PROS: A Plug-in for Routability Optimization
applied in the State-of-the-art commercial EDA tool
using deep learning

Jingsong Chen¹, Jian Kuang², Guowei Zhao², Dennis J.-H. Huang²,
Evangeline F. Y. Young¹

¹CSE Dept., The Chinese University of Hong Kong
²Cadence Design Systems
Speaker Bio

Jingsong Chen is a fourth-year Ph.D. candidate at the Department of Computer Science and Engineering, The Chinese University of Hong Kong, under the supervision of Prof. Evangeline F.Y. Young since Fall 2017.

Before joining CUHK, he received his Bachelor Degree of Computer Science and Technology from Zhejiang University.

His research interests include physical design of VLSI circuits and related machine learning-based problems.

His homepage: https://jingsongchen.github.io/.
Outline

• Problem Background & Motivation
• Overall Flow of PROS
• FCN-based Predictor for GR Congestion
• Optimizer for GR Cost Parameters
• Experimental Results
• Conclusion
Outline

• Problem Background & Motivation
  • Overall Flow of PROS
  • FCN-based Predictor for GR Congestion
  • Optimizer for GR Cost Parameters
• Experimental Results
• Conclusion
Prior Knowledge Is Useful for P&R

- Routing congestion:
  - Obtained after global routing
  - Affect DRC violation distribution, power, timing, and etc.

- Design rule check (DRC) violation:
  - Obtained after detailed routing
  - Decide whether the design can be taped out successfully or not.

*These figures are from “Accurate Prediction of Detailed Routing Congestion using Supervised Data Learning”, in Proc. ICCD, 2014.
Acquire Prior Knowledge by ML Techniques

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
<th>Work</th>
<th>Applied in an EDA tool?</th>
<th>Avoid extra runtime overhead for feature preparation?</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>ICCD¹</td>
<td>¹“Accurate prediction of detailed routing…”</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>2015</td>
<td>ASQED²</td>
<td>²“An accurate detailed routing routability…”</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>2017</td>
<td>ISPD³</td>
<td>³“Routability optimization for industrial…”</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>2018</td>
<td>ICCAD⁴</td>
<td>⁴“Routenet: Routability prediction for…”</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>2019</td>
<td>TCAD⁵</td>
<td>⁵“Eh? predictor: A deep learning framework…”</td>
<td>No</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Acquire Prior Knowledge by ML Techniques

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
<th>Work Description</th>
<th>Applied in an EDA tool?</th>
<th>Avoid extra runtime overhead for feature preparation?</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>ICCD$^1$</td>
<td>1“Accurate prediction of detailed routing...”</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>2015</td>
<td>ASQED$^2$</td>
<td>2“An accurate detailed routing routability...”</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>2017</td>
<td>ISPD$^3$</td>
<td>3“Routability optimization for industrial...”</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>2018</td>
<td>ICCAD$^4$</td>
<td>4“Routenet: Routability prediction for...”</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td>2019</td>
<td>TCAD$^5$</td>
<td>5“Eh? predictor: A deep learning framework...”</td>
<td>No</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>WANTED</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
PROS: A Plug-in for Routability Optimization

• Main features of PROS:
  • Predict routing congestion by deep learning
    • PROS can learn the behavior of a specific router.
  • Require only data from the placement result
    • Runtime overhead of PROS is negligible.
  • Just optimize the cost parameters of global routing
    • PROS can be easily embedded into any other routers as a plug-in.
  • Work well when integrated into the State-of-the-art commercial EDA tool*
    • PROS is the first ML framework which demonstrates its practicality.

• Target of PROS:
  • Reduce routing congestion and thus improve routability optimization (#DRC)

* Cadence Innovus v20.1
Outline

• Problem Background & Motivation

• Overall Flow of PROS

• FCN-based Predictor for GR Congestion

• Optimizer for GR Cost Parameters

• Experimental Results

• Conclusion
Overall Flow of PROS

The EDA tool equipped with PROS

- For one technology node, the predictor only needs to be trained once.
- Feature extraction, prediction, and optimization of GR cost parameters can be performed very quickly.
- The original parts of the EDA tool are not changed a lot.
Outline

• Problem Background & Motivation
• Overall Flow of PROS
• FCN-based Predictor for GR Congestion
• Optimizer for GR Cost Parameters
• Experimental Results
• Conclusion
Features & Label Summary

• Features:
  • Routing capacity map (horizontal, vertical)
  • Standard cell density map
  • Standard cell pin density map
  • Pin accessibility density map
  • Cross net density map (horizontal, vertical)
  • Flip-flop cell density map
  • Fixed cell density map
  • RUDY map (small nets, large nets)
  • Pin RUDY map

• Label:
  • A 2-D binary map reflecting GR routing congestion:
    • 1 → congestion
    • 0 → no congestion
• The capacity value: low $\rightarrow$ high; the color in image: black $\rightarrow$ white.
• 3-D features $\rightarrow$ 2-D image: accumulate all the layers.
Standard cell & cell pin density map
Pin accessibility density map

- Pin accessibility density map \((F)\) :
  - For each cell \(C\):
    - For each pin \(p\) of \(C\):
      - \(p\) location: \(x, y\),
      - \(npat\) = \#pin accessing patterns of \(C\),
      - \(npin\) = \#pins of \(C\),
      - \(F(x, y) = \frac{1.5^{npin}}{(npat+1)\times npin}\).
  - Higher value \(\rightarrow\) More difficult to route
Cross net density map

• Blue boxes: a 5-pin net.
• Red box: BBox of the net.
• Green box: g-cells which will get net density.
• Horizontal (vertical) cross net density = 1 / #g-cells in a column (row).
Flip-flop cell & fixed cell density map

Flip-flop cell density map

Fixed cell density map
RUDY map

- Large net: HPWL of the net BBox $\geq 15 \times$ g-cell size.
- For each net:
  - RUDY value of each g-cell in the net BBox = wire length / #g-cells in the net BBox.
  - Wire length = HPWL of the net BBox $\times$ Ratio_#pins.

<table>
<thead>
<tr>
<th>#pins</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>&lt;=8</th>
<th>&lt;=10</th>
<th>&lt;=15</th>
<th>&lt;=20</th>
<th>&gt;20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ratio_#pins</td>
<td>1.06</td>
<td>1.13</td>
<td>1.19</td>
<td>1.31</td>
<td>1.42</td>
<td>1.66</td>
<td>1.87</td>
<td>2.22</td>
</tr>
</tbody>
</table>
Pin RUDY map

• Highlight the role of large nets on congestion prediction.
• A combination of cell pin density map and large-net RUDY map.
• The contribution of each pin on its location equals to the wire density of the large net it belongs to.
Label Generation: Smoothening Process

- Purpose: generate more clear label to improve prediction accuracy.
Label Generation: Smoothening Process

- Smoothening process from raw GR congestion map to congestion label:
  1. INSERT: If there are >= 6 congested surrounding g-cells, the center non-congested g-cell will be relabeled as congested.
Label Generation: Smoothening Process

- **Smoothening process from raw GR congestion map to congestion label:**

  1. **INSERT:** If there are $\geq 6$ congested surrounding g-cells, the center non-congested g-cell will be relabeled as congested.

  2. **CLEAN (10 iters):** If there are $\leq 3$ congested surrounding g-cells, the center congested g-cell will be relabeled as non-congested.
Prediction Model

(a) Whole Framework

Input features

(b) Dilated Convolution (DC)
rate = 1
rate = 2

(c) Refinement Block (RB)
Conv (1x1, o_c)
Conv (3x3, o_c, BN, ReLU)
ReLU

(d) Sup-pixel Upsampling Block (SUB)
Outline

• Problem Background & Motivation
• Overall Flow of PROS
• FCN-based Predictor for GR Congestion
• Optimizer for GR Cost Parameters
• Experimental Results
• Conclusion
Optimizer for GR Cost Parameters

• For each congested grid cell:
  • Increase the overflow cost
  • Increase the wire/via cost for the nets with large BBox
Outline

• Problem Background & Motivation
• Overall Flow of PROS
• FCN-based Predictor for GR Congestion
• Optimizer for GR Cost Parameters
• Experimental Results
• Conclusion
Experimental Settings

• Dataset:
  • 19 designs: each design has ~80 different placements. Totally, we have ~1600 design cases.

• Experiment:
  • Divide 19 designs into 5 groups: 4, 4, 4, 4, 3. When testing one group, the remaining four groups will be used for training. Repeat the round of training and testing for 5 times.

• Evaluation:
  • Positive: congested in label; Negative: no congested in label.
  • True positive rate (TPR) = #True positive / #Positive,
  • Precision (PRE) = #True positive / (#True positive + #False positive),
  • False positive rate (FPR) = #False positive / #Negative,
  • F1 score (F1) = (2 * TPR * PRE) / (TPR + PRE),
  • Accuracy (ACC) = (#True positive + #True negative) / (#Positive + #Negative).
Results of Congestion Prediction

- Baselines for congestion prediction:
  - LR1X1: Logistic regression.
  - LR9X9: Enhanced LR1X1 with a window size of $9 \times 9$ g-cells to capture neighboring features.
  - OneSUB: Replace three cascaded SUBs by one RB and one SUB.
  - ThreeSUB-NoSkipAdd: Remove all the skip connections and addition operators.

<table>
<thead>
<tr>
<th>Model</th>
<th>F1 (%)</th>
<th>FPR (%)</th>
<th>ACC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR1X1</td>
<td>63.86</td>
<td>25.39</td>
<td>75.25</td>
</tr>
<tr>
<td>LR9X9</td>
<td>66.38</td>
<td>16.70</td>
<td>79.67</td>
</tr>
<tr>
<td>OneSUB</td>
<td>70.09</td>
<td>10.45</td>
<td>84.23</td>
</tr>
<tr>
<td>ThreeSUB-NoSkipAdd</td>
<td>70.87</td>
<td>8.74</td>
<td>85.32</td>
</tr>
<tr>
<td>PROS</td>
<td>73.34</td>
<td>8.92</td>
<td>86.15</td>
</tr>
</tbody>
</table>
## Results of Routability Optimization

<table>
<thead>
<tr>
<th>Designs</th>
<th>Congested G-cell Ratio</th>
<th>#DRC Violations</th>
<th>Wire Length</th>
<th>Via Count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Orig (%)</td>
<td>PROS (%)</td>
<td>Diff* (%)</td>
<td>Orig</td>
</tr>
<tr>
<td>Design1</td>
<td>3.93</td>
<td>3.74</td>
<td>-6.27</td>
<td>40</td>
</tr>
<tr>
<td>Design2</td>
<td>3.74</td>
<td>3.78</td>
<td>1.07</td>
<td>62</td>
</tr>
<tr>
<td>Design3</td>
<td>2.91</td>
<td>2.78</td>
<td>-4.47</td>
<td>266</td>
</tr>
<tr>
<td>Design4</td>
<td>3.62</td>
<td>3.59</td>
<td>-0.8</td>
<td>59</td>
</tr>
<tr>
<td>Design5</td>
<td>7.40</td>
<td>7.25</td>
<td>-2.03</td>
<td>30</td>
</tr>
<tr>
<td>Design6</td>
<td>5.32</td>
<td>5.28</td>
<td>-0.75</td>
<td>31</td>
</tr>
<tr>
<td>Design7</td>
<td>11.41</td>
<td>10.99</td>
<td>-3.68</td>
<td>2350</td>
</tr>
<tr>
<td>Design8</td>
<td>8.83</td>
<td>8.04</td>
<td>-8.95</td>
<td>1251</td>
</tr>
<tr>
<td>Design9</td>
<td>3.01</td>
<td>2.91</td>
<td>-3.32</td>
<td>1422</td>
</tr>
<tr>
<td>Design10</td>
<td>4.52</td>
<td>4.25</td>
<td>-5.97</td>
<td>453</td>
</tr>
<tr>
<td>Design11</td>
<td>5.48</td>
<td>5.11</td>
<td>-6.75</td>
<td>1105</td>
</tr>
<tr>
<td>Design12</td>
<td>6.05</td>
<td>6.18</td>
<td>2.15</td>
<td>628</td>
</tr>
<tr>
<td>Design13</td>
<td>5.40</td>
<td>5.31</td>
<td>-1.67</td>
<td>65</td>
</tr>
<tr>
<td>Design14</td>
<td>5.35</td>
<td>5.13</td>
<td>-4.11</td>
<td>793</td>
</tr>
<tr>
<td>Design15</td>
<td>4.57</td>
<td>4.44</td>
<td>-2.84</td>
<td>870</td>
</tr>
<tr>
<td>Design16</td>
<td>4.45</td>
<td>4.19</td>
<td>-5.84</td>
<td>636</td>
</tr>
<tr>
<td>Design17</td>
<td>4.28</td>
<td>3.94</td>
<td>-7.94</td>
<td>581</td>
</tr>
<tr>
<td>Design18</td>
<td>8.38</td>
<td>7.91</td>
<td>-5.61</td>
<td>267</td>
</tr>
<tr>
<td>Design19</td>
<td>6.09</td>
<td>5.83</td>
<td>-4.27</td>
<td>879</td>
</tr>
<tr>
<td>Average</td>
<td>-3.79</td>
<td>-49.47</td>
<td>-11.65</td>
<td>0.10</td>
</tr>
</tbody>
</table>

* Diff = 100% * (PROS – Orig) / Orig
Outline

• Problem Background & Motivation
• Overall Flow of PROS
• FCN-based Predictor for GR Congestion
• Optimizer for GR Cost Parameters
• Experimental Results

• Conclusion
Conclusion

• Propose a prediction model used in PROS to predict routing congestion:
  • Based on fully convolutional network (FCN),
  • Only use data collected from placement,
  • Achieve a high prediction accuracy.

• By utilizing prediction results, PROS can improve routability:
  • Effectively reduce routing congestion ratio (-3.79%) and DRC violation number (-11.65%) by optimizing cost parameters of GR,
  • Maintain wire length (+0.10%) and via count (+0.02%).
Thank you for your attention!
Q & A