PCB-Migrator: Automated PCB PnR Migration

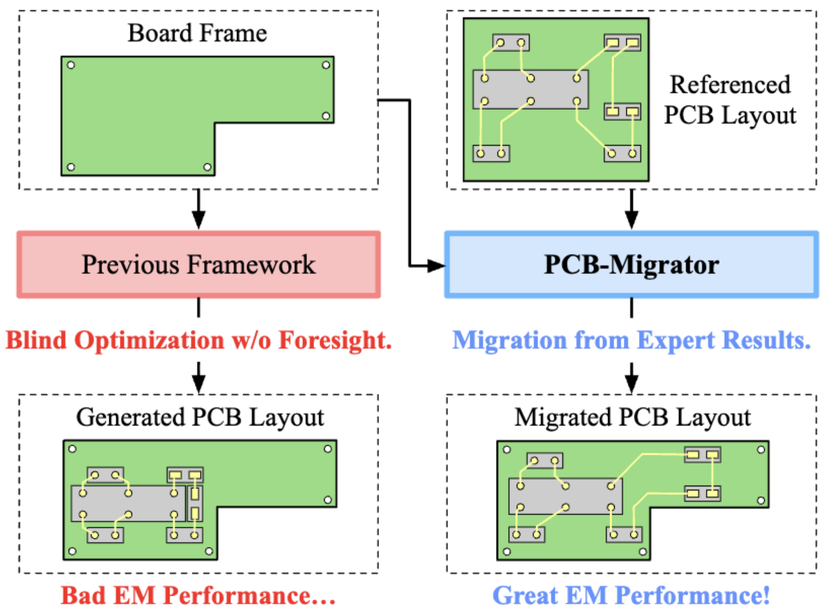
Yaohui Han[1], Beichen Li[2], Rongliang Fu[1], Qunsong Ye[3], Zhiyuan Lu[2], Junchen Liu[2], Bei Yu[1], Tsung-Yi Ho[1], Tinghuan Chen[2,*]
[1]CUHK, [2]CUHK-Shenzhen, [3]Index

Introduction

The exiting “design-from-scratch” frameworks [1-5] only use some approximate estimation indicators. These methods may have the ability to find feasible PnR solutions, but it falls short of meeting the industrial requirements, with poor performance.

Previous PCB PnR Frameworks:
- **Heuristic**: SA-PCB [1], NS-Place [2].
- **AI-based**: DeepPCB [3], PCBAgent [4].

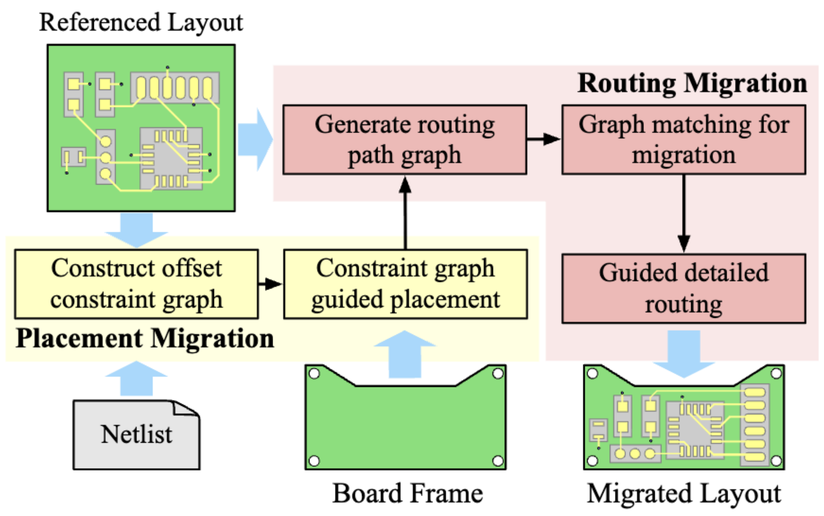
Existing Limitations:
- Lack of foresight for routing at placement stage.
- Unable to reuse existing expert design results.
- The metrics (e.g. HPWL) are not good at measuring layout performance and usability.



Comparison between PCB-Migrator and previous frameworks

Core Contributions:
- The first PCB PnR migration framework.
- **Offset Constraint Graph** (OCG) is proposed to guide placement migration.
- Routing migration is transformed into a **graph matching** problem to guide A*-based routing.
- Experiments demonstrate that PCB-Migrator improves the overall performance compared with all baselines.

Methodology



Overall flow of PCB-Migrator

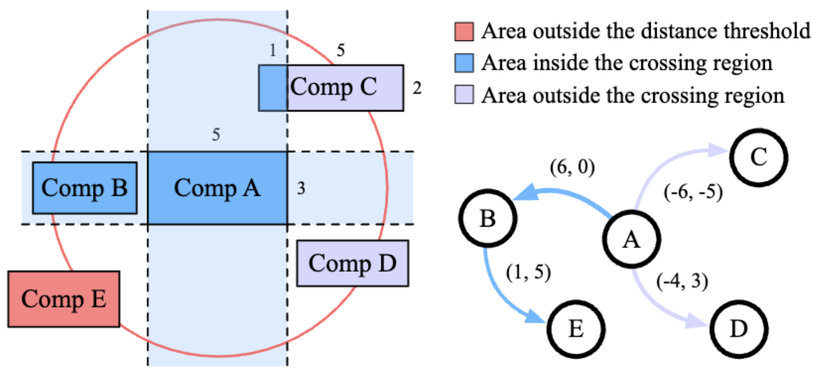
1. Placement Migration

1.1. Offset Constraint Graph Construction

In placement migration, we propose OCG to extract and preserve the positional relationship between components.

In OCG, edges are constructed for all other components within the distance threshold. The weights of the edges are calculated as:

$$\mathcal{W}_{ij} = e^{-d_{ij}} + \mu \cdot \frac{S_{ij}}{\min(S_i, S_j)},$$



Example of OCG construction

There are stronger constraints between components with **absolute horizontal or vertical position relationships**, and closer components have stronger constraints.

1.2. OCG-Guided Placement

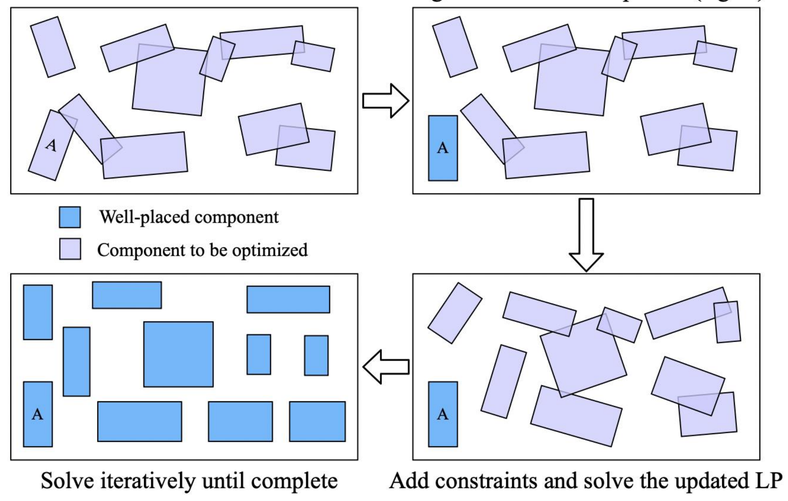
With the OCG, PCB-Migrator then formulates the placement migration into an MILP-like problem:

$$\min \sum_{i=1}^{|C|}\sum_{j=i}^{|C|} \mathcal{W}_{ij} \cdot (|\Delta x_{ij}^o - \Delta x_{ij}^n| + |\Delta y_{ij}^o - \Delta y_{ij}^n|) + \gamma_1 \cdot \sum_{i=1}^{|C|} \Delta\theta_i + \gamma_2 \cdot \sum_{m=1}^{|net|} \mathrm{HPWL}(x, y, \theta),$$

$$\text{s.t. } \theta_i \in \{0°, 90°, 180°, 270°\},\ \forall i \in C,$$

No overlapping is approved between components, and all components should be placed within the board frame.

However, the efficiency of directly using the solver [6] to perform placement is very low. Therefore, we design a **relaxation and legalization strategy**:

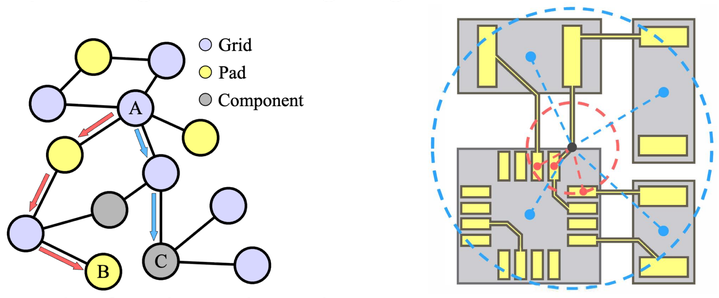


Relaxation and legalization iteratively to speed up the placement

2. Routing Migration

2.1. Routing Path Graph Generation

To preserve the routing characteristics, we transform the placement results into routing path graphs to guide routing migration.



Example of routing path graph

Fine- and coarse-grained areas

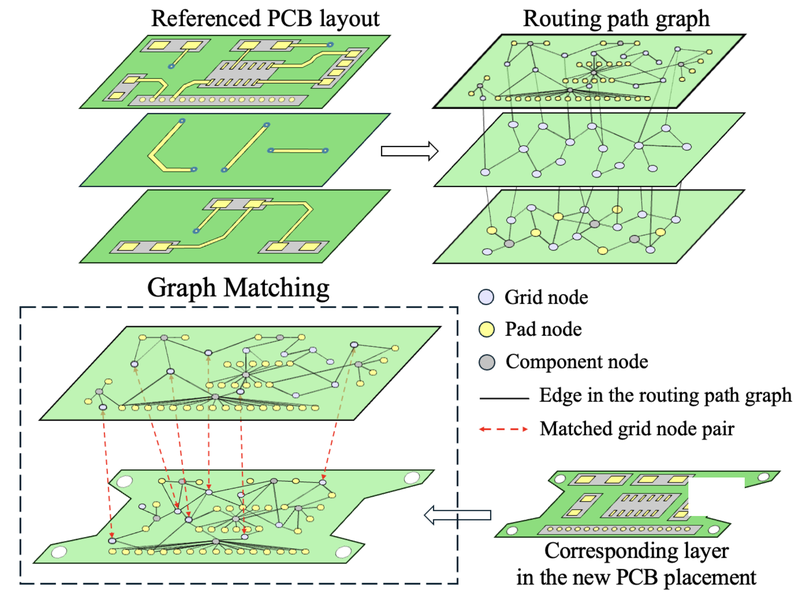
To capture the neighborhood connections for pads and components, we perform BFS for each grid node in routing path graphs.

2.2. Routing Path Graph Matching

PCB-Migrator maps the routing result of referenced PCB layout to the new PCB. And this process can be formulated as **bipartite graph matching**:

$$\max \sum_{u=1}^{|N_o|}\sum_{v=1}^{|N_n|} \cos(u, v) \cdot x_{uv},$$

$$\text{s.t. } \sum_{v=1}^{|N_n|} x_{uv} = 1,\ \forall u \in N_o,\ x_{uv} \in \{0, 1\},\ \forall u, v,$$



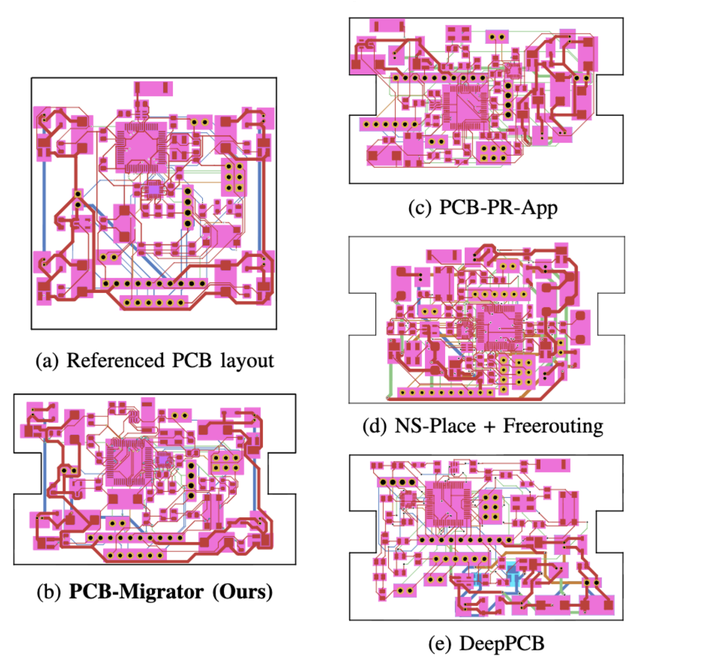
Routing path graph generation and matching for migration

2.3. Guided Detailed Routing

Following the graph-based global routing, the generated paths serve as guidance for A* search-based detailed routing.

Experimental Results

- Compared with the classical baseline, PCB-PR-App [1], PCB-Migrator shows 22.4% wirelength and 48.1% via count reduction.
- Compared with the SOTA baseline, DeepPCB [3], PCB-Migrator still shows significant lower wirelength (15.6%) and via count (11.1%).
- PCB-Migrator uses only 5.56% of the runtime and attains 110.51% cost compared to MILP [6].
- Moreover, PCB-Migrator’s PnR results have siginifcant lower insertion loss and crosstalk, which means better EM performance.



Conclusion

- **PCB-Migrator**: **the first** PCB PnR migration framework to reuse the expert designs.
- A complete and stable PnR migration process.
- Experimental results demonstrate that PCB-Migrator can improve the overall performance of PCB PnR migration.

References

[1] PCB-PR-App. [Online]. Available: https://github.com/The-OpenROAD-Project/PCB-PR-App.
[2] Cheng, Chung-Kuan, Chia-Tung Ho, and Chester Holtz. "Net separation-oriented printed circuit board placement via margin maximization." 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2022.
[3] DeepPCB. [Online]. Available: https://app.deeppcb.ai/.
[4] Chen, Lin, et al. "PCBAgent: An agent-based framework for high-density printed circuit board placement." Proceedings of the 30th Asia and South Pacific Design Automation Conference. 2025.
[5] Li, Shanyi, et al. "HiePlace: Efficient Hierarchical PCB Placement." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2025).
[6] Gurobi Optimization. [Online]. Available: https://www.gurobi.com/