Dr. CU: Detailed Routing by Sparse Grid Graph and Minimum-Area-Captured Path Search

Gengjie Chen, Chak-Wa Pui, Haocheng Li, Jingsong Chen, Bentian Jiang, Evangeline F. Y. Young

CSE Department, The Chinese University of Hong Kong

Jan. 24, 2019
Introduction – Key Challenges of Detailed Routing

- Compared to global routing
  - On a significantly larger solution space
    $(10^4 \times 10^4 \times 10$ grid graph)
  - Has many design rules
- Even more time-consuming and complicated in advanced tech nodes
Introduction – Design Rules

- Short
- Spacing: parallel-run spacing, EOL spacing, cut spacing, ...
- Min area

(a) EOL spacing

(b) Parallel-run spacing
Introduction – Problem Formulation of Detailed Routing

Given

- Placed netlist
- Routing guides (from global routing)
- Routing tracks
- Design rules

Route all the nets & minimize a weighted sum of

- Total wire length
- Total via count
- Non-preferred usage (including out-of-guide & off-track wires/vias, wrong-way wires)
- Design rule violations
Introduction – Our Approach

- Two-level sparse data structures $\implies$ efficiency
- Min-area-captured path search $\implies$ quality
- Bulk synchronous parallel $\implies$ further speed-up
Outline

Two-Level Sparse Data Structures

Min-Area-Captured Path Search

Bulk Synchronous Parallel

Experimental Results
Two-Level Sparse Data Structures

- Routing region of a net
- Local grid graph
- Global grid graph
- Routing topology
- Maze
- Route
- Cache
- Query
- Record
- Edge
- Usage
Sparse Global Grid Graph

- Store routed edges by BSTs & intervals
- Query via/wire conflict efficiently with help of LUTs
  - Support batch query
Example: query the conflict with a single candidate via
Example: query the conflict with many neighboring candidate vias

Sparse Global Grid Graph
Sparse Local Grid Graph

- Cache global graph on routing region
  - Subgraph of full-chip grid graph on routing region of a net
  - Store vertex/edge information explicitly by direct-address tables
- Remove redundant vertices

(a) Before removing redundant vertices
(b) After removing redundant vertices
Sparse Local Grid Graph

- Edge cost in local grid graph captures
  - Base wire & via cost
  - Out-of-guide penalty
  - Short & spacing violation penalty
- How about min-area violation?
Outline

Two-Level Sparse Data Structures

Min-Area-Captured Path Search

Bulk Synchronous Parallel

Experimental Results
Min-Area-Captured Path Search

Capture min area cost in path search (without considering wire extension)

(a) A normal path search without considering min-area violation
(b) Post fixing by extending wire
(c) Forcing the min length of wire segment in path search

(Suppose the min area rule implies a length of three pitches)
Min-Area-Captured Path Search

Capture min area cost in path search (with wire extension considered)

(d) Detour due to the forcing

(e) Path search with wire extension considered
Min-Area-Captured Path Search

Normal Dijkstra’s algorithm

- Cost/distance that can be directly incremented
  - $\text{cost}(v_1 \leadsto v_2 \leadsto v_3) = \text{cost}(v_1 \leadsto v_2) + \text{cost}(v_2 \leadsto v_3)$

- A vertex is visited at most once

- Back track by each vertex having a prefix

Min-area-captured path search

- Uncertain cost
  - Lower bound: edge cost sum + min-area penalty on previous wires
  - Upper bound: lower bound + min-area penalty on the current wire

- A vertex may be visited multiple times

- Back track by (smart) pointers
Outline

Two-Level Sparse Data Structures

Min-Area-Captured Path Search

Bulk Synchronous Parallel

Experimental Results
Bulk Synchronous Parallel

- Route batches of independent nets one after another
- Fast scheduling followed by load balancing

Figure: Scheduling without load balancing
Bulk Synchronous Parallel

- Route batches of independent nets one after another
- Fast scheduling followed by load balancing

Figure: Scheduling with load balancing
Separate a batch into **routing** and **committing** phases

<table>
<thead>
<tr>
<th></th>
<th>Routing phase</th>
<th>Committing phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain</td>
<td>whole routing region</td>
<td>solution paths</td>
</tr>
<tr>
<td>global grid graph</td>
<td>read</td>
<td>write</td>
</tr>
<tr>
<td>access</td>
<td></td>
<td></td>
</tr>
<tr>
<td>locked?</td>
<td>lock-free</td>
<td>locked</td>
</tr>
</tbody>
</table>
Outline

Two-Level Sparse Data Structures

Min-Area-Captured Path Search

Bulk Synchronous Parallel

Experimental Results
## Experimental Results

- On ISPD 2018 Contest Benchmarks

<table>
<thead>
<tr>
<th>Benchmark</th>
<th># std. cells</th>
<th># block macros</th>
<th># nets</th>
<th># pins</th>
<th># IO pins</th>
<th># layers</th>
<th>M2 # tracks</th>
<th>M2 pitch (μm)</th>
<th>Tech. node (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>8879</td>
<td>0</td>
<td>3153</td>
<td>17203</td>
<td>0</td>
<td>9</td>
<td>977</td>
<td>0.2</td>
<td>45</td>
</tr>
<tr>
<td>test2</td>
<td>35913</td>
<td>0</td>
<td>36834</td>
<td>159201</td>
<td>1211</td>
<td>9</td>
<td>3254</td>
<td>0.2</td>
<td>45</td>
</tr>
<tr>
<td>test3</td>
<td>35973</td>
<td>4</td>
<td>36700</td>
<td>159703</td>
<td>1211</td>
<td>9</td>
<td>4943</td>
<td>0.2</td>
<td>45</td>
</tr>
<tr>
<td>test4</td>
<td>72094</td>
<td>0</td>
<td>72401</td>
<td>318245</td>
<td>1211</td>
<td>9</td>
<td>8886</td>
<td>0.1</td>
<td>32</td>
</tr>
<tr>
<td>test5</td>
<td>71954</td>
<td>0</td>
<td>72394</td>
<td>318195</td>
<td>1211</td>
<td>9</td>
<td>9800</td>
<td>0.1</td>
<td>32</td>
</tr>
<tr>
<td>test6</td>
<td>107919</td>
<td>0</td>
<td>107701</td>
<td>475541</td>
<td>1211</td>
<td>9</td>
<td>5312</td>
<td>0.1</td>
<td>32</td>
</tr>
<tr>
<td>test7</td>
<td>179865</td>
<td>16</td>
<td>179863</td>
<td>793289</td>
<td>1211</td>
<td>9</td>
<td>13500</td>
<td>0.1</td>
<td>32</td>
</tr>
<tr>
<td>test8</td>
<td>191987</td>
<td>16</td>
<td>179863</td>
<td>793289</td>
<td>1211</td>
<td>9</td>
<td>13500</td>
<td>0.1</td>
<td>32</td>
</tr>
<tr>
<td>test9</td>
<td>192911</td>
<td>0</td>
<td>178857</td>
<td>791761</td>
<td>1211</td>
<td>9</td>
<td>13500</td>
<td>0.1</td>
<td>32</td>
</tr>
<tr>
<td>test10</td>
<td>290386</td>
<td>0</td>
<td>182000</td>
<td>811761</td>
<td>1211</td>
<td>9</td>
<td>13500</td>
<td>0.1</td>
<td>32</td>
</tr>
</tbody>
</table>
## Experimental Results

### On ISPD 2018 Contest Benchmarks

<table>
<thead>
<tr>
<th>Metric</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>wire length</td>
<td>0.5</td>
</tr>
<tr>
<td># vias</td>
<td>2</td>
</tr>
<tr>
<td>out-of-guide wire length</td>
<td>1</td>
</tr>
<tr>
<td># out-of-guide vias</td>
<td>1</td>
</tr>
<tr>
<td>off-track wire length</td>
<td>0.5</td>
</tr>
<tr>
<td># off-track vias</td>
<td>1</td>
</tr>
<tr>
<td>wrong-way wire length</td>
<td>1</td>
</tr>
<tr>
<td>short metal area</td>
<td>500</td>
</tr>
<tr>
<td># spacing violations</td>
<td>500</td>
</tr>
<tr>
<td># min-area violations</td>
<td>500</td>
</tr>
</tbody>
</table>
Experimental Results

We do not abuse contest metric by converting spacing violations into short ones with zero area.
Experimental Results

- 8 threads gives almost $4 \times$ speed-up
- Load balancing contributes 2.52% improvement
## Experimental Results

<table>
<thead>
<tr>
<th>WL # vias</th>
<th>Non-preferred usage</th>
<th>Design rule violations</th>
<th>Quality score</th>
<th>Mem (GB)</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out-of-guide</td>
<td>Off-track</td>
<td>Wrong-way</td>
<td># short</td>
<td>Short area</td>
<td># min spacing area</td>
</tr>
<tr>
<td>WL # vias</td>
<td>WL # vias</td>
<td>WL # vias</td>
<td>WL # vias</td>
<td>WL # vias</td>
<td>WL # vias</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Dr. CU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>test1</td>
<td>434914</td>
<td>34443</td>
<td>4352</td>
<td>859</td>
<td>276</td>
</tr>
<tr>
<td>test2</td>
<td>781728</td>
<td>339055</td>
<td>104720</td>
<td>11784</td>
<td>4353</td>
</tr>
<tr>
<td>test3</td>
<td>8707641</td>
<td>331958</td>
<td>176736</td>
<td>10731</td>
<td>4344</td>
</tr>
<tr>
<td>test4</td>
<td>26042785</td>
<td>701994</td>
<td>769265</td>
<td>31444</td>
<td>41791</td>
</tr>
<tr>
<td>test5</td>
<td>27852167</td>
<td>942588</td>
<td>694224</td>
<td>43071</td>
<td>13390</td>
</tr>
<tr>
<td>test6</td>
<td>35813473</td>
<td>1446807</td>
<td>976672</td>
<td>68656</td>
<td>20357</td>
</tr>
<tr>
<td>test7</td>
<td>65360688</td>
<td>2349580</td>
<td>2187794</td>
<td>101866</td>
<td>33105</td>
</tr>
<tr>
<td>test8</td>
<td>65668468</td>
<td>2360231</td>
<td>2288159</td>
<td>102982</td>
<td>33737</td>
</tr>
<tr>
<td>test9</td>
<td>54993356</td>
<td>2358857</td>
<td>1604576</td>
<td>115465</td>
<td>33470</td>
</tr>
<tr>
<td>test10</td>
<td>68282001</td>
<td>2532666</td>
<td>2826908</td>
<td>140343</td>
<td>32865</td>
</tr>
<tr>
<td>Avg. ratio</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>1st place of ISPD 2018</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>test1</td>
<td>472032</td>
<td>41641</td>
<td>6246</td>
<td>1385</td>
<td>3528</td>
</tr>
<tr>
<td>test2</td>
<td>8150588</td>
<td>409551</td>
<td>71685</td>
<td>13451</td>
<td>20402</td>
</tr>
<tr>
<td>test3</td>
<td>9086139</td>
<td>427410</td>
<td>69182</td>
<td>2450</td>
<td>33470</td>
</tr>
<tr>
<td>test4</td>
<td>27514053</td>
<td>858224</td>
<td>240226</td>
<td>8841</td>
<td>150961</td>
</tr>
<tr>
<td>test5</td>
<td>29151781</td>
<td>1140804</td>
<td>309785</td>
<td>39002</td>
<td>45523</td>
</tr>
<tr>
<td>test6</td>
<td>3798769</td>
<td>1775407</td>
<td>467961</td>
<td>24448</td>
<td>135900</td>
</tr>
<tr>
<td>test7</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
<td>fail</td>
</tr>
<tr>
<td>test8</td>
<td>69559382</td>
<td>2929578</td>
<td>1006247</td>
<td>82478</td>
<td>375236</td>
</tr>
<tr>
<td>test9</td>
<td>58803453</td>
<td>2920259</td>
<td>813750</td>
<td>67367</td>
<td>331766</td>
</tr>
<tr>
<td>test10</td>
<td>72244024</td>
<td>3110163</td>
<td>1414338</td>
<td>81831</td>
<td>625291</td>
</tr>
<tr>
<td>Avg. ratio</td>
<td>1.06</td>
<td>1.23</td>
<td>0.58</td>
<td>0.73</td>
<td>9.02</td>
</tr>
</tbody>
</table>
Experimental Results

Compared with 1st place of ISPD Contest

- 35% better routing quality under contest metric
  - 5% less wire length and 19% fewer vias
  - 3.7× design rule violation clearance
- 26.7× reduction in number of design rule violations
- 80% - 93% memory reduction
- 2.5× - 15× speed-up
Experimental Results

Figure: Comparison on quality score under the metric of ISPD 2018 Contest*

*1st place fail in test7, 3rd place fail in test7 & test8
Experimental Results

Figure: Comparison on total number of short, minimum area, and spacing violations
Experimental Results

Figure: Comparison on runtime
Conclusion

Our approach

- Two-level sparse data structures $\implies$ efficiency
- Min-area-captured path search $\implies$ quality
- Bulk synchronous parallel $\implies$ further speed-up

A stronger Dr. CU? To be published...

- Shorter wire length, fewer vias
- Significantly fewer design rule violations (by 1–2 orders of magnitudes)
- Faster