Probabilistic Evaluation of Solutions in Variability-Driven Optimization

Azadeh Davoodi and Ankur Srivastava
University of Maryland, College Park.

Presenter: Vishal Khandelwal
Outline

• Motivation
  – Challenge in probabilistic optimization considering process variations

• Pruning Probability
  – Metric for comparison of potential solutions

• Computing the Pruning Probability

• Application
  – Dual-Vth assignment considering process variations

• Results
Motivation

• Many VLSI CAD optimization problems rely on comparison of potential solutions
  – To identify the solution with best quality, or to identify a subset of potentially good solutions

• Any potential solution $S_i$ has a corresponding timing $r_i$ & cost $c_i$:
  – e.g., A solution to the gate-sizing problem has:
    • Timing: Delay of the circuit
    • Cost: Overall sizes of the gates
Motivation

• A good solution is the one with better timing and cost

\[ S_i \text{ superior } S_j \iff r_i \leq r_j \text{ and } c_i \leq c_j \]

• Process variations randomize the timing and cost associated with a potential solution

\[ S_i \text{ superior } S_j \iff P(R_i \leq R_j \text{ and } C_i \leq C_j) \approx 1 \]
Pruning Probability

\[ S_i \text{ superior } S_j \iff P(R_i \leq R_j \& C_i \leq C_j) \approx 1 \]

- Let \( C = C_j - C_i \) and \( R = R_j - R_i \)

\[ P(R \geq 0 \& C \geq 0) = \int_0^\infty \int_0^\infty f_{R,C}(r,c)drdc \]

\( f_{R,C} : \) joint probability density function (jpdf) of random variables \( R \) and \( C \)
Computing the Pruning Probability: Challenges

\[ P(R \geq 0 & C \geq 0) = \int_0^\infty \int_0^\infty f_{R,C}(r,c)drdc \]

• Accuracy
  – Might not have an analytical expression for \( f_{R,C} \)
  – Might require numerical methods to compute the probability

• Fast computation
  – Necessary in an optimization framework
  – Makes the use of numerical techniques such as Monte Carlo simulation impractical
Computing the Pruning Probability: Methods

• Based on analytical approximation of the \( jpdf \) \( (f_{R,C}) \)
  – With a well studied \( jpdf \)
  – For which computing the probability integral is analytically possible

• Using Conditional Monte Carlo simulation
  – Bound-based numerical evaluation of the probability
  – Potentially much faster than Monte Carlo
Computing the Pruning Probability:
Approximating \( jpdf \) by Moment Matching

- Approximate \( R,C \) with new random variables \( X,Y \) where the type of \( jpdf \) of \( X,Y \) is known.

- Compute the first few terms of the characteristic functions (Fourier transform) of the two \( jpdfs \) (i.e., moments).

- Match the first few moments and determine the parameters of \( f_{X,Y} \).

- Compute the pruning probability for \( X \) and \( Y \).
Computing the Pruning Probability: Approximating \( jpdf \) by Moment Matching

\[
\Phi_{X,Y}(t_1, t_2) = \int \int e^{i(t_1 x + t_2 y)} f_{X,Y}(x, y) \, dx \, dy
\]

\[
= 1 + it_1 m_{10} + it_2 m_{01} - \frac{t_1^2}{2} m_{20} - \ldots
\]

\[
m_{ij} = \int \int x^i y^j f_{X,Y}(x, y) \, dx \, dy = E[X^i Y^j]
\]
Computing the Pruning Probability: Approximating $pdf$ by Moment Matching

- **Challenges:**
  - Very few bivariate $pdf$s have closed form expressions for their moments
  - Integration of very few known $pdf$s over the quadrant are analytically possible

- Will study the example of bivariate Gaussian approximation given polynomial representation of $R$ and $C$
Example: Bivariate Gaussian \( jpdf \) for Polynomials

Polynomial representation of \( R \) and \( C \) under process variations

- Can represent \( R \) and \( C \) as polynomials
  - By doing Taylor Series expansion of the \( R \) and \( C \) expressions in terms of random variables representing the varying parameters due to process variations (e.g., \( L_{\text{eff}} \), \( T_{\text{ox}} \), etc.)
  - Higher accuracy needs higher order of expansion
  - These r.v.s can be assumed to be independent
    - Using Principal Component Analysis (PCA)

\[
\begin{align*}
R & = f_1(L_{\text{eff}}, T_{\text{ox}}, \ldots) \\
C & = f_2(L_{\text{eff}}, T_{\text{ox}}, \ldots)
\end{align*}
\]

\[
R = \text{Poly}_1(X_1, X_2, \ldots) \\
C = \text{Poly}_2(X_1, X_2, \ldots)
\]

PCA and Taylor Series Expansion
Example: Bivariate Gaussian \( jpdf \) for Polynomials

\[
R = \text{Poly}_1(X_1, X_2, \ldots) \approx r_0 + \sum r_i X_i \quad C = \text{Poly}_2(X_1, X_2, \ldots) \approx c_0 + \sum c_i X_i
\]

- Assuming \( \{X_1, X_2, \ldots\} \) are independent r.v.s with Gaussian density functions
  - The \( jpdf \) \( (f_{R,C}) \) is approximated to be bivariate Gaussian
  - Using linear approximation of \( R \) and \( C \)
    \[
    f_{x,y} = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp[-\frac{z}{2(1-\rho^2)}]
    \]
    \[
    z = \frac{(x_1 - \mu_1)^2}{\sigma_1^2} - \frac{2\rho(x_1 - \mu_1)(x_2 - \mu_2)}{\sigma_1\sigma_2} + \frac{(x_2 - \mu_2)^2}{\sigma_2^2}
    \]
- Moments of bivariate Gaussian \( jpdf \) are related to \( \mu_1, \mu_2, \sigma_1, \sigma_2, \rho \)
  - Need to specify the values of these parameters using moment matching
Example: Bivariate Gaussian \( jpdf \) for Polynomials

\[
f_{x,y} = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{z}{2(1-\rho^2)}\right]
\]

\[
z = \frac{(x_1 - \mu_1)^2}{\sigma_1^2} - \frac{2\rho(x_1 - \mu_1)(x_2 - \mu_2)}{\sigma_1\sigma_2} + \frac{(x_2 - \mu_2)^2}{\sigma_2^2}
\]

\[
R = \text{Poly}_1(X_1, X_2,...) \approx r_0 + \sum r_iX_i
\]

\[
C = \text{Poly}_2(X_1, X_2,...) \approx c_0 + \sum c_iX_i
\]

\[
\rho \sigma_x \sigma_y + \mu_x \mu_y = E[RC]
\]

\[
\mu_2 = E[C] \quad \sigma_1^2 + \mu_1^2 = E[R^2]
\]

\[
\mu_1 = E[R] \quad \sigma_2^2 + \mu_2^2 = E[C^2]
\]

Analytical expression for probability integral of bivariate Gaussian \( jpdf \) is available (Hermite Polynomials)*

*[Vasicek 1998]
Computing the Pruning Probability: Conditional Monte Carlo (CMC)

\[ P(R \geq 0 \& C \geq 0) = \int_0^\infty \int_0^\infty f_{R,C}(r,c)drdc \]

- CMC is similar to MC but:
  - Uses simple bounds that can evaluate the sign of R and C for most of the MC samples
    - *Evaluation of simple bounds are much more efficient than polynomial expressions that are potentially of high order*
  - Only in the cases that the simple bounds can not predict the sign of R and C, the complicated polynomial expressions are evaluated
Computing the Pruning Probability: Conditional Monte Carlo (CMC)

2. Generate Samples Based on pdf of the Xi variables
3. Evaluate $R^L < 0, R^U > 0$ or $C^L < 0, C^U > 0$
4. Determine if $R > 0$ & $C > 0$ by evaluating $R$ and $C$
5. Update count of # $R & C > 0$
6. Pruning Probability: $\frac{\#(R \geq 0, C \geq 0)}{\#total}$

- Can NOT predict sign of $R$ and $C$ from its bounds
- Can predict sign of $R$ and $C$ from its bounds

Decision Tree:
- Y: Can predict sign of $R$ and $C$ from its bounds
- N: Can NOT predict sign of $R$ and $C$ from its bounds
Computing the Pruning Probability: Conditional Monte Carlo (CMC)

• Accurately predicts the probability value

• Speedup is due to the following intuition:
  - Evaluation of simple bounds are much faster than high-order polynomials
  - If the bounds are accurate, they predict the sign of the polynomials very frequently resulting in significant speedup
Example: Computing Bounds on Polynomials

\[ Poly(x_1, \ldots, x_n) = \sum_{i_1=0}^{l_1} \ldots \sum_{i_n=0}^{l_n} a_{i_1, \ldots, i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \]

- Bernstein coefficients define convex hull for any polynomial*

\[
\left( \frac{i_1}{l_1}, \ldots, \frac{i_n}{l_n}; b_{i_1, \ldots, i_n} \right) \quad \text{where} \quad b_{i_1, \ldots, i_n} = \sum_{j_1=0}^{i_1} \ldots \sum_{j_n=0}^{i_n} \frac{j_1}{l_1} \ldots \frac{j_n}{l_n} a_{i_1, \ldots, i_n}
\]

\[ P(x) = 1 + 0.5x - 0.33x^2 + 0.25x^3 \]

*[Cargo, Shisha 1966]*
Example: Computing Bounds on Polynomials

• Simple hyper-plane lower-bounds are defined for each polynomial from its Bernstein coefficients*

*Garloff, Jansson, Smith 2003

\[ P(x) = 1 + 0.5x - 0.33x^2 + 0.25x^3 \]
Application:

Dual-V\textsubscript{th} Assignment for Leakage Optimization Under Process Variations

- Assignment of either high or low threshold voltage to gates in a circuit (represented as nodes in a graph)
  - Higher threshold (slow), lower threshold (leaky)
- Under process variations the goal is:
  - To minimize expected value of overall leakage (E[L])
  - Subject to bounding the maximum probability of violating a Timing Constraint (T\textsubscript{cons}) at the Primary Output
Dual-\(V_{th}\) Assignment for Leakage Optimization Under Process Variations

- **Dynamic programming based formulation**
  - Topological traversal from PIs to POs
  - Solution at a node:
    - \(V_{th}\) assignment to sub-tree rooted at the node
  - Solution set at each node:
    - Generated by combining solutions of a node’s children + node’s \(V_{th}\) possibilities
    - Pareto-optimal set identified & stored*

- **Each solution:**
  - Overall leakage at the node’s subtree: \(L_k = l_0^{(k)} + \sum l_i^{(k)} X_i + \sum \sum l_{ij}^{(k)} X_i X_j + \ldots\)
  - Arrival time of the node’s subtree: Approximated as a linear combination of parameters

* [Davoodi, Srivastava ISLPED05]
%Error in Estimation of Pruning Probability

For 2600 solution pairs from the dual-Vth framework
Speedup in Computing the Pruning Probability

Bivariate Normal Approximation
Conditional Monte Carlo
Comparing Quality of Solution in Dual-Vth Assignment

<table>
<thead>
<tr>
<th>$T_{cons}$ (nsec)</th>
<th>$E[I]$</th>
<th>$P_v(T)$</th>
<th>t</th>
<th>$%_{imp}$</th>
<th>$P_v(T)$</th>
<th>t</th>
<th>$%_{imp}$</th>
<th>$P_v(T)$</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>C432</td>
<td>33.0</td>
<td>10634</td>
<td>0.11</td>
<td>10</td>
<td>35</td>
<td>0.27</td>
<td>13</td>
<td>46</td>
<td>0.25</td>
</tr>
<tr>
<td>C499</td>
<td>17.5</td>
<td>14285</td>
<td>0.14</td>
<td>29</td>
<td>15</td>
<td>0.17</td>
<td>51</td>
<td>22</td>
<td>0.14</td>
</tr>
<tr>
<td>C880</td>
<td>32.0</td>
<td>16650</td>
<td>0.11</td>
<td>12</td>
<td>39</td>
<td>0.29</td>
<td>14</td>
<td>49</td>
<td>0.30</td>
</tr>
<tr>
<td>C1355</td>
<td>18.0</td>
<td>17182</td>
<td>0.08</td>
<td>40</td>
<td>29</td>
<td>0.11</td>
<td>50</td>
<td>36</td>
<td>0.09</td>
</tr>
<tr>
<td>C1908</td>
<td>29.0</td>
<td>13768</td>
<td>0.13</td>
<td>37</td>
<td>33</td>
<td>0.18</td>
<td>40</td>
<td>39</td>
<td>0.16</td>
</tr>
<tr>
<td>C3540</td>
<td>42.0</td>
<td>38561</td>
<td>0.18</td>
<td>123</td>
<td>23</td>
<td>0.23</td>
<td>181</td>
<td>42</td>
<td>0.22</td>
</tr>
<tr>
<td>C5315</td>
<td>31.0</td>
<td>42032</td>
<td>0.12</td>
<td>160</td>
<td>5</td>
<td>0.13</td>
<td>173</td>
<td>36</td>
<td>0.16</td>
</tr>
<tr>
<td>C6288</td>
<td>110.0</td>
<td>45343</td>
<td>0.19</td>
<td>1131</td>
<td>2</td>
<td>0.19</td>
<td>1699</td>
<td>2</td>
<td>0.19</td>
</tr>
<tr>
<td>alu2</td>
<td>8.0</td>
<td>13340</td>
<td>0.03</td>
<td>13</td>
<td>23</td>
<td>0.04</td>
<td>20</td>
<td>35</td>
<td>0.03</td>
</tr>
<tr>
<td>alu4</td>
<td>12.0</td>
<td>23317</td>
<td>0.06</td>
<td>65</td>
<td>16</td>
<td>0.07</td>
<td>70</td>
<td>40</td>
<td>0.07</td>
</tr>
<tr>
<td>dalu</td>
<td>27.0</td>
<td>35812</td>
<td>0.12</td>
<td>68</td>
<td>21</td>
<td>0.15</td>
<td>104</td>
<td>39</td>
<td>0.17</td>
</tr>
<tr>
<td>Ave.</td>
<td></td>
<td>0.12</td>
<td></td>
<td>21.9</td>
<td>0.17</td>
<td>35.1</td>
<td>0.16</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Run Time (sec)
E[I] in pA

Maximum allowed risk (probability) for violating the timing constraint: 0.3
Conclusions

• Introduced pruning probability as metric to compare potential solutions in a variability-driven optimization framework

• Illustrated computing of pruning probability:
  – Using efficient \( jpdf \) approximation
  – Using accurate Conditional Monte Carlo simulation
  – Both methods significantly faster the MC
Thank You For Your Attention!

For details please contact the authors:
{azade,ankurs}@eng.umd.edu