha} \mathbb{E}[I_\omega(r) \frac{\phi_\omega(r) s_\omega(u)}{w_r(p_\omega) + 1}]$$

- Convex constraints: $w_r(u, v) \geq 0, \forall (u, v) \in E$
- The continuous relaxation is a convex program.
- Risk aversion min-period retiming requires an integer optimal solution of the continuous relaxation.
Convexity of the Continuous Relaxation

- \( I_\omega(r) \): 1 if \( \phi_\omega(r) \geq \text{VaR}_\alpha[\phi_\omega(r)] \) and 0 otherwise.
- Convexity of the objective function from that of \( \phi_\omega(r) \)

\[
\text{CVaR}_\alpha[\phi_\omega(r')]-\text{CVaR}_\alpha[\phi_\omega(r)] \\
\geq \sum_{u \in V} \frac{r'(u)-r(u)}{1-\alpha} \mathbb{E}[I_\omega(r) \frac{\phi_\omega(r)s_\omega(u)}{w_r(p_\omega)+1}]
\]

- Convex constraints: \( w_r(u, v) \geq 0, \forall (u, v) \in E \)
- The continuous relaxation is a convex program.
- Risk aversion min-period retiming requires an integer optimal solution of the continuous relaxation.
Outline

Introduction

Problem Formulation

Algorithm Details

Experiments

Conclusions
Subgradient Computation

- Subgradient of $\text{CVaR}_\alpha[\phi_\omega(r)]$ was derived when proving its convexity.

- Compute subgradient by drawing independent samples from a black box model representing $\Omega$.
  - Can handle arbitrary distribution of process variations.
  - Reuse existing deterministic analysis algorithms.
  - May develop analytical methods to speed up computation for specific distributions.

- Most previous works on statistical gate sizing approximated subgradient (sensitivity) by computing secant directions.
  - Require multiple runs of SSTA – time consuming.
  - Trade-off accuracy for running time.
Subgradient Computation

- Subgradient of $\text{CVaR}_\alpha[\phi_\omega(r)]$ was derived when proving its convexity.

- Compute subgradient by drawing independent samples from a black box model representing $\Omega$.
  - Can handle arbitrary distribution of process variations.
  - Reuse existing deterministic analysis algorithms.
  - May develop analytical methods to speed up computation for specific distributions.

- Most previous works on statistical gate sizing approximated subgradient (sensitivity) by computing secant directions.
  - Require multiple runs of SSTA – time consuming.
  - Trade-off accuracy for running time.
Subgradient Computation

- Subgradient of \( \text{CVaR}_\alpha[\phi_\omega(r)] \) was derived when proving its convexity.
- Compute subgradient by drawing independent samples from a black box model representing \( \Omega \).
  - Can handle arbitrary distribution of process variations.
  - Reuse existing deterministic analysis algorithms.
  - May develop analytical methods to speed up computation for specific distributions.
- Most previous works on statistical gate sizing approximated subgradient (sensitivity) by computing secant directions.
  - Require multiple runs of SSTA – time consuming.
  - Trade-off accuracy for running time.
Subgradient Guided Optimization

- An intuitive idea: iteratively incremental improvement

\[
\text{Minimize } \sum_{v \in V} \hat{g}_r(v)(r'(v) - r(v)) \quad \text{s.t.}
\]
\[
w_{r'}(u, v) \geq 0, \forall (u, v) \in E, \text{ and } 0 \leq r'(v) - r(v) \leq 1, \forall v \in V,
\]

where \( \hat{g}_r \) is the subgradient.

- Dual of network-flow problem: integer-valued optimal solution
- Not good in practice since even changing \( r \) by 1 for some vertices will result in huge changes in the minimum clock period.
- Need additional constraints to improve the accuracy of the estimation.

- Cutting plane techniques cannot guarantee an integer-valued optimal solution.
Subgradient Guided Optimization

- An intuitive idea: iteratively incremental improvement

\[
\text{Minimize } \sum_{v \in V} \hat{g}_r(v)(r'(v) - r(v)) \quad \text{s.t.}
\]
\[
w_{r'}(u, v) \geq 0, \forall (u, v) \in E, \text{ and } 0 \leq r'(v) - r(v) \leq 1, \forall v \in V,
\]

where \( \hat{g}_r \) is the subgradient.

- Dual of network-flow problem: integer-valued optimal solution
- Not good in practice since even changing \( r \) by 1 for some vertices will result in huge changes in the minimum clock period.
- Need additional constraints to improve the accuracy of the estimation.

- Cutting plane techniques cannot guarantee an integer-valued optimal solution.
Use statistical timing critical paths as additional constraints without affecting optimality

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

Remain dual of network-flow problem – integer-valued optimal solution

Inefficient in practice, replace with

\[ w_{r'}(p) \geq 1, \forall \text{ simple path } p \text{ satisfying } \bar{d}(p) > \beta \bar{\phi}(r). \]

where \( \bar{d} \) are the nominal delays, \( \bar{\phi}(r) \) is the nominal minimum clock period, and \( \beta \geq 1 \) is a parameter specified by the designer.
Use statistical timing critical paths as additional constraints without affecting optimality

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

Remain dual of network-flow problem – integer-valued optimal solution

Inefficient in practice, replace with

\[ w_{r'}(p) \geq 1, \forall \text{ simple path } p \text{ satisfying } \bar{d}(p) > \beta \bar{\phi}(r). \]

where \( \bar{d} \) are the nominal delays, \( \bar{\phi}(r) \) is the nominal minimum clock period, and \( \beta \geq 1 \) is a parameter specified by the designer.
Use statistical timing critical paths as additional constraints without affecting optimality

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

Remain dual of network-flow problem – integer-valued optimal solution

Inefficient in practice, replace with

\[ w_r'(p) \geq 1, \forall \text{ simple path } p \text{ satisfying } \overline{d}(p) > \beta \overline{\phi}(r). \]

where \( \overline{d} \) are the nominal delays, \( \overline{\phi}(r) \) is the nominal minimum clock period, and \( \beta \geq 1 \) is a parameter specified by the designer.
Incremental Risk Aversion Retiming

- Incremental risk aversion retiming

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

- Similar to deterministic min-area retiming

- Can be solved by incremental deterministic min-area retiming algorithm [Wang et al. 2008]
Incremental Risk Aversion Retiming

- Incremental risk aversion retiming

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

\[
w_{r'}(u, v) \geq 0, \forall (u, v) \in E, \text{ and } 0 \leq r'(v) - r(v) \leq 1, \forall v \in V,
\]

\[
w_{r'}(p) \geq 1, \forall \text{ simple path } p \text{ satisfying } \overline{d}(p) > \beta \phi(r).
\]

- Similar to deterministic min-area retiming

- Can be solved by incremental deterministic min-area retiming algorithm [Wang et al. 2008]
The Incremental Risk Aversion Retiming Algorithm

- Iteratively improve any initial valid retiming $r$.
- In each iteration
  - Compute $\text{CVaR}_\alpha[\phi_\omega(r)]$ and record the best retiming so far
  - Compute the subgradient
  - Formulate and solve the incremental risk aversion retiming problem for $r'$
  - Claim optimality if $r' = r$ and stop
  - Update $r$ to $r'$. Stop if a predefined number of iterations have been reached.
Outline

- Introduction
- Problem Formulation
- Algorithm Details
- Experiments
- Conclusions
Experimental Setup

- **Benchmarks:** ISCAS89 sequential circuits
  - Placed onto a 4×4 grid to build a model for process variations
- Compared to a risk-aware deterministic approach similar to [Wang et al. 2004]
  - Assign each gate a delay of
    \[
    E[d_\omega(v)] + \gamma \sqrt{E[(d_\omega(v) - E[d_\omega(v)])^2]}
    \]
  - Run [Zhou 2005] for a min-period retiming
  - Take the one with the best CVaR for \(\gamma = 0, 1, 3\)
## Experimental Results

<table>
<thead>
<tr>
<th>name</th>
<th>Statistics</th>
<th>CVaR of Det.</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
<td>V</td>
<td>$</td>
</tr>
<tr>
<td>s1196</td>
<td>530</td>
<td>1023</td>
<td>66.75</td>
</tr>
<tr>
<td>s1238</td>
<td>509</td>
<td>1055</td>
<td>70.67</td>
</tr>
<tr>
<td>s1423</td>
<td>658</td>
<td>1169</td>
<td>211.37</td>
</tr>
<tr>
<td>s1488</td>
<td>654</td>
<td>1406</td>
<td>190.86</td>
</tr>
<tr>
<td>s1494</td>
<td>648</td>
<td>1412</td>
<td>196.48</td>
</tr>
<tr>
<td>s5378</td>
<td>2780</td>
<td>4261</td>
<td>66.64</td>
</tr>
<tr>
<td>s9234.1</td>
<td>5598</td>
<td>4604</td>
<td>114.28</td>
</tr>
<tr>
<td>s13207.1</td>
<td>7952</td>
<td>11082</td>
<td>193.40</td>
</tr>
<tr>
<td>s15850.1</td>
<td>9773</td>
<td>13566</td>
<td>243.37</td>
</tr>
<tr>
<td>s35932</td>
<td>16066</td>
<td>28589</td>
<td>187.44</td>
</tr>
<tr>
<td>s38417</td>
<td>22180</td>
<td>31127</td>
<td>173.83</td>
</tr>
<tr>
<td>s38584.1</td>
<td>19254</td>
<td>33060</td>
<td>267.32</td>
</tr>
</tbody>
</table>

runtimes $\leq$ 1s
Experimental Result: Timing Yield

<table>
<thead>
<tr>
<th>name</th>
<th>Deterministic Approach</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>init</td>
<td>$\gamma = 0$</td>
</tr>
<tr>
<td>s1196</td>
<td>88.2%</td>
<td>89.8%</td>
</tr>
<tr>
<td>s1238</td>
<td>90.0%</td>
<td>90.0%</td>
</tr>
<tr>
<td>s1423</td>
<td>22.3%</td>
<td>88.4%</td>
</tr>
<tr>
<td>s1488</td>
<td>63.0%</td>
<td>90.0%</td>
</tr>
<tr>
<td>s1494</td>
<td>71.2%</td>
<td>90.0%</td>
</tr>
<tr>
<td>s5378</td>
<td>82.5%</td>
<td>82.5%</td>
</tr>
<tr>
<td>s9234.1</td>
<td>90.6%</td>
<td>90.6%</td>
</tr>
<tr>
<td>s13207.1</td>
<td>82.8%</td>
<td>88.9%</td>
</tr>
<tr>
<td>s15850.1</td>
<td>46.5%</td>
<td>87.2%</td>
</tr>
<tr>
<td>s35932</td>
<td>74.2%</td>
<td>91.5%</td>
</tr>
<tr>
<td>s38417</td>
<td>0.2%</td>
<td>83.1%</td>
</tr>
<tr>
<td>s38584.1</td>
<td>85.3%</td>
<td>90.0%</td>
</tr>
</tbody>
</table>
Outline

Introduction

Problem Formulation

Algorithm Details

Experiments

Conclusions
Conclusions

- Formulate the Risk Aversion Min-Period Retiming problem based on two-stage stochastic programs with fixed recourse and the conditional value-at-risk measure.
- Prove the convexity of the proposed formulation and derive an analytical formula for the subgradient of the objective function.
- Present an incremental algorithm to solve the proposed problem heuristically.
- We expect similar techniques to be applied to other statistical circuit optimizations.
Q & A
Thank you!