AdaOPC: A Self-Adaptive Mask Optimization Framework For Real Design Patterns

Wenqian Zhao¹, Xufeng Yao¹ Ziyang Yu¹, Guojin Chen¹, Yuzhe Ma², Bei Yu¹, Martin D.F. Wong¹

¹ The Chinese University of Hong Kong
² HKUST(GZ)

Sept. 15, 2022
Background and Motivation
Background of Mask Optimization
Background of Mask Optimization

- Rule-based OPC
- Model-based OPC
Background of Mask Optimization

- ILT-based method

![Diagram showing the forward and backward simulation processes with convolution, sigmoid, and sigmoid output approximations.]

Forward simulation process:
- Input \(\{m\}\)
- Convolution
- Aerial Image \(\{Hm\}\)
- Sigmoid
- Output \(\{z = \text{sig}(Hm)\}\)

```
“approximates the aerial image formation process”
“approximates the hard thresholding (resist effect)”
“close to binary”
```

Backward gradient calculation process:
• ML-based method

Deep learning model generate mask or initial mask for iterations

---

All methods have certain drawbacks:

- Rule-based methods lack local fidelity
- Both model-based restricted by the solutions space in advanced technology nodes.
- ILT-based methods iteratively call the imaging system while optimizing an objective function which is time-consuming.
- ML-based OPCs have shown remarkable speed-up in the OPC flows, however not guaranteed to work for some critical patterns.
Hopkins diffraction model decomposed into a sum of coherent systems:

\[
I(x, y) = \sum_{k=1}^{N^2} w_k |M(x, y) \otimes h_k(x, y)|^2, \quad x, y = 1, 2, \ldots N
\]  

(1)

- \( h_k \): k-th kernel, \( w_k \): corresponding weight. "\( \otimes \)" convolution.

\[
I(x, y) \approx \sum_{k=1}^{K} w_k |M(x, y) \otimes h_k(x, y)|^2
\]  

(2)

- Lithography intensity \( I \) sent to photoresist model to generate the final binary pattern \( Z \) with exposure resist threshold \( I_{th} \):

\[
Z(x, y) = \begin{cases} 
1, & \text{if } I(x, y) \geq I_{th}, \\
0, & \text{if } I(x, y) < I_{th},
\end{cases}
\]  

(3)

---

(a) Visualization of EPE measurement (b) Visualization of PVBand.

\[ EPE_{violation}(x, y) = \begin{cases} 
1, & D(x, y) \geq th_{EPE}, \\
0, & D(x, y) \leq th_{EPE}, 
\end{cases} \quad (4) \]

\[ PVBand = \sum_{x,y}^{N^2} |Z_{out} - Z_{in}|, \quad (5) \]
• Patterns scattered unevenly with different complexity. → Solver selection
• Patterns have large ratio of repetition on a full layout. → Mask Reuse
Adaptive Framework
Main Contributions:

- Adaptive solver selection
- Mask reuse ← Critical Patterns
- Dynamic Pattern Library ← Fast Pattern Matching
• Simple and Intuitive: Binary classification with cross-entropy loss $L$:

$$L = -\frac{1}{N} \sum_{i}^{N} y_i \log(p_i) + (1 - y_i) \log(1 - p_i),$$

(6)

• Solver pool extensible, modify loss by adding num of class:

$$L = -\frac{1}{N} \sum_{i}^{N} \sum_{c=1}^{C} y_{ic} \log(p_{ic}).$$

(7)
Whether and how can an optimized mask with location shift be reused?
How to match a same pattern accurately within an acceptable time?
How to measure the geometric similarity of patterns with location shift?
• **Whether** and how can an optimized mask with location shift be reused?

![Diagram showing the process from Target, through Mask, to Wafer Image with OPC and Litho steps, and shift equivariance indicated by $\delta_{\Delta x, \Delta y}$]

• **Shift Equivariance:**

$$\delta_{\Delta x, \Delta y}(P) = Litho(\delta_{\Delta x, \Delta y}(M_P)).$$  \hspace{1cm} (8)$$

We only need to calculate pattern shift since printed masks share same shift as corresponding patterns.
• Whether and **how** can an optimized mask with location shift be reused?

• Pattern shift calibration
  
  • Pixel-wise **cross-correlation** of $P$ and $P'$ reflects the pixel-wise similarity
  • Cross-correlation computation of two large 2-D pattern is time-consuming.
  • Equal to convolution of $P$ and $\text{Rotate}_{180^\circ}(P')$.
  • Accelerated with Fast Fourier Transform (FFT)\(^4\):

\[
x^*, y^* = \arg\max_{x, y} \text{Conv}_\text{FFT}(P, \text{Rotate}(P')),
\]

\[
\Delta x = x^* - x_{ctr}, \quad \Delta y = y^* - y_{ctr}, \tag{9}
\]

---

• How to match a same pattern accurately within an acceptable time?

To tackle the problem of computation intensity of the rigorous method, we maintain a pattern library.

• store the features and optimized masks of previously encountered critical patterns
• use the result of saved masks with similar geometric structure as initial mask, hence reduce the iteration time
• How to match the patterns? - Pattern Library
  • Sparse neighborhood graph structure
  • Graph is divided into hierarchical layers
Hierarchical Navigable Small World (HNSW)\textsuperscript{5}

- the overall number of distance computations is roughly proportional to a product of the average number of greedy algorithm hops by the average degree of the nodes on the greedy path

---

• How to measure the geometric similarity of patterns with location shift?

• embedding for all critical patterns
  • positive samples are patterns that are same or similar
  • negative samples are patterns that are different
  • In learned embedding space, nearest neighbor tend to share similar geometric pattern

• similarity measurement: Euclidean Distance

\[
d_{\text{Euclid}}(V_{P_1}, V_{P_2}) = \|V_{P_1} - V_{P_2}\|_2^2 = \sqrt{\sum_{i=0}^{k} (V_{P_1,i} - V_{P_2,i})^2}. \tag{10}
\]
• Recap on Contrastive Learning

![Contrastive Learning Diagram]

• Data Augmentation: Cropping and shifting

• Supervised Contrastive Loss\(^6\):

\[
L_{\text{supCon}} = - \sum_{i \in I} \frac{1}{|J(i)|} \sum_{j \in J(i)} \log \frac{\exp(z_i \cdot z_j / \tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a / \tau)},
\]

Experimental Results
Results

(a) EPE convergence comparison  
(b) Runtime breakdown

Mask convergence speed comparison with/without Pattern Matching.
Table: Comparisons of baseline approaches

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>DAMO-DGS(^7)</th>
<th>ILT-GPU(^8)</th>
<th>AdaOPC</th>
<th>Avg. Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#EPE</td>
<td>PVB (nm(^2))</td>
<td>RT (s)</td>
<td>#EPE</td>
</tr>
<tr>
<td>1</td>
<td>22</td>
<td>23323</td>
<td>5.20</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>26729</td>
<td>5.26</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>26938</td>
<td>5.22</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>36</td>
<td>27975</td>
<td>5.18</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>28805</td>
<td>5.32</td>
<td>30</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>26960</td>
<td>5.31</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>33</td>
<td>26382</td>
<td>5.23</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>30646</td>
<td>5.38</td>
<td>25</td>
</tr>
<tr>
<td>9</td>
<td>25</td>
<td>24054</td>
<td>5.25</td>
<td>24</td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td>21939</td>
<td>5.29</td>
<td>23</td>
</tr>
<tr>
<td>Avg. Ratio</td>
<td>29.0</td>
<td>26375</td>
<td>5.26</td>
<td>25.6</td>
</tr>
</tbody>
</table>


THANK YOU!