Starfish: An Efficient P&R Co-Optimization Engine with A*-based Partial Rerouting

Fangzhou Wang, Lixin Liu, Jingsong Chen, Jinwei Liu, Xinshi Zang, and Martin D.F. Wong

CSE Dept., The Chinese University of Hong Kong

Nov. 03, 2021
Outline

Introduction

Overall Flow

Experimental Results
Outline

Introduction

Overall Flow

Experimental Results
## Placement and Routing (P&R)

### A common two-stage flow

- **Placement and Routing as 2 sub-problems**

  - **Placement**: decide the locations of circuit components
    - Objective: focuses on optimizing the half-perimeter wire length (HPWL) under cell/pin density constraints
  
  - **Routing**: build the connection between cells
    - Objective: minimize the routed wire length and the number of overflows

- **Pros**: Reduction in design complexity
- **Cons**: Misalignment between objectives can degrade the final solution quality
Misalignment between Placement and Routing

Illustration for routing with cell movement. Both (a) and (b) have the same HPWL, but the routed wire length in (b) is shorter.
Routing with Cell Movement (ICCAD 2020 Contest)

**Input**
- 3D Design layout (L × R × C)
- Netlist and min-routing-layer for each net
- Initial placement and routing result
- Max-cell-move

**Goals**
Generate a set of **placement** and **routing** (P&R) results that
- minimize total routed wire length (that is, maximize the wire length reduction)
- comply with hard constraints:
  - max-cell-move
  - min-routing-layer
  - connectivity
  - preferred routing direction
  - overflow-free
Outline

Introduction

Overall Flow

Experimental Results
Overall Flow

(1) Pre-processing
   ▶ Initial solution refinement
   ▶ Congestion-driven rerouting

(2) Multi-threaded cell movement
   ▶ Lookup table-based gain estimation
   ▶ A*-based partial rerouting
   ▶ Selective net rerouting: detours removal
   ▶ Gain threshold $\tau$ for implicit cell ordering

(3) Post-processing
   ▶ Greedy cell put-back
   ▶ Wire length-driven rerouting

Gain Threshold $\tau = K(K > 0)$

Overall flow.
Outline

Introduction

Overall Flow

Experimental Results
## Comparison with the Top-3 Winners

**Table:** Experimental Results on the ICCAD 2020 Benchmark

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>1st Place Team</th>
<th>2nd Place Team</th>
<th>3rd Place Team</th>
<th>Ours (Starfish)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TWL&lt;sub&gt;input&lt;/sub&gt;</td>
<td>Score</td>
<td>RT (s)</td>
<td>Score</td>
</tr>
<tr>
<td>case3</td>
<td>32600</td>
<td>11425</td>
<td>34</td>
<td>11428</td>
</tr>
<tr>
<td>case4</td>
<td>4680681</td>
<td>2046811</td>
<td>2221</td>
<td>2048105</td>
</tr>
<tr>
<td>case5</td>
<td>1763627</td>
<td>695219</td>
<td>903</td>
<td>685173</td>
</tr>
<tr>
<td>case6</td>
<td>7188481</td>
<td>2721274</td>
<td>3171</td>
<td>2687926</td>
</tr>
<tr>
<td>case3B</td>
<td>29748</td>
<td>11237</td>
<td>31</td>
<td>11073</td>
</tr>
<tr>
<td>case4B</td>
<td>4886698</td>
<td>2182574</td>
<td>2299</td>
<td>2180172</td>
</tr>
<tr>
<td>case5B</td>
<td>1721530</td>
<td>664347</td>
<td>886</td>
<td>654797</td>
</tr>
<tr>
<td>case6B</td>
<td>7340802</td>
<td>2748097</td>
<td>3406</td>
<td>2722222</td>
</tr>
<tr>
<td>Avg. ratio</td>
<td>-</td>
<td>1.000</td>
<td>1.000</td>
<td>0.992</td>
</tr>
</tbody>
</table>

* The score and runtime statistics of the top-3 winners are provided by the contest organizer with Intel Xeon E7-4820 CPU (2.00 GHz, 8 cores). For us, all experiments are run on a Linux machine with 2.90 GHz Intel Xeon CPU and 8 threaded enabled.

---

1 Kai-Shun Hu et al. (2020). “ICCAD-2020 CAD contest in Routing with Cell Movement”. In: Proc. ICCAD.
Thank You!
Hu, Kai-Shun, Ming-Jen Yang, Tao-Chun Yu, and Guan-Chuen Chen (2020). “ICCAD-2020 CAD contest in Routing with Cell Movement”. In: Proc. ICCAD.