Buffer sizing for Clock Networks Using Robust Geometric Programming

Laleh Behjat,
Logan Rakai, Amin Farshidi, Dave Westwick
University of Calgary,
Calgary, Canada

March 27, 2013

SCHULICH
School of Engineering
UNIVERSITY OF CALGARY

ALBERTA
INGENUITY
FUND

compute • calcul
CANADA

NSERC
CRSNG

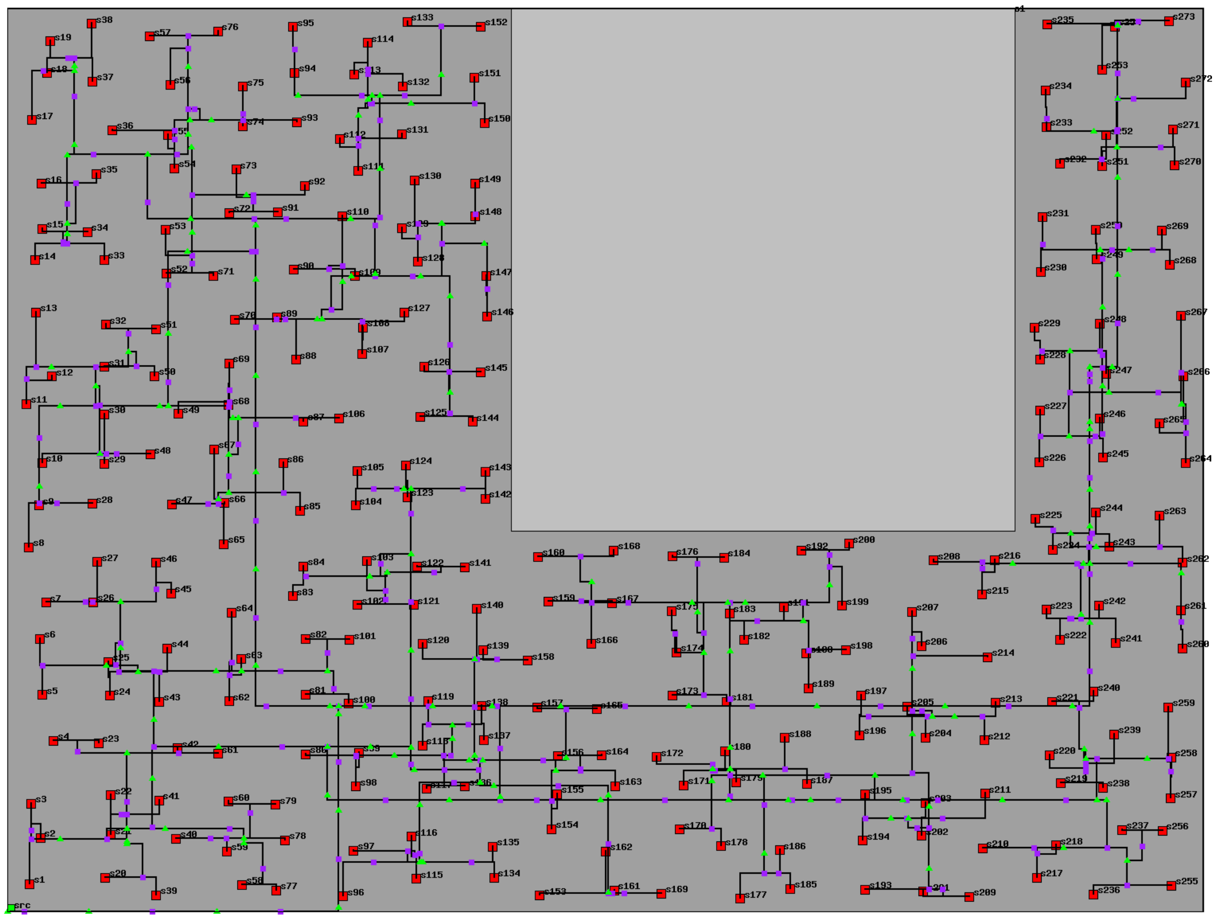
“The essence of Mathematics is not to make simple things complicated, but to make complicated things simple,” Dr. S. Gudder

Outline

- Introduction
- Clock Network Buffer Sizing:
  - Geometric Programming Model
  - Robust Model
- Conclusions and Future Work

Introduction

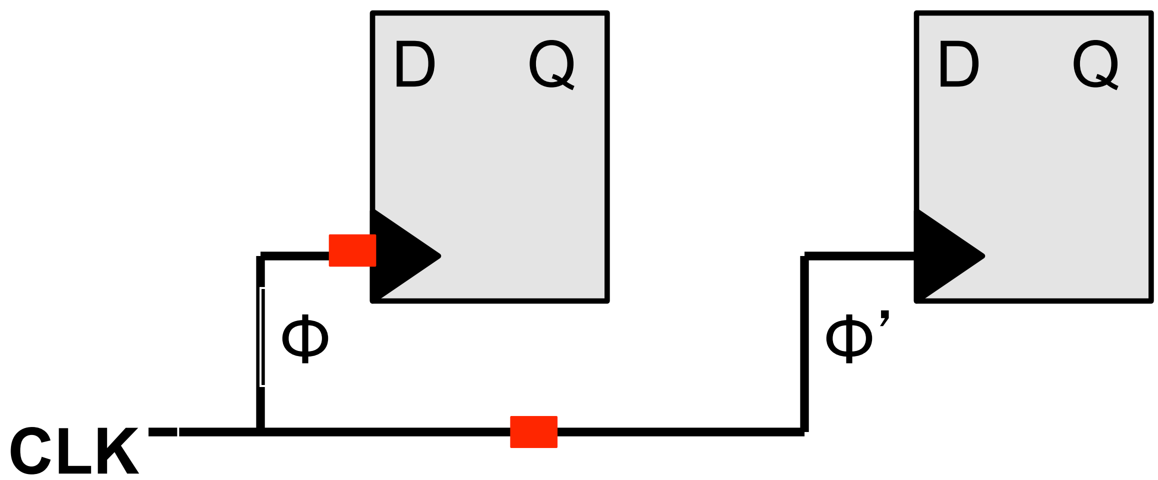
Clock Networks



Clock network for Circuit ISPD09-31

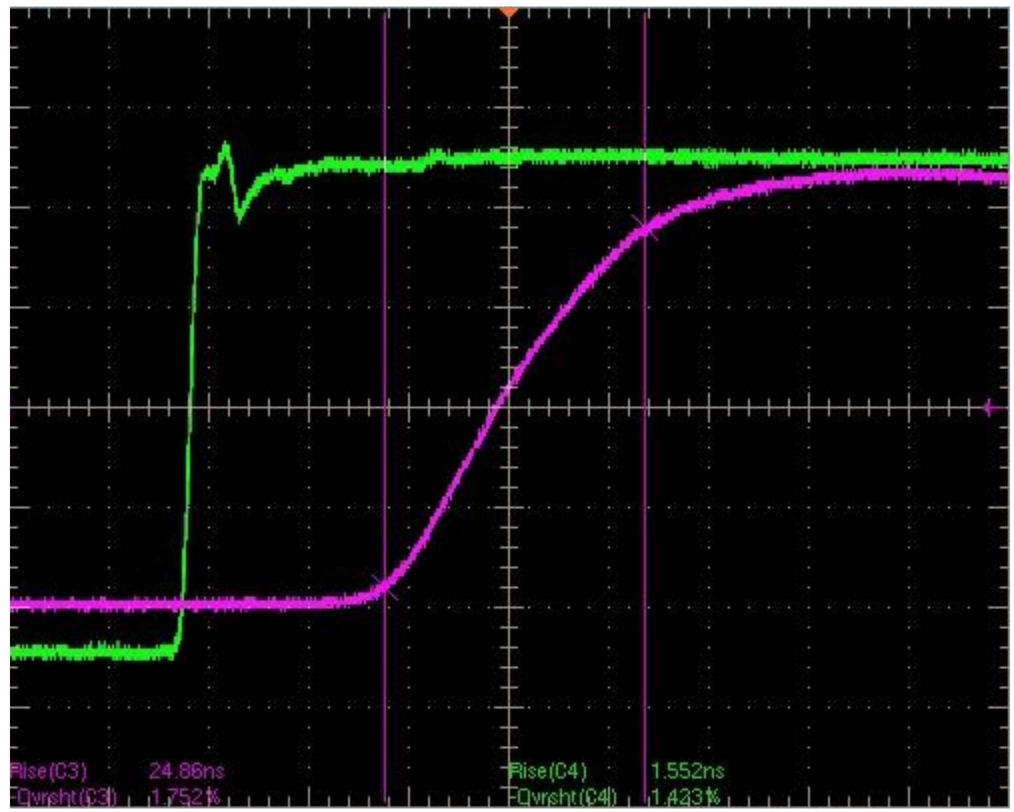
Buffer sizing for Clock Networks

- Skew = $|\Phi - \Phi'|$
- Keep as low as possible

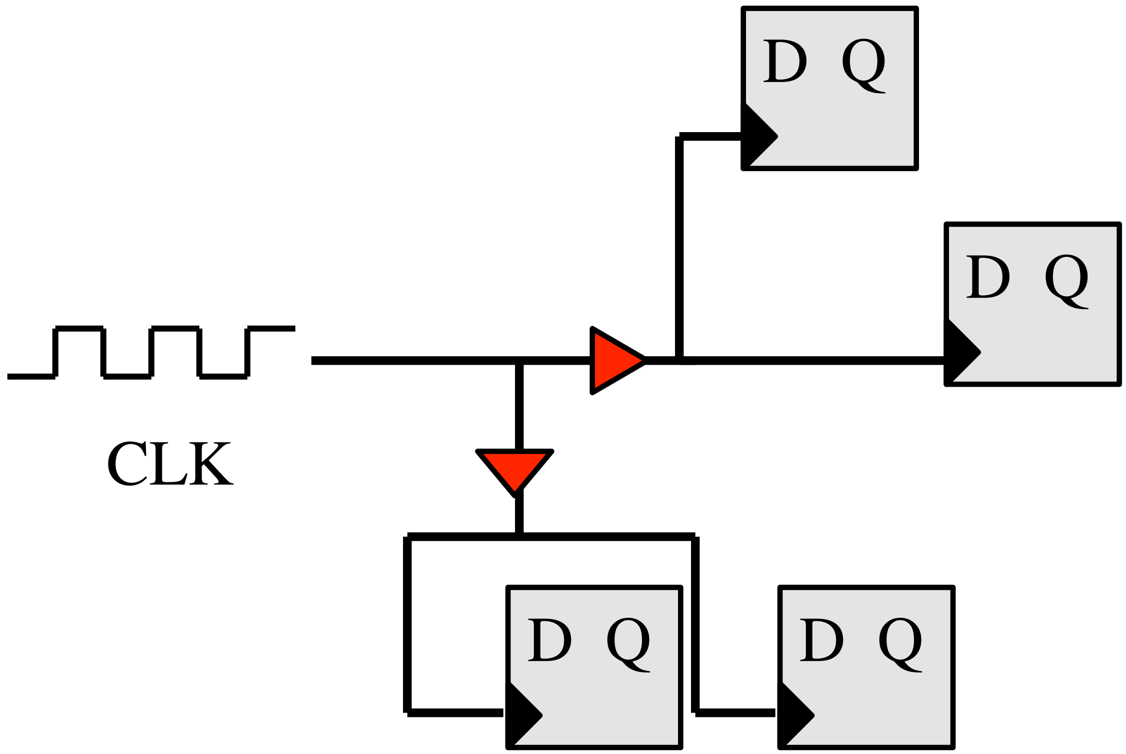


Buffer sizing for Clock Networks

- Slew:
  - Rise/fall time
- Keep as low as possible



Buffer sizing for Clock Networks



Generally, Increasing the sizes of buffers will reduce skew and slew

Buffer sizing for Clock Networks

However 40% of total power is used in Clock networks [1].

[1] Q. Zhu, "High-Speed Clock Network Design." Boston: Kluwer Academic Publishers, 2003.

Buffer sizing for Clock Networks

For any clock network

Minimize: power

While maintaining: skew

slew

Mathematical Model

minimize:

power $\Rightarrow$ $$Area(x) = \sum_{b \in B} x_{l_b} x_{w_b}$$

subject to :

skew $\Rightarrow$ $$\max\{d_i(x) - d_j(x)\} \leq t_{skew}, \quad \forall i, j \in S, i \neq j$$

slew $\Rightarrow$ $$slew_k(x) \leq t_{slew}, \quad \forall k \in N$$

buffer sizes $\Rightarrow$

$$\mathrm{x}_{l_b} = l_{\min}, \quad \forall b \in B$$

$$\mathrm{x}_{w_b} \in \{w_{\min}, w_{\min} + \lambda, \ldots\}, \ \forall b \in B$$

How Hard is the Problem?

This problem is a

non-convex

Nonlinear

non-differentiable

with integer variables.

Which means it islike the K2 mountain of optimization.

Discrete Variables

minimize:

$$Area(x) = \sum_{b \in B} x_{l_b} x_{w_b}$$

subject to :

$$\max\{d_i(x) - d_j(x)\} \leq t_{skew}, \quad \forall i, j \in S, i \neq j$$

$$slew_k(x) \leq t_{slew}, \quad \forall k \in N$$

$$\mathrm{x}_{l_b} = l_{\min}, \quad \forall b \in B$$

$$\mathrm{x}_{w_b} \in \{w_{\min}, w_{\min} + \lambda, \ldots\}, \ \forall b \in B$$

Continuous Variables

minimize:

$$Area(x) = \sum_{b \in B} x_{l_b} x_{w_b}$$

subject to :

$$\max\{d_i(x) - d_j(x)\} \le t_{skew}, \quad \forall i, j \in S, i \ne j$$

$$slew_k(x) \le slew_{\max}, \quad \forall k \in N$$

$$\boxed{\begin{array}{l} \mathrm{x}_{l_b} \ge l_{\min}, \quad \forall b \in B \\ \mathrm{x}_{w_b} \ge w_{\min}, \quad \forall b \in B \end{array}}$$

Geometric Programming for Buffer Sizing

Geometric Programming

Monomial: $\alpha x_1^{a_1} x_2^{a_2} x_3^{a_3} \ldots x_n^{a_n}, \alpha \geq 0, x \in R^+$

Posynomial: $\sum Monomials$

Geometric Programming

Geometric Programming Problem:

minimize:

$$f_0(x)$$ ⇐ posynomial

subject to :

$$f_i(x) \le 1, \quad i = 1, \ldots, m$$ ⇐ posynomial

$$h_i(x) = 1, \quad \forall i = 1, \ldots, p$$ ⇐ monomial

Buffer Sizing Problem

minimize:

$$Area(x) = \sum_{b \in B} x_{l_b} x_{w_b}$$ posynomial

subject to :

$$\max\{d_i(x) - d_j(x)\} \le t_{skew}, \quad \forall i, j \in S, i \ne j$$

$$slew_k(x) slew_{\max}^{-1} \le 1, \quad \forall k \in N$$ monomial

$$\mathrm{x}_{l_b} l_{\min}^{-1} \ge 1, \quad \forall b \in B$$ monomial

$$\mathrm{x}_{w_b} w_{\min}^{-1} \ge 1, \ \forall b \in B$$ monomial

Geometric Relaxation

minimize:

$$Area(x) = \sum_{b \in B} x_{l_b} x_{w_b}$$ posynomial

subject to :

$$\boxed{\max\{d_i(x) - d_j(x)\} \leq t_{skew}, \quad \forall i, j \in S, i \neq j}$$

$$slew_k(x) slew_{\max}^{-1} \leq 1, \quad \forall k \in N$$ monomial

$$\mathrm{x}_{l_b} l_{\min}^{-1} \geq 1, \quad \forall b \in B$$ monomial

$$\mathrm{x}_{w_b} w_{\min}^{-1} \geq 1, \ \forall b \in B$$ monomial

Skew Constraint Relaxation

$$\max\{d_i(x) - d_j(x)\} \leq t_{skew}$$

$$\max\{d_i(x)\} \leq t_{skew} + d_j(x)$$

$$\max\{d_i(x)\} \leq t_{skew} + d_{\min}$$

$$d_i(x) \leq \delta, \qquad \delta \leq t_{skew} + d_{\min}$$

monomial

monomial

$$d_i(x)\delta^{-1} \leq 1, \quad \delta(t_{skew} + d_{\min})^{-1} \leq 1$$

GP Formulation

Equivalent geometric program (GP)

minimize:

$$Area(x) = \sum_{b \in B} x_{l_b} x_{w_b}$$

subject to :

$$\max\{d_i(x) - d_j(x)\} \le t_{skew}$$

$$slew_k(x) \le t_{slew}$$

$$x_{l_b} \ge l_{\min}$$

$$x_{w_b} \ge w_{\min}$$

$\Rightarrow$

minimize:

$$Area(x) = \sum_{b \in B} x_{l_b} x_{w_b}$$

subject to :

$$\delta(t_{skew} - d_{\min})^{-1} \le 1$$

$$d_i(x)\delta^{-1} \le 1$$

$$slew_k(x) t_{slew}^{-1} \le 1$$

$$l_{\min} x_{l_b}^{-1} \le 1$$

$$w_{\min} x_{w_b}^{-1} \le 1$$

GP Relaxation

Algorithm: GP Buffer Sizing

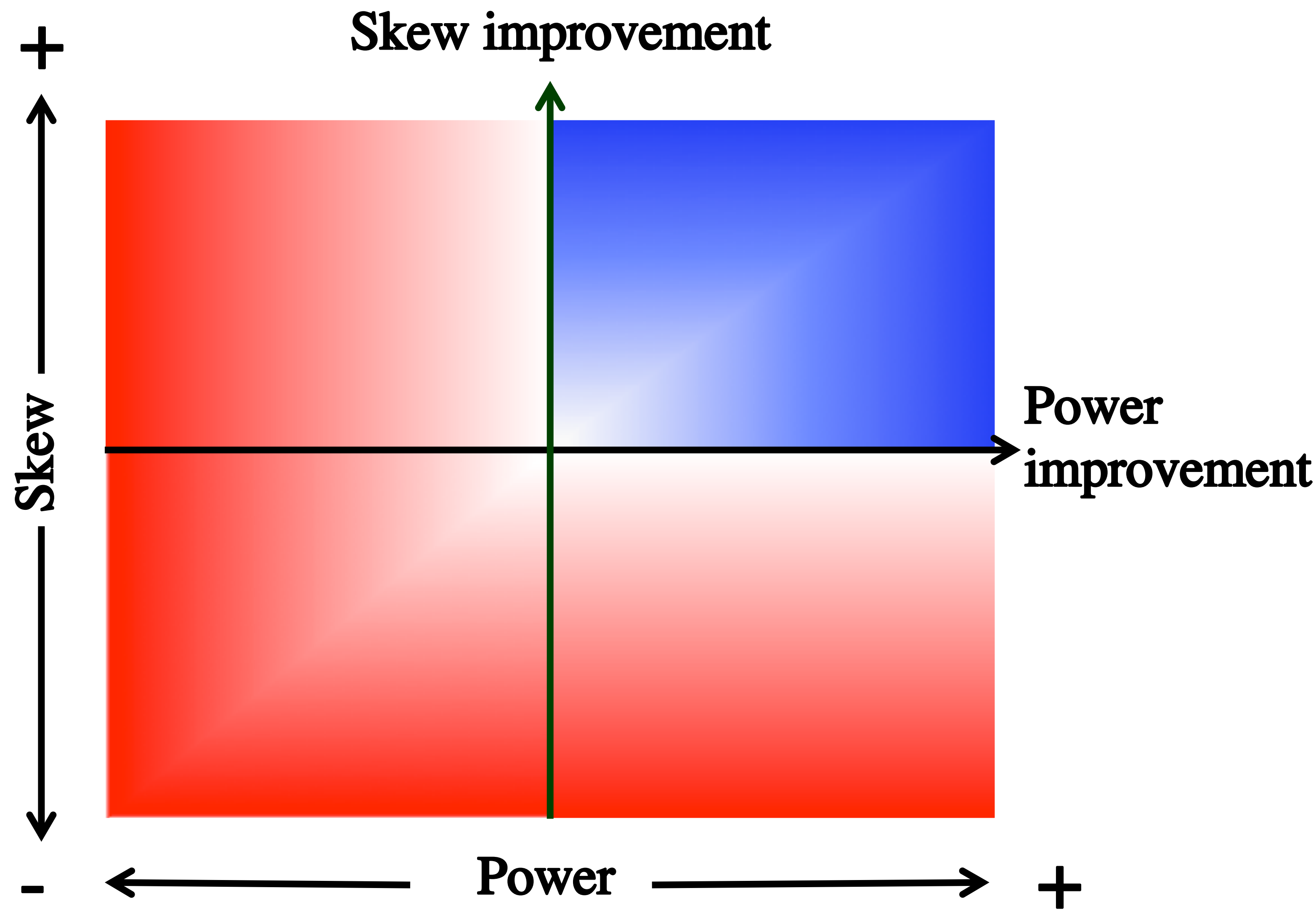
Input: Clock network (ISPD09), t_{skew}, t_{slew}

Output: Optimal buffer sizes

GP Optimal Nominal Sizing

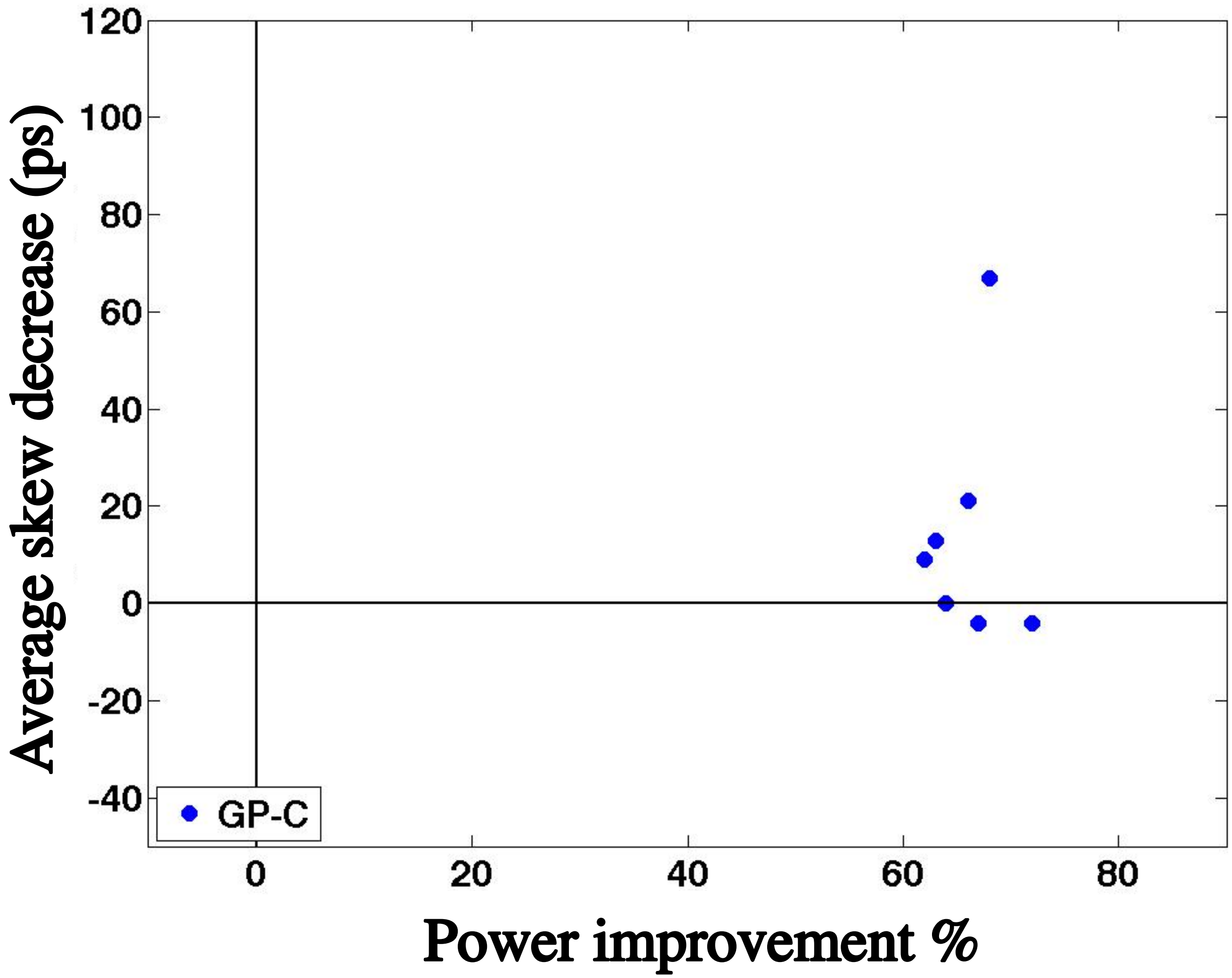
1. Perform GP relaxation
2. Solve GP (MOSEK)
3. Introduce defects (Monte Carlo)
4. Calculate power, slew, and skew (NGSPICE)

Geometric Programming Results



Geometric Programming Results

Power and skew improvement as a result of GP programming. Each circle represent the GPC result for one ISPD09 circuit.



Continuous Back to Discrete

$$\mathrm{x}_{l_b} \geq l_{\min}, \quad \forall b \in B$$

$$\mathrm{x}_{w_b} \geq w_{\min}, \ \forall b \in B$$

$$\Downarrow$$

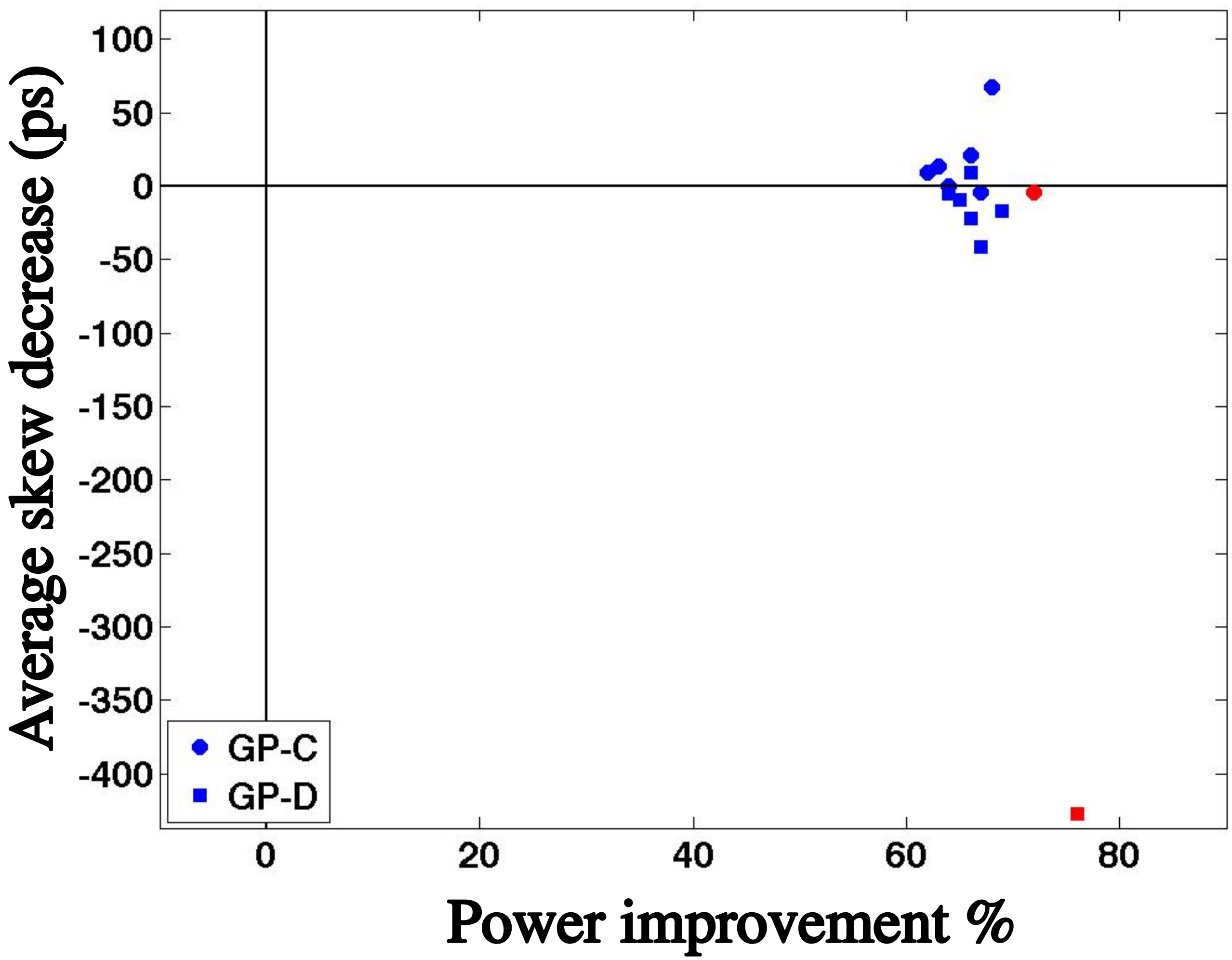
$$x_{lb} = l_{\min}, \ \forall b \in B$$

$$\mathrm{x}_{w_b} \in \{w_{\min}, w_{\min} + \lambda, \ldots\}, \ \forall b \in B$$

Round $\mathrm{x}_{l_b}, \mathrm{x}_{w_b}$ down.

Discretized GP Solution

Power and skew changes as a result of discretization.
Each point represent the result for one ISPD09 circuit.

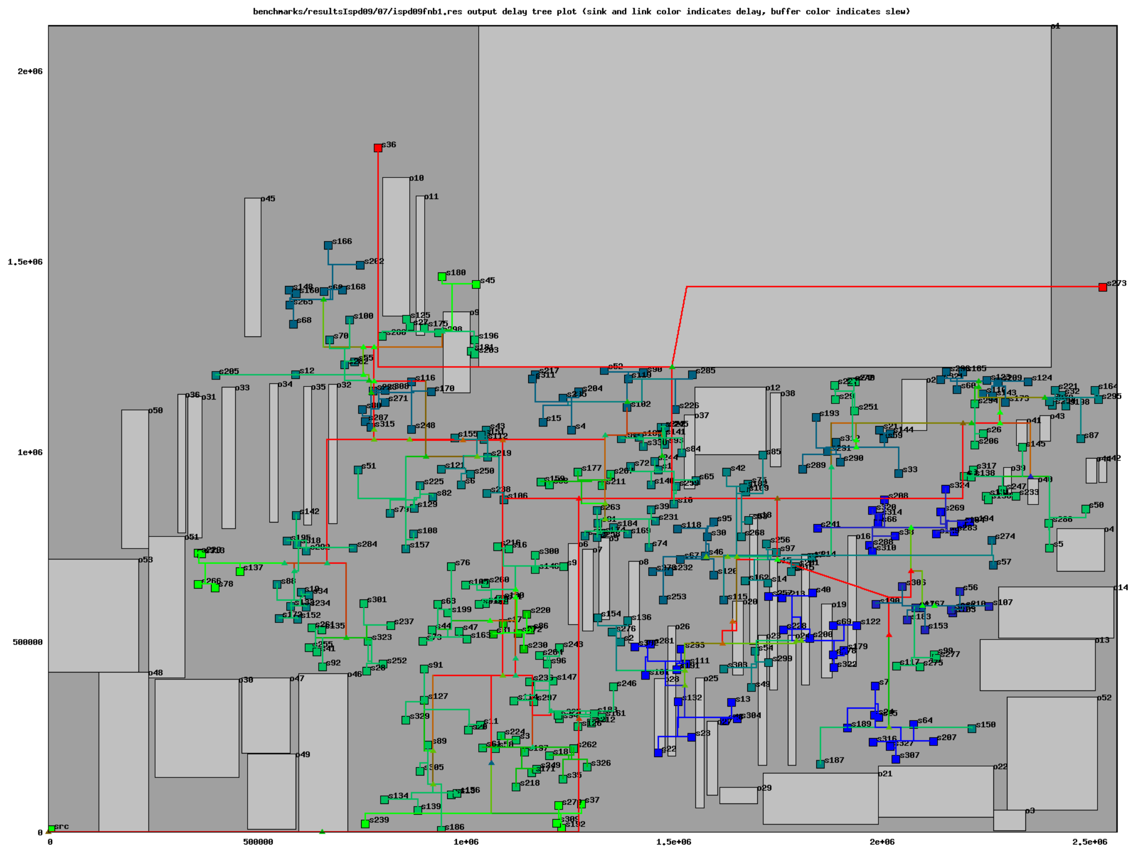


The Outlier

The difficult case: Circuit nb1:

Power improvement: 72%

Skew improvement: -427 ps



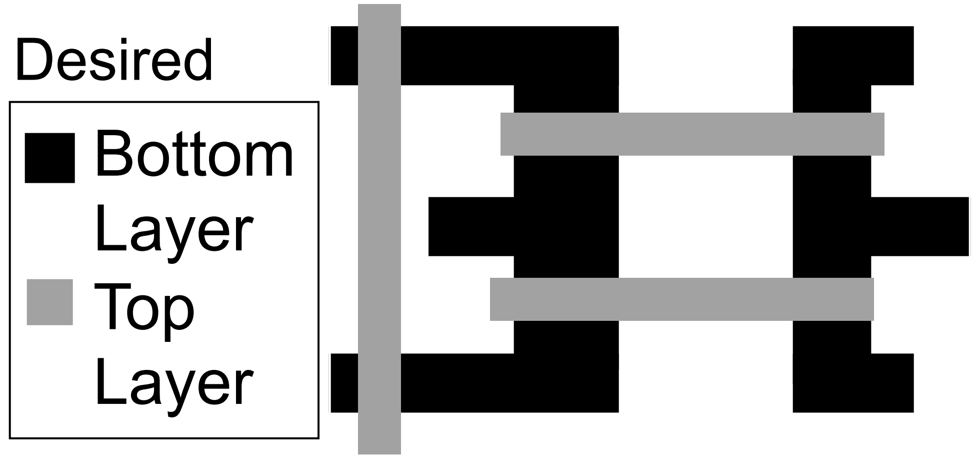
Clock network for Circuit ISPD09-nb1

Robust Geometric Buffer Sizing

“So far as the theories of mathematics are about reality, they are not certain; so far as they are certain, they are not about reality.”
Albert Einstein

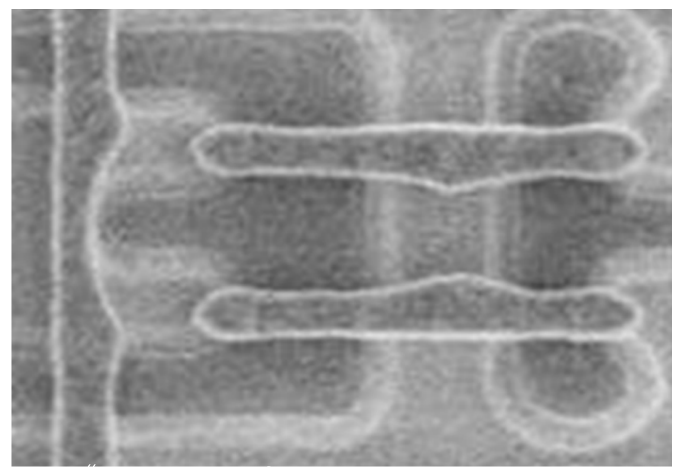
Why Did the Solution Change?

Desired

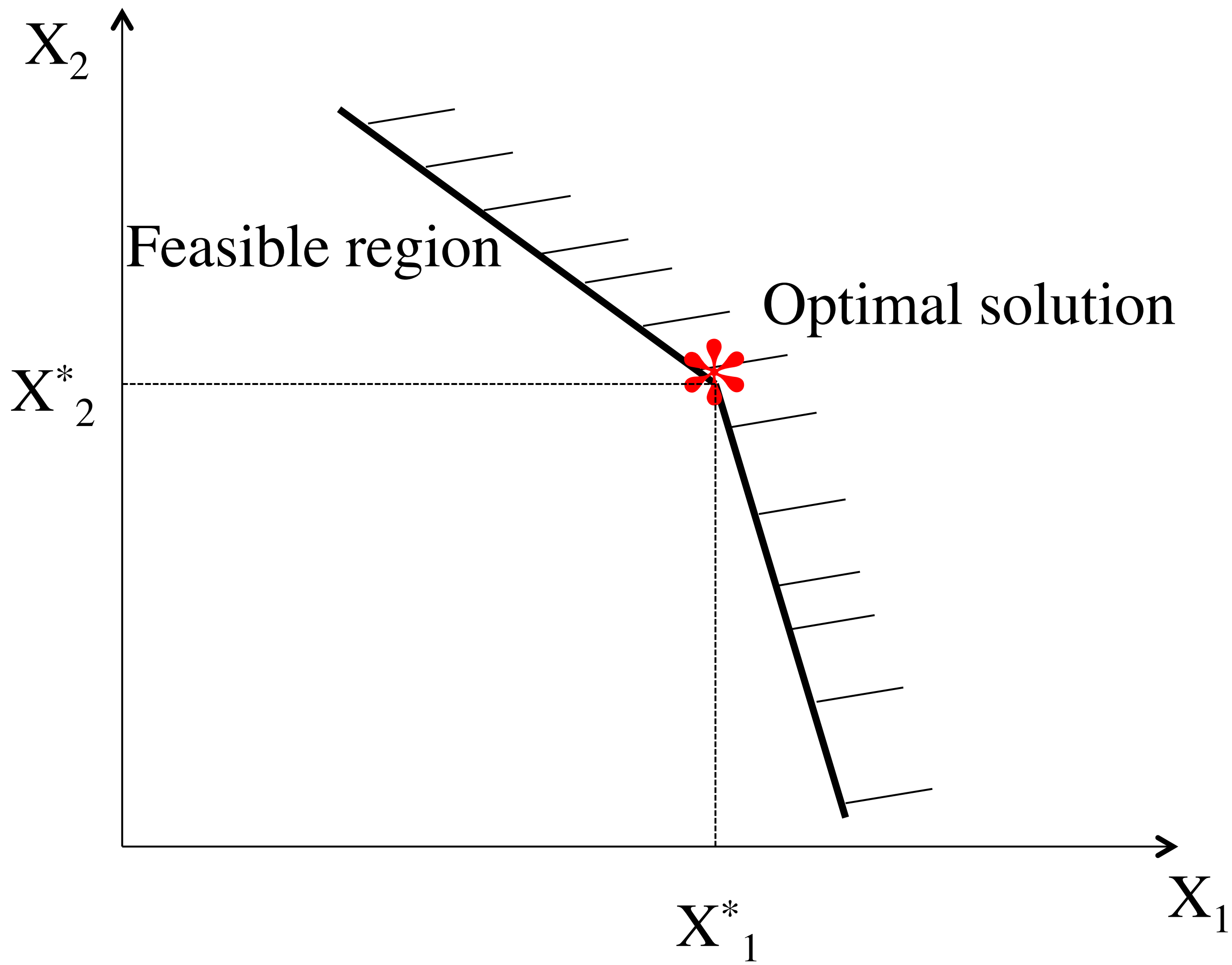


Modern process adds variations

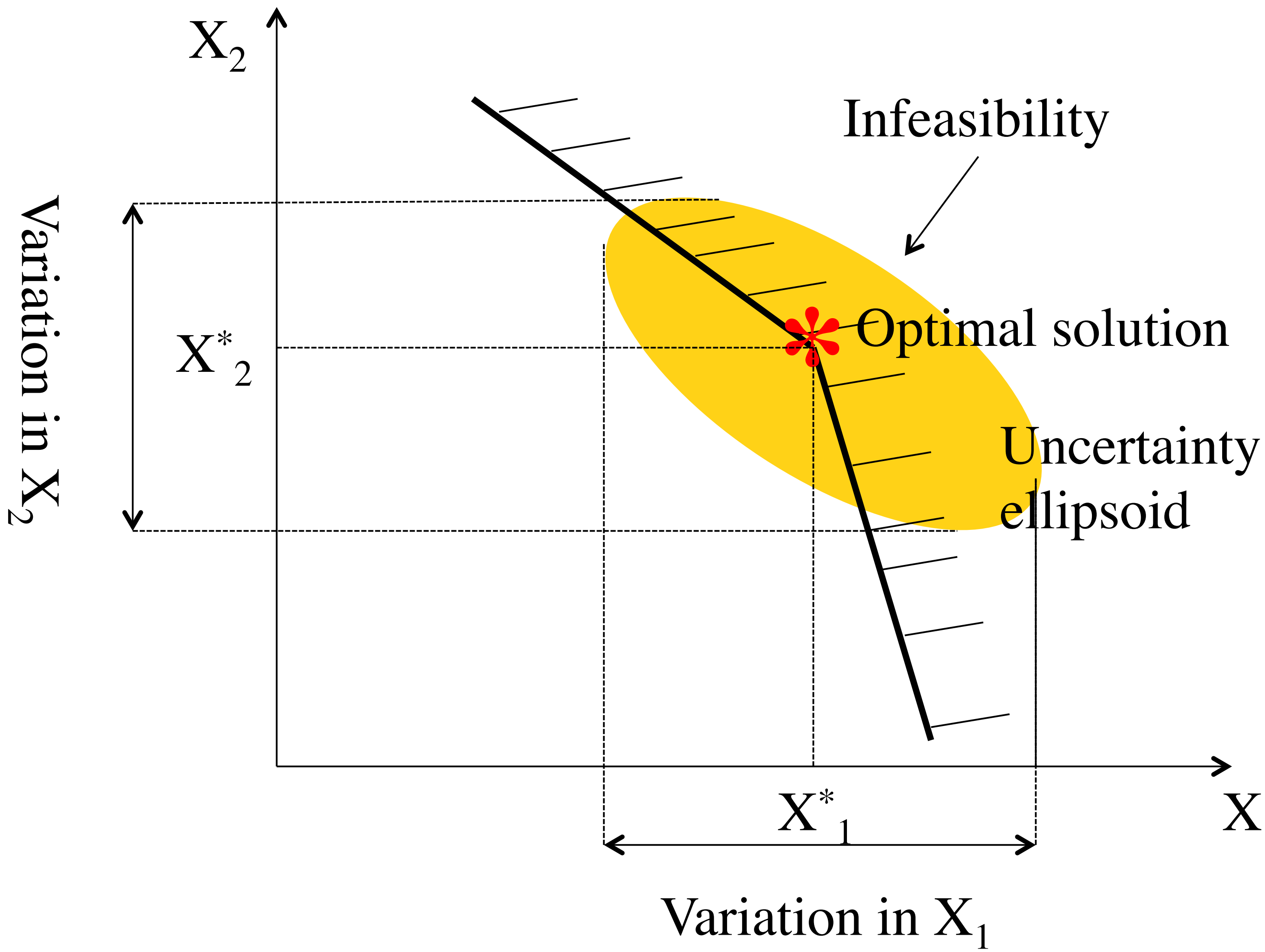
Actual



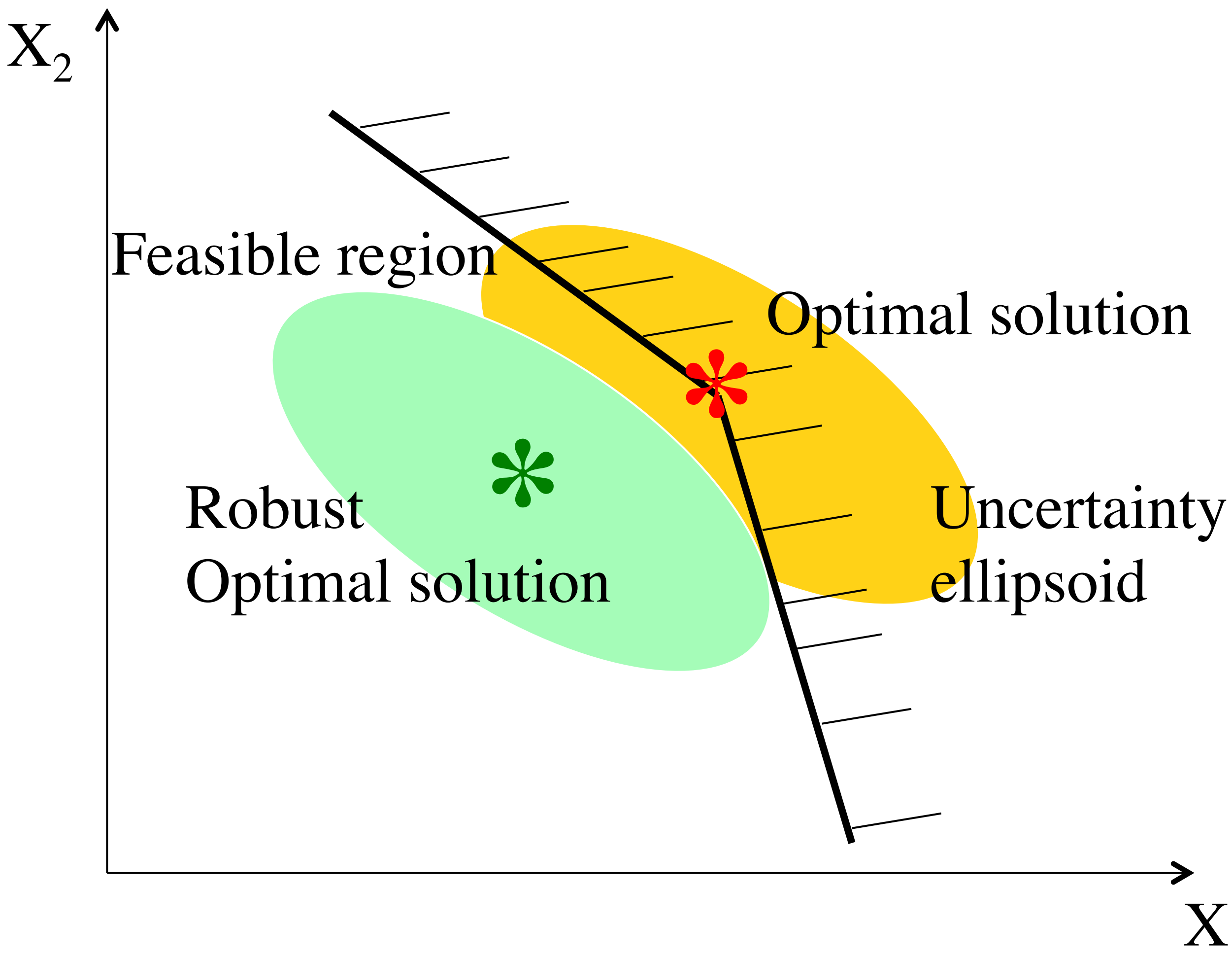
Deterministic Optimization



Effects of Uncertainty



Robust Optimization



Robust Optimization: Buffer Sizing

Each ideal variable x has an associated random variable Δx

Constraints are expanded via Taylor Series

$$f(x) \leq 1$$

$$\Downarrow$$

$$f(x_0 + \Delta x) \approx f(x_0) + \nabla f(x_0)\Delta x \leq 1$$

Consider only the maximum variation

$$f(x_0) + \max_{\forall \Delta x}\{\nabla f(x_0)\Delta x\} \leq 1$$

RGP Algorithm

Algorithm: Robust GP Buffer Sizing

Input: Clock network (ISPD09), t_{skew}, t_{slew}

Output: Optimal buffer sizes

Phase 1. Perform GP sizing

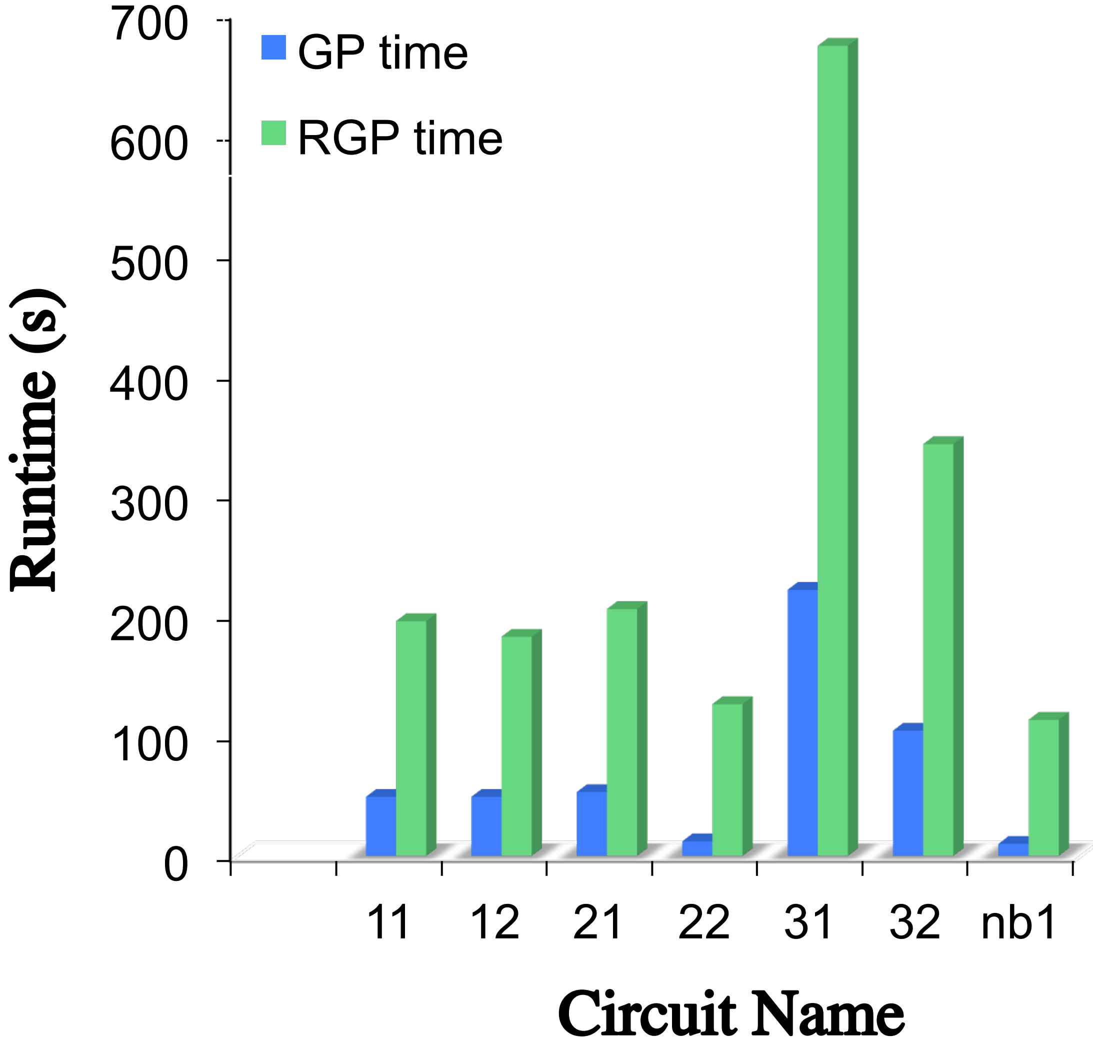
Phase 2. Variation-Aware Sizing

2.a Formulate Robust GP (RGP)

2.b Solve RGP (MOSEK)

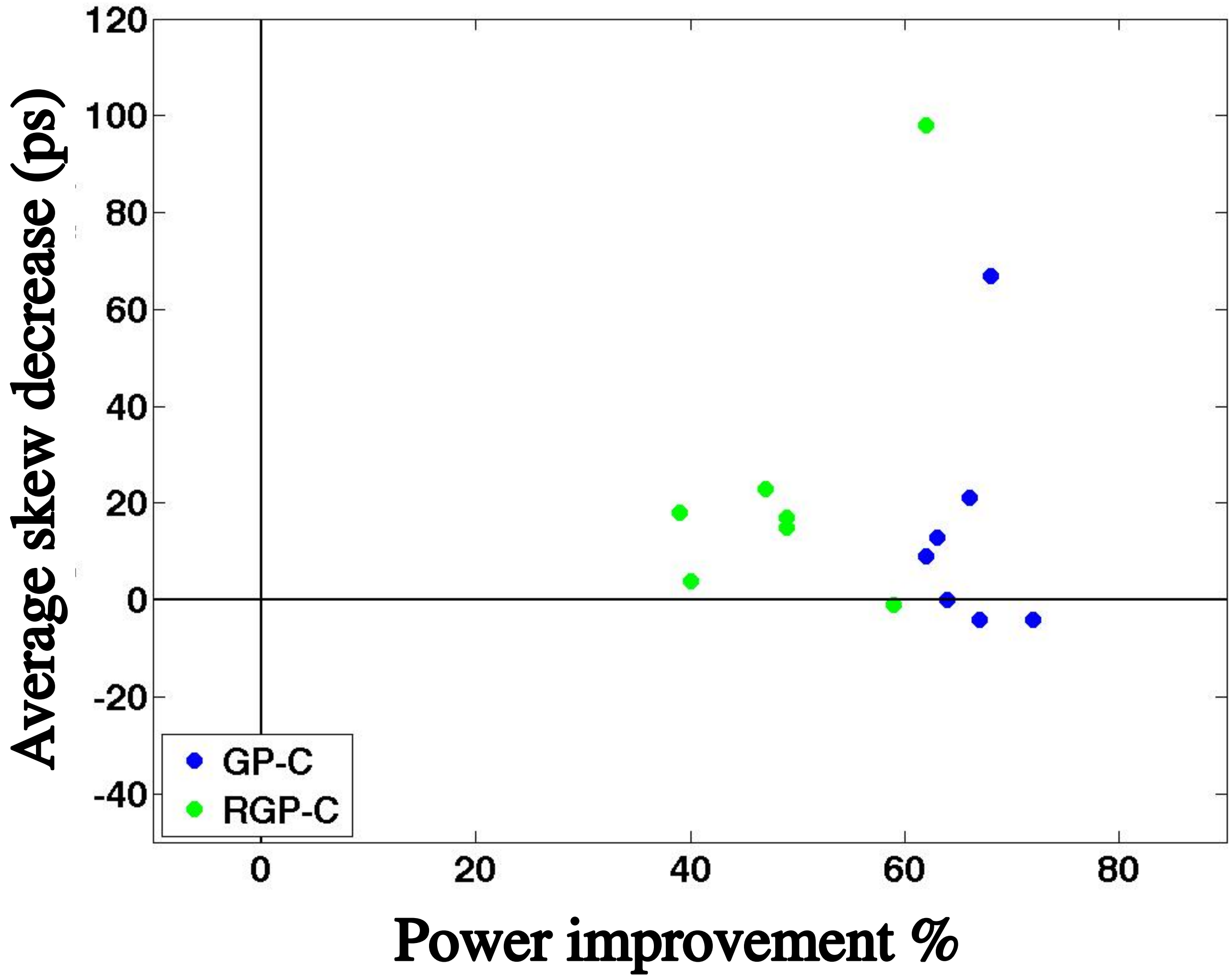
2.c Calculate power, slew, and skew (NGSPICE)

Robust Runtime Results



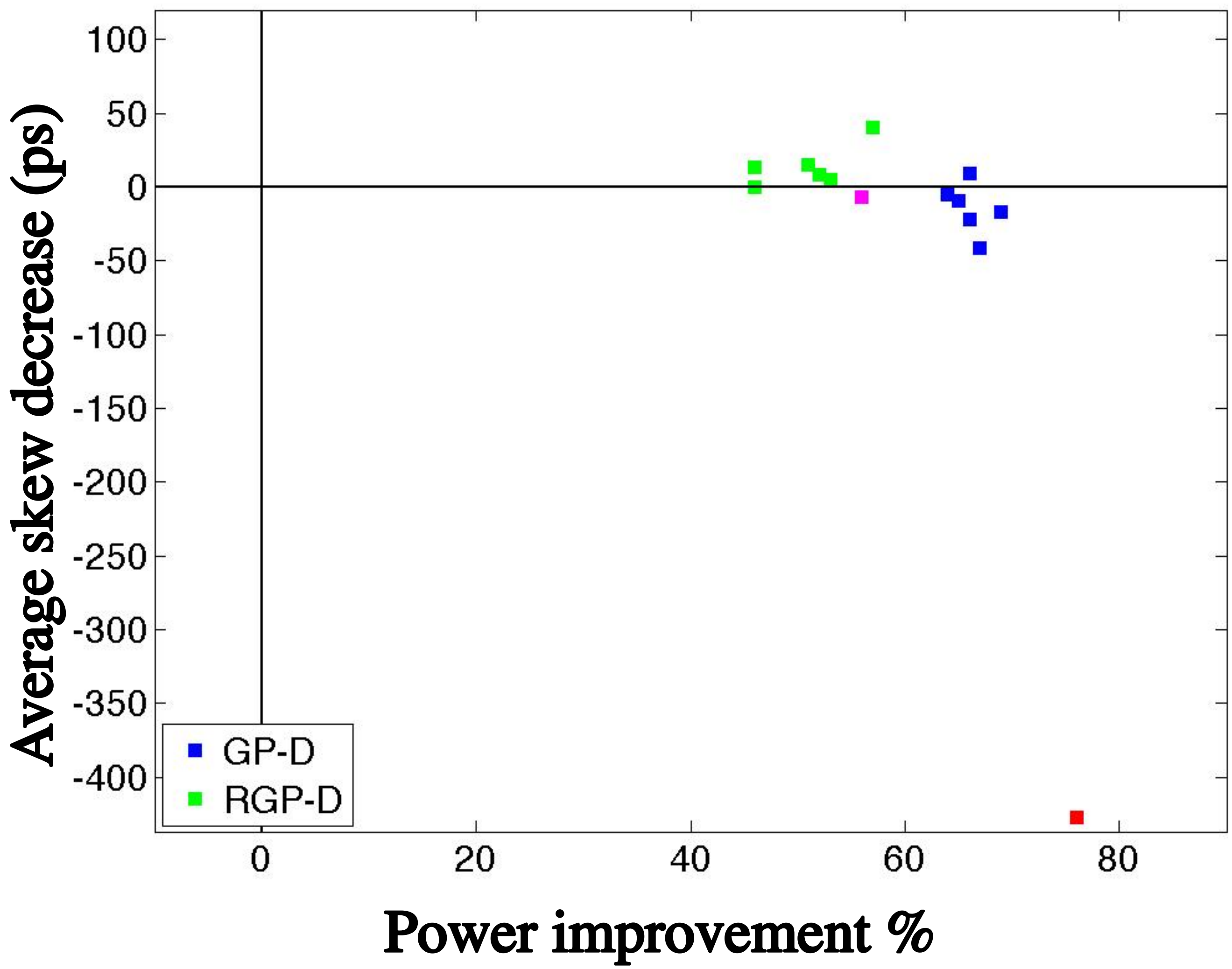
Robust Continuous Results

Comparison of power and skew for continuous GP and RGP.
Each point is power/skew results for one ISPD09 circuit.



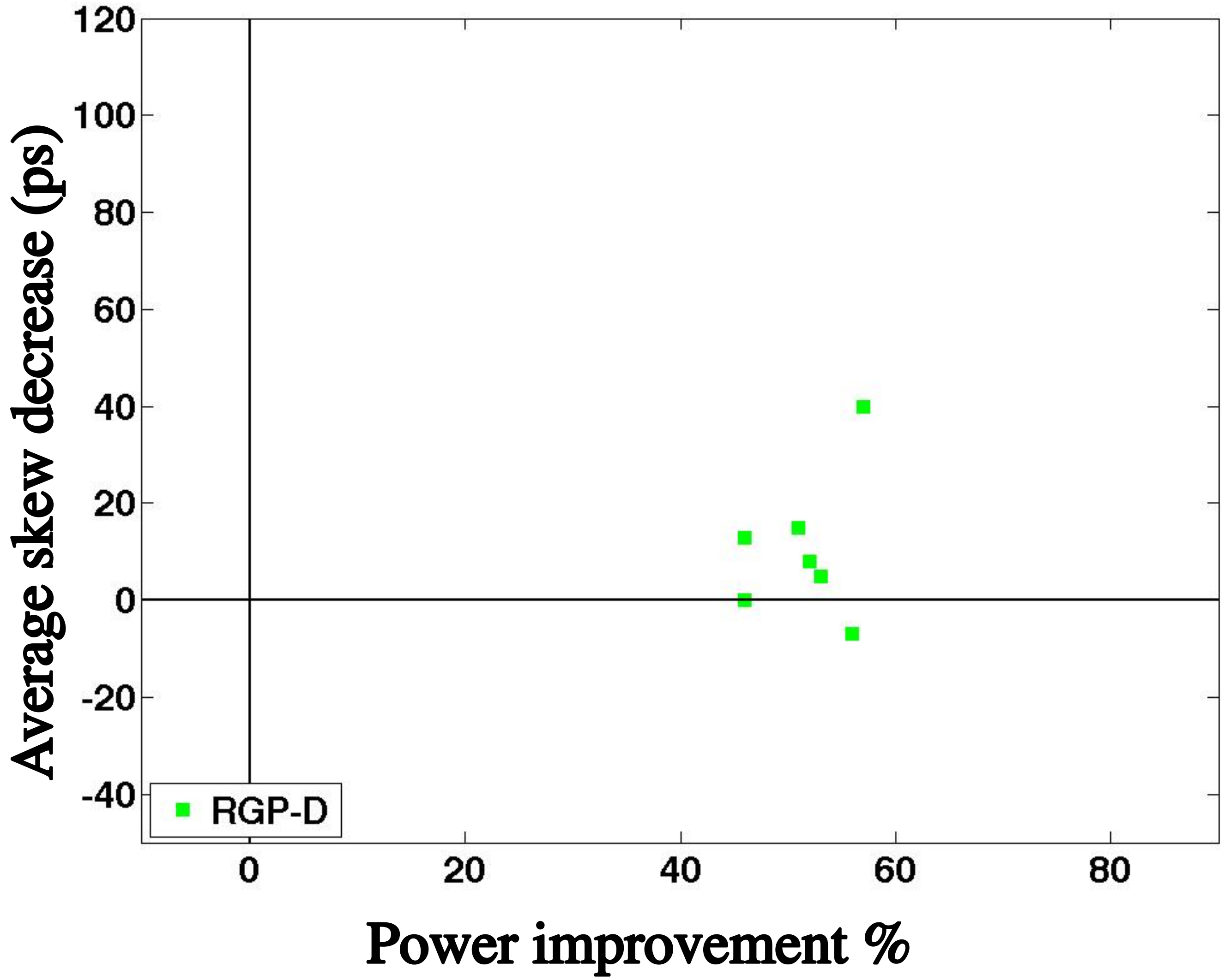
Robust Discrete Results

Comparison of power and skew improvement for discrete GP and RGP. Each point represents the results for one ISPD09 circuit.

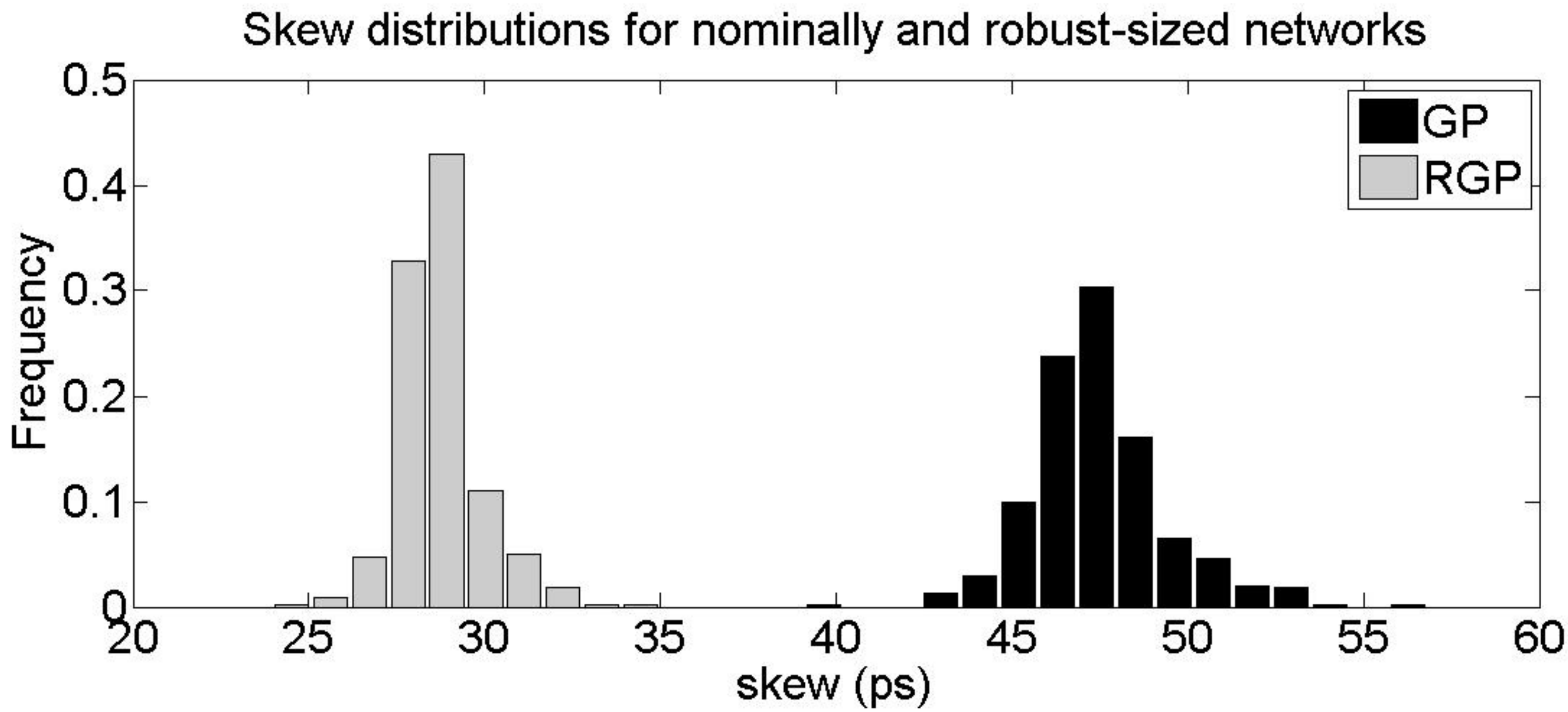


Final Robust Results

Improvement in power and skew for discrete RGP. Each point represents the result for one ISPD09 circuit.



Robust Results



Conclusions and Future Work

Conclusion

New optimization techniques allow us to obtain results that can capture the reality of today’s deep submicron technology.

“In mathematics, you don’t understand things. You just get to use them.”, Johann von Neumann

Future Work

Can we reduce the time for RGP by parallelization?

Can we use mathematical techniques for discretization?

Can we optimize power and skew at the same time?

Buffer sizing for Clock Networks Using Robust Geometric Programming

Laleh Behjat,
Logan Rakai, Amin Farshidi, Dave Westwick
University of Calgary,
Calgary, Canada

March 27, 2013