DATE 21
DESIGN, AUTOMATION & TEST IN EUROPE
01 – 05 February 2021 · virtual conference
The European Event for Electronic System Design & Test

# Global Placement with Deep Learning-Enabled Explicit Routability Optimization

**Siting Liu**[1], Qi Sun[1], Peiyu Liao[1], Yibo Lin[2], Bei Yu[1],

[1]The Chinese University of Hong Kong
[2]Peking University

# Global Placement with Deep Learning-Enabled Explicit Routability Optimization

## Background & Motivation

- Previous works proposed probabilistic estimator or a real global router to evaluate routability.
- With deep learning, the routability prediction can be **fast** and **accurate**.
- Most studies on routability prediction **stop** to integrate the models to the SOTA global placer for routability optimization.

## Methodologies

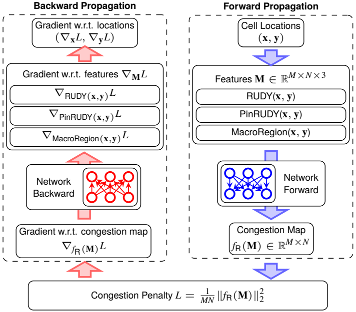
- Estimate routability performance with deep learning model.
- Add the result of a routability estimator into the objective function as a penalty.
- Get gradient *w.r.t.* locations with backward propagation to explicitly optimize routability.
- **Objective Function**:

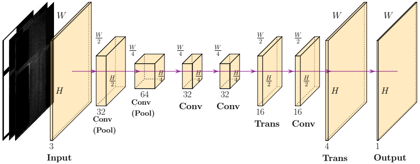
$$\sum_{e \in E} W_e(\mathbf{x}, \mathbf{y}) + \lambda D(\mathbf{x}, \mathbf{y}) + \eta L(\mathbf{x}, \mathbf{y}).$$

- **Gradient** to optimize cell locations:

$$\sum_{e \in E} \nabla_x W_e(\mathbf{x}, \mathbf{y}) + \lambda \nabla_x D(\mathbf{x}, \mathbf{y}) + \eta \nabla_x L(\mathbf{x}, \mathbf{y}),$$

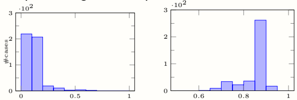
$$\sum_{e \in E} \nabla_y W_e(\mathbf{x}, \mathbf{y}) + \lambda \nabla_y D(\mathbf{x}, \mathbf{y}) + \eta \nabla_y L(\mathbf{x}, \mathbf{y}).$$




## Results

- Apply NRMS(left) and SSIM(right) to evaluate the deep learning model's performance.




- H/V-CR: Horizontal and Vertical Congestion Rate; WL : routed WireLength by global router (e+06 um); RT : RunTime.

| Benchmark | DREAMPlace | | | | Ours | | | |
|---|---|---|---|---|---|---|---|---|
| | H-CR | V-CR | WL | RT (s) | H-CR | V-CR | WL | RT (s) |
| des_perf_1 | 0.143 | 0.129 | 1.23 | 10.868 | 0.153 | 0.126 | 1.23 | 44.07 |
| des_perf_a | 0.015 | 0.021 | 2.05 | 12.834 | 0.020 | 0.028 | 1.91 | 44.51 |
| des_perf_b | 0.005 | 0.010 | 1.71 | 11.829 | 0.004 | 0.010 | 1.71 | 45.99 |
| fft_1 | 0.106 | 0.063 | 0.45 | 7.656 | 0.101 | 0.061 | 0.45 | 43.66 |
| fft_2 | 0.664 | 0.006 | 0.44 | 7.636 | 0.665 | 0.006 | 0.44 | 48.05 |
| fft_a | 0.248 | 0.016 | 1.08 | 7.317 | 0.191 | 0.015 | 0.97 | 42.78 |
| fft_b | 0.177 | 0.026 | 1.21 | 7.942 | 0.142 | 0.047 | 1.12 | 46.81 |
| matrix_mult_1 | 0.165 | 0.340 | 2.19 | 13.69 | 0.168 | 0.334 | 2.19 | 52.9 |
| matrix_mult_2 | 0.253 | 0.238 | 2.28 | 13.69 | 0.251 | 0.242 | 2.28 | 52.25 |
| matrix_mult_a | 0.145 | 0.113 | 5.45 | 15.30 | 0.020 | 0.024 | 3.49 | 47.91 |
| matrix_mult_b | 0.044 | 0.025 | 4.51 | 14.91 | 0.020 | 0.028 | 3.47 | 50.5 |
| matrix_mult_c | 0.089 | 0.017 | 4.87 | 14.38 | 0.029 | 0.016 | 3.42 | 48.41 |
| pci_bridge32_a | 0.192 | 0.098 | 0.52 | 7.33 | 0.076 | 0.036 | 0.43 | 44.64 |
| pci_bridge32_b | 0.001 | 0.005 | 0.83 | 8.08 | 0.002 | 0.008 | 0.65 | 43.72 |
| superblue12 | 0.125 | 0.374 | 38.1 | 96.18 | 0.131 | 0.379 | 36.46 | 547.8 |
| superblue14 | 0.041 | 0.051 | 26.1 | 54.26 | 0.055 | 0.081 | 25.28 | 168.35 |
| superblue16_a | 0.090 | 0.013 | 28.2 | 54.92 | 0.164 | 0.028 | 28.70 | 170.21 |
| superblue19 | 0.033 | 0.093 | 17.0 | 42.23 | 0.039 | 0.091 | 16.70 | 97.84 |
| Average | 0.141 | 0.091 | 7.68 | **22.28** | **0.124** | **0.087** | **7.27** | 91.13 |

- We get 9.05% and 5.3% reduction on total congestion rate and routed wirelength respectively compared with the placer without congestion penalty.