Layout Decomposition via Boolean Satisfiability

Hongduo Liu\textsuperscript{1}, Peiyu Liao\textsuperscript{1}, Mengchuan Zou\textsuperscript{2}, Bowen Pang\textsuperscript{2}, Xijun Li\textsuperscript{2}, Mingxuan Yuan\textsuperscript{2}, Tsung-Yi Ho\textsuperscript{1}, Bei Yu\textsuperscript{1}

\textsuperscript{1}Chinese University of Hong Kong
\textsuperscript{2}Huawei Noah’s Ark Lab
Outline

1. Introduction
2. SAT-based Layout Decomposer
3. Layout Decomposition as Bilevel Optimization
4. Experimental Results
• A gap between lithography resolution and advanced technology nodes.
• Multiple Patterning Lithography can enhance the feature density.
Problem Formulation

- Layout Decomposition: Decompose one layout onto multiple masks for better manufacturability.
- Layout decomposition can be formulated as graph coloring. The coloring result should minimize the weighted sum of conflict cost and stitch cost.

Figure: Dashed edges are stitch edges, and real lines are conflict edges.
Literature Review

- Exact Algorithm: Integer Linear Programming

- Approximation Algorithm:
  - Semidefinite Programming
  - Linear Programming
  - Heuristic methods

---

Motivation

Two important observations:

- Boolean nature of decision variables in ILP formulation $\Rightarrow$ Boolean satisfiability $\Rightarrow$ Faster convergence.

- Conflict optimization and stitch minimization are two problems nested with each other $\Rightarrow$ Bilevel Reformulation $\Rightarrow$ Tighter Approximation.
A propositional logic formula is said to be in Conjunctive Normal Form (CNF) if it is a conjunction (“and”) of disjunctions (“ors”) of literals.

A literal is either a boolean variable $x$ or its negation $\neg x$.

For example, $(p \lor q) \land (\neg p \lor \neg q)$ is a CNF, where $p$, $q$, $\neg p$, $\neg q$ are all literals. The disjunctions $(p \lor q)$ and $(\neg p \lor \neg q)$ are also called clauses.

The satisfiability (SAT) problem is to find a satisfying assignment to the boolean variables such that the CNF formula yields true.
ILP Formulation for Triple Patterning

\[
\begin{align*}
\min & \quad \sum_{r_i \in p_m, r_j \in p_n, c_{ij} \in CE} C_{mn} + \alpha \sum_{s_{ij} \in SE} s_{ij}, \\
\text{s.t.} & \quad x_{i1} + x_{i2} \leq 1, \quad \forall i \in V, \quad (1a) \\
& \quad x_{i1} + x_{i2} + x_{j1} + x_{j2} + C_{mn} \geq 1, \quad \forall c_{ij} \in CE, r_i \in p_m, r_j \in p_n, \quad (1b) \\
& \quad x_{i1} - x_{i2} + x_{j1} - x_{j2} - C_{mn} \leq 1, \quad \forall c_{ij} \in CE, r_i \in p_m, r_j \in p_n, \quad (1c) \\
& \quad -x_{i1} + x_{i2} - x_{j1} + x_{j2} - C_{mn} \leq 1, \quad \forall c_{ij} \in CE, r_i \in p_m, r_j \in p_n, \quad (1d) \\
& \quad x_{i1} + x_{i2} + x_{j1} + x_{j2} - C_{mn} \leq 1, \quad \forall c_{ij} \in CE, r_i \in p_m, r_j \in p_n, \quad (1e) \\
& \quad x_{i1} + x_{i2} + x_{j1} + x_{j2} - C_{mn} \leq 3, \quad \forall c_{ij} \in CE, r_i \in p_m, r_j \in p_n, \quad (1f) \\
& \quad x_{i1} - x_{j1} + s_{ij} \geq 0, \quad \forall e_{ij} \in SE, \quad (1g) \\
& \quad x_{i1} - x_{j1} + s_{ij} \leq 0, \quad \forall e_{ij} \in SE, \quad (1h) \\
& \quad x_{i2} - x_{j2} + s_{ij} \geq 0, \quad \forall e_{ij} \in SE, \quad (1i) \\
& \quad x_{i2} - x_{j2} + s_{ij} \leq 0, \quad \forall e_{ij} \in SE, \quad (1j) \\
& \quad \text{All decision variables are binary.} \quad (1k)
\end{align*}
\]
Overall Flow

• SAT indicates that a better solution has been found.
• UNSAT means the previous satisfiable solution is the optimal solution.
A Toy Example
Constraint \( x_1 + x_2 + \ldots + x_k \geq 1 \) is equal to a CNF clause \((x_1 \lor x_2 \lor \ldots \lor x_k)\).

\( x_{i1} + x_{i2} \leq 1 \) can be transformed into a CNF clause through the following steps:

- Let the \( \leq \) be \( \geq \) by multiplying \(-1\) on both sides of the inequality. We have \(-x_{i1} - x_{j1} \geq -1\).
- Replace \( x_{i1}, x_{j1} \) by \(-(1 - \overline{x_{i1}}), -(1 - \overline{x_{j1}})\) respectively. We can get \(- (1 - \overline{x_{i1}}) - (1 - \overline{x_{j1}}) \geq -1\). Here \( \overline{x} \) is the negation of \( x \), and it is easy to see \( \overline{\overline{x}} = x \).
- Reorganize the terms we have \( \overline{x_{i1}} + \overline{x_{j1}} \geq 1 \), which can be represented by a CNF clause \((\overline{x_{i1}} \lor \overline{x_{j1}})\).
Consider constraint $5x + 2y + 4z \leq 5$.

- Construct a Binary Decision Diagram.
- Extract all path to false.
  - $x \xrightarrow{1} y \xrightarrow{1} \text{false}$ derives clause $\neg x \vee \neg y$.
  - $x \xrightarrow{0} y \xrightarrow{1} z \xrightarrow{1} \text{false}$ derives clause $x \vee \neg y \vee \neg z$. 

<table>
<thead>
<tr>
<th>Objective Bound to Clause</th>
</tr>
</thead>
<tbody>
<tr>
<td>$13/23$</td>
</tr>
</tbody>
</table>
The layout decomposition problem can also be formulated as a bilevel optimization problem. The upper-level optimization problem is given by

$$\min_{C,s} \sum_{r_i \in p_m, r_j \in p_n, c_{ij} \in CE} C_{mn} + \alpha \sum_{s_{ij} \in SE} s_{ij},$$

s.t. constraint (1b) – constraint (1f),

$$s \in S(C),$$

where $S(C)$ is the set of optimal solutions of the $C$-parameterized problem

$$\min_{s} \sum_{s_{ij} \in SE} s_{ij},$$

s.t. constraint (1b) – constraint (1j).
Approximation Algorithm

- Lower level decision space
- Upper level decision space
- Conflict variable
- Optimal stitch
- Variable solution
- Stitch minimization
How to solve the bilevel optimization problem?

- Single level reduction: the reduced single-level problem is shown exactly as the original ILP formulation.
- Nested optimization: solves the lower-level optimization problem corresponding to every upper-level member until convergence.

Our approximation algorithm:

- Get the assignments of upper-level variables by solving the upper-level problem ignoring the lower-level variables (**Conflict Minimization**).
- Solve the lower-level problem with fixed conflict variables obtained from the previous step (**Stitch Minimization**).
Evaluation of Our Exact Algorithm

Table: Results on ISCAS benchmarks. “RT” indicates runtime.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C432</td>
<td>0.4</td>
<td>0.087</td>
<td>0.4</td>
<td>0.027</td>
<td>0.4</td>
<td>0.021</td>
<td>0.4</td>
<td>0.029</td>
</tr>
<tr>
<td>C499</td>
<td>0.0</td>
<td>0.081</td>
<td>0.0</td>
<td>0.028</td>
<td>0.0</td>
<td>0.025</td>
<td>0.0</td>
<td>0.030</td>
</tr>
<tr>
<td>C880</td>
<td>0.7</td>
<td>0.083</td>
<td>0.8</td>
<td>0.032</td>
<td>0.7</td>
<td>0.026</td>
<td>0.7</td>
<td>0.034</td>
</tr>
<tr>
<td>C1355</td>
<td>0.3</td>
<td>0.062</td>
<td>0.3</td>
<td>0.039</td>
<td>0.3</td>
<td>0.036</td>
<td>0.3</td>
<td>0.044</td>
</tr>
<tr>
<td>C1908</td>
<td>0.1</td>
<td>0.063</td>
<td>0.1</td>
<td>0.054</td>
<td>0.1</td>
<td>0.051</td>
<td>0.1</td>
<td>0.056</td>
</tr>
<tr>
<td>C2670</td>
<td>0.6</td>
<td>0.109</td>
<td>0.6</td>
<td>0.084</td>
<td>0.6</td>
<td>0.079</td>
<td>0.6</td>
<td>0.090</td>
</tr>
<tr>
<td>C3540</td>
<td>1.8</td>
<td>0.153</td>
<td>1.8</td>
<td>0.112</td>
<td>1.8</td>
<td>0.100</td>
<td>1.8</td>
<td>0.123</td>
</tr>
<tr>
<td>C5315</td>
<td>0.9</td>
<td>0.217</td>
<td>0.9</td>
<td>0.147</td>
<td>0.9</td>
<td>0.130</td>
<td>0.9</td>
<td>0.156</td>
</tr>
<tr>
<td>C6288</td>
<td>21.4</td>
<td>2.999</td>
<td>27.3</td>
<td>0.434</td>
<td>21.4</td>
<td>0.300</td>
<td>21.4</td>
<td>0.606</td>
</tr>
<tr>
<td>C7552</td>
<td>2.3</td>
<td>0.402</td>
<td>2.3</td>
<td>0.235</td>
<td>3.1</td>
<td>0.208</td>
<td>2.3</td>
<td>0.255</td>
</tr>
<tr>
<td>S1488</td>
<td>0.2</td>
<td>0.082</td>
<td>0.2</td>
<td>0.051</td>
<td>0.2</td>
<td>0.043</td>
<td>0.2</td>
<td>0.057</td>
</tr>
<tr>
<td>S38417</td>
<td>24.4</td>
<td>2.352</td>
<td>31.6</td>
<td>1.445</td>
<td>24.4</td>
<td>0.771</td>
<td>24.4</td>
<td>2.072</td>
</tr>
<tr>
<td>S35932</td>
<td>48.0</td>
<td>6.451</td>
<td>66.0</td>
<td>4.248</td>
<td>48.7</td>
<td>2.034</td>
<td>48.0</td>
<td>6.069</td>
</tr>
<tr>
<td>S38584</td>
<td>47.6</td>
<td>6.533</td>
<td>58.5</td>
<td>4.195</td>
<td>47.7</td>
<td>2.216</td>
<td>47.6</td>
<td>5.915</td>
</tr>
<tr>
<td>S15850</td>
<td>43.7</td>
<td>5.854</td>
<td>56.3</td>
<td>3.821</td>
<td>44.0</td>
<td>2.075</td>
<td>43.7</td>
<td>5.415</td>
</tr>
<tr>
<td>Avg. Ratio</td>
<td><strong>1.00</strong></td>
<td>1.79</td>
<td>1.11</td>
<td>0.85</td>
<td>1.02</td>
<td>0.67</td>
<td><strong>1.00</strong></td>
<td>1.00</td>
</tr>
</tbody>
</table>
### Table: Layout decomposition results on ISPD19 benchmarks. “RT” indicates runtime.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost</td>
<td>RT (s)</td>
<td>Cost</td>
<td>RT (s)</td>
</tr>
<tr>
<td>test1_100</td>
<td>242.9</td>
<td>56.24</td>
<td>297.7</td>
<td>2.61</td>
</tr>
<tr>
<td>test5_101</td>
<td>452.0</td>
<td>78.32</td>
<td>549.8</td>
<td>5.60</td>
</tr>
<tr>
<td>test6_102</td>
<td>153.4</td>
<td>188.56</td>
<td>191.7</td>
<td>35.58</td>
</tr>
<tr>
<td>test8_100</td>
<td>6005.9</td>
<td>82.13</td>
<td>6206.2</td>
<td>32.27</td>
</tr>
<tr>
<td>test9_100</td>
<td>9223.3</td>
<td>128.91</td>
<td>9532.4</td>
<td>52.72</td>
</tr>
<tr>
<td>test10_100</td>
<td>10 449.5</td>
<td>244.93</td>
<td>10 910.1</td>
<td>85.52</td>
</tr>
<tr>
<td>Avg. Ratio</td>
<td>1.00</td>
<td>4.43</td>
<td>1.13</td>
<td>0.67</td>
</tr>
<tr>
<td>test1_101*</td>
<td>71.8</td>
<td>2370.45</td>
<td>107.4</td>
<td>19.65</td>
</tr>
<tr>
<td>test2_100*</td>
<td>5236.7</td>
<td>12 941.22</td>
<td>7259.4</td>
<td>187.31</td>
</tr>
<tr>
<td>test2_102*</td>
<td>213.4</td>
<td>7810.46</td>
<td>526.7</td>
<td>304.76</td>
</tr>
<tr>
<td>Avg. Ratio</td>
<td>0.98</td>
<td>167.07</td>
<td>1.75</td>
<td>2.13</td>
</tr>
</tbody>
</table>

* Our approximation algorithm is enabled. For ILP, we set the timelimit to 3600s.
Advantage 1: The scale of SAT problems remains controllable.

- Original ILP constraints are all cardinality constraints (all coefficients are 1).
- Cardinality constraints can be converted to clauses easily.
- The CNF obtained from cardinality constraints is relatively small.
Advantage 2: Optimality is easier to prove.

Figure: A case study on convergence of ILP and SAT-based decomposers. The first dashed line indicates when an optimal solution is found, and the second indicates when the optimality is proven.
As the graphs get larger, our approximation algorithm remains effective, while the runtime of other methods can grow drastically.
References


Thanks!