

MegaRoute: Universal Automated Large-Scale PCB Routing Method with Adaptive Step-Size Search

Haiyun Li^{1,2}, Jixin Zhang^{1,†}

¹School of Computer Science, Hubei University of Technology, Wuhan, China

²Shenzhen International Graduate School, Tsinghua University, Shenzhen, China

Abstract—The automation of very large-scale PCB routing has long been an unresolved problem within the industry due to the variant electronic components and complex design rules. Existing automated PCB routing methods are primarily designed for single component (e.g., BGA, BTB, etc.) or for simple and small-scale PCBs, and often fail to meet the industry requirements for large-scale PCBs. The biggest challenge is to ensure nearly 100% routability and DRC compliance while achieving high efficiency for large-scale PCBs with various components. To address this challenge, we propose MegaRoute, a precise, efficient, and universal PCB routing method that surpasses the routing routability and DRC compliance of existing methods, including commercial tools, for PCBs with thousands of nets. MegaRoute introduces an adaptive step-size search algorithm that adjusts exploration steps based on design rules and surrounding obstacles, improving both routability and efficiency. We incorporate shape-based obstacle detection for strict DRC compliance and use routing optimization techniques to enhance routability. We conduct extensive experiments on hundreds of real-world PCBs, including mainboard PCBs with thousands of nets. The results show that MegaRoute achieves over 98% routability across all PCBs with DRC-free results, significantly outperforming the state-of-the-art methods and mainstream commercial tools.

Index Terms—Adaptive Step-Size Search, Large-Scale PCB Routing, Mainboard, Universal PCB Routing

I. INTRODUCTION

Printed circuit board (PCB) routing, which plays a pivotal role in the modern electronics industry, involves creating electrical connections between components on the board [1], [2]. Routing large-scale PCBs, such as mainboards, can be very time-consuming, often taking months of manual effort. Despite decades of research, most existing automated routing methods, even including commercial tools, are designed for local components (e.g., BGA, BTB, etc.) or for simple, small-scale PCBs, and often fail to meet the industry requirements for large-scale PCBs. The primary reason is that, in large-scale PCBs with thousands of connections, achieving nearly 100% routability requires a significant increase in computational complexity for these methods. If attempts are made to simplify the routing problem to mitigate this increase in complexity, it often results in inadequate DRC compliance. With the increasing scale of electronic components, the demand for efficient automated large-scale PCB routing has risen significantly [3].

PCB automatic routing methods can be broadly categorized into two types: modular routing techniques and universal PCB routing. Modular routing techniques, such as escape routing

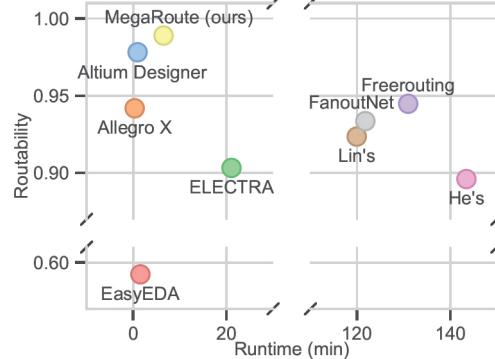


Fig. 1. Comparison of our MegaRoute with state-of-the-art methods in terms of average runtime and routability trade-offs on 101 real-world PCBs.

[4]–[7], bus routing [6], [8], [9], and layer assignment [10]–[12], focus on dividing PCB routing into independent modules to address specific components or staged problems. Unlike these approaches, our goal with universal routing is to develop an end-to-end solution with complete routing results for entire large-scale PCBs. Universal routing requires algorithms to uniformly model the entire PCB, handling various components and design rules. However, existing universal routing methods, which can be broadly classified into grid-based [13]–[16] and shape-based methods [17]–[20], face several limitations.

- **Grid-based methods:** These methods focus on improving routability by using minimum-cost grid search algorithms to find optimal routing paths, relying on fixed grids to detect obstacles and search for paths. However, the grid limits the precision of obstacle detection, which directly results in DRC violations. Increasing the grid resolution can improve precision but lead to huge time-consumption.
- **Shape-based methods:** In contrast, shape-based methods focus on improving efficiency by modeling the PCB layout as geometric objects and searching heuristic path without grid partitioning. Such solutions avoid precision errors and are more efficient, but they often result in low routability due to the incomplete search space.

These limitations make it non-trivial to improve routability, DRC compliance, and efficiency simultaneously, particularly when scaling up to large-scale PCBs where these issues become more significant.

To address these limitations, we aim to integrate the strengths of both methods, overcome their respective inherent limitations,

[†]Corresponding author: Jixin Zhang {Email: zhangjx@hbut.edu.cn}.

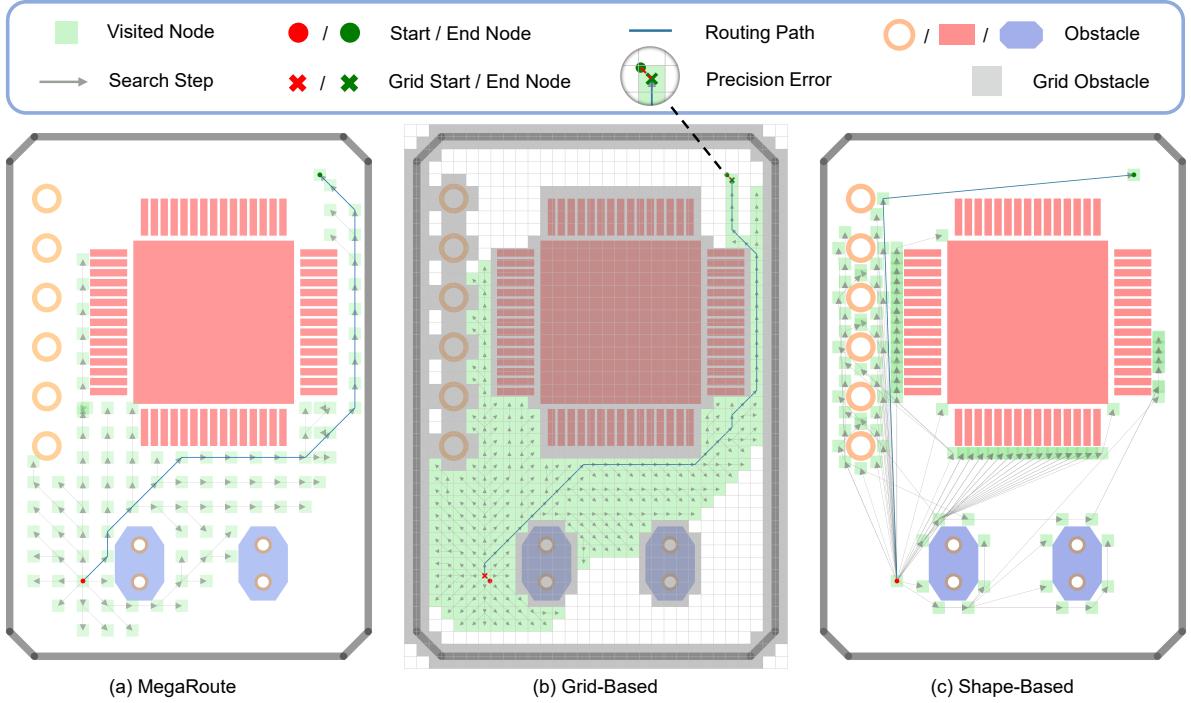


Fig. 2. Comparative visualization of routing algorithms. This diagram shows the routing strategies of MegaRoute, grid-based routing, and shape-based routing on a simplified PCB layout. (a) MegaRoute: Uses adaptive step-size search to achieve routing results comparable to grid-based methods with fewer steps, while maintaining necessary search space for optimal routing path and achieving high precision by shape-based obstacle detection. (b) Grid-Based Routing: Finds optimal paths but requires numerous search steps and can suffer from precision errors due to limited grid resolution. (c) Shape-Based Routing: Uses direct geometric calculations to minimize search steps and avoid precision errors, but often results in sub-optimal routing due to its incomplete search space.

and develop an universal PCB routing algorithm for large-scale PCBs. The key idea is to leverage the path searching capabilities of grid-based methods to ensure stable routing paths, while also utilizing the geometric modeling techniques of shape-based methods to achieve high precision and efficiency. To achieve this goal, we must address two primary challenges:

- Designing a routing search algorithm that avoids fixed grids to eliminate precision errors while maintaining necessary search space for optimal routing path.
- Ensuring DRC compliance and optimal path selection in the routing process requires frequent precise obstacle detection, leading to significant time consumption, especially for large-scale PCBs.

In this paper, we propose MegaRoute, an universal routing algorithm for large-scale PCBs. To the best of our knowledge, this work is the first in the literature to scale automated routing to mainboard PCBs. MegaRoute introduces an adaptive step-size search algorithm that efficiently explores routing paths by adjusting step-sizes based on design rules and surrounding obstacles. This approach eliminates the need for fixed grids, avoids precision errors, and maintains necessary search space for optimal routing path, as shown in Figure 2. It illustrates the relationship and differences between MegaRoute, grid-based, and shape-based methods. Additionally, MegaRoute develops a shape-based obstacle detection method, using direct geometric calculations to provide a high-precision foundation for the adaptive step-size search. Finally, MegaRoute incorporates routing optimization to enhance overall routing quality while

accelerating the search process. MegaRoute significantly improves routability, DRC compliance, and efficiency for large-scale PCB routing.

Our main contributions are summarized as follows:

- We propose MegaRoute, an universal routing algorithm for large-scale PCBs with routing results comparable to advanced commercial tools. To the best of our knowledge, this work is the first in the literature to successfully scale automated routing to mainboard PCBs.
- We introduce a adaptive step-size search algorithm that adjusts exploration steps based on design rules and surrounding obstacles. This approach eliminates the need for fixed grids, avoids precision errors, and maintains necessary search space for optimal routing path.
- We present a shape-based obstacle detection method to provide efficient geometric calculations for the adaptive step-size search algorithm, ensuring strict DRC compliance and enhancing efficiency for large-scale PCBs.
- We conduct extensive experiments on hundreds of real-world PCBs, including mainboards with thousands of nets. MegaRoute achieves over 98% routability with consistently DRC-free results, outperforming various commercial tools and state-of-the-art methods.

II. RELATED WORK

A. Grid-Based Methods

Grid-based methods rely on fixed grids to search for routing paths by using minimum-cost grid search algorithms. He et al.

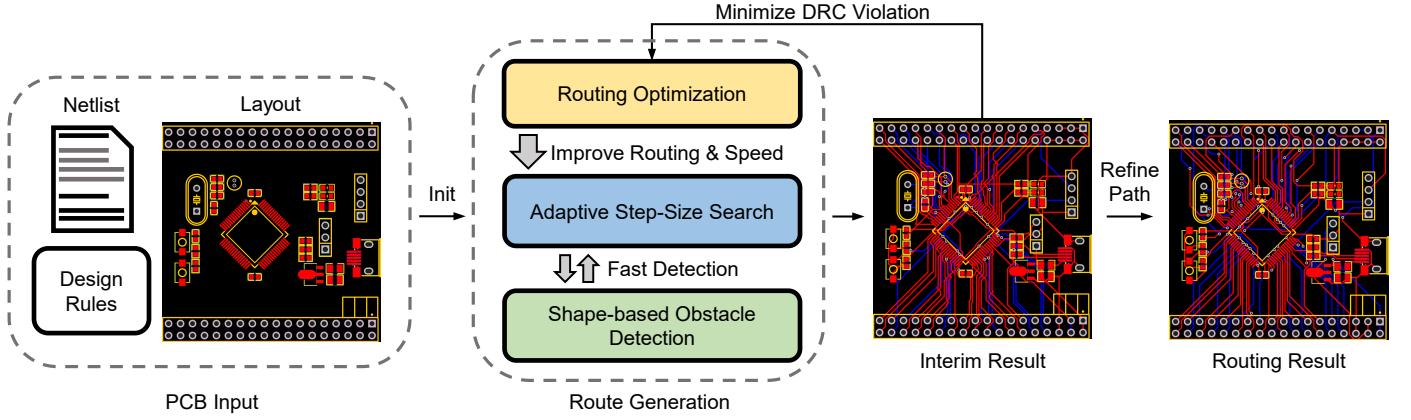


Fig. 3. Overall workflow of our proposed MegaRoute automated PCB routing algorithm

[14] and Lin et al. [16] propose A*-based universal routing approaches, but these suffer from low routability and excessive via usage. Li et al. [13] and FreeRouting [15] introduce optimization algorithms to address these issues, but the overall efficiency remains too low, especially for large-scale PCBs.

B. Shape-Based Methods

Shape-based methods represent PCB layout geometrically, offering higher precision and flexibility. Commercial tools like EasyEDA [20] provide basic shape-based routing, but struggle with complex designs. Advanced commercial tools like Allegro X [18], Altium Designer [19], and ELECTRA [17] offer sophisticated shape-based algorithms. However, modeling components with special angles, such as 45-degree pins, often leads to DRC violations and reduced routability.

III. BACKGROUND OF PCB ROUTING

The task of PCB routing involves connecting the component pins in a PCB layout according to the netlist, while adhering to design rules such as wire width, via size, and clearances. A routing result includes paths that connect all nets, consisting of wires and vias. Wires are line sections used for connections within the same layer, and vias are used to change wires between layers.

We can thus formulate the problem of PCB routing: Given a PCB layout \mathbf{L} , a netlist \mathbf{N} , and design rules \mathbf{D} , the objective of an automated routing algorithm $f(\cdot)$ is to generate a set of paths \mathbf{P} within \mathbf{L} that maximize routability R —connecting as many nets in \mathbf{N} as possible—while minimizing wire length W and via count V , and ensuring compliance with all design rules \mathbf{D} . Formally, this can be expressed as:

$$\begin{aligned} \mathbf{P} &= f(\mathbf{L}, \mathbf{N}, \mathbf{D}) \\ \max \quad R(\mathbf{P}) - W(\mathbf{P}) - V(\mathbf{P}) \\ \text{s.t.} \quad \mathbf{P} &\in \mathcal{S}(\mathbf{D}) \end{aligned} \quad (1)$$

where $\mathcal{S}(\mathbf{D})$ denotes the set of all paths that satisfy the design rules \mathbf{D} .

IV. MEGAROUTE

MegaRoute is an universal PCB routing method, as illustrated in Figure 3, consisting of three main components: adaptive step-size search, shape-based obstacle detection, and routing optimization. The adaptive step-size search algorithm is used to explore routing paths by dynamically adjusting the exploration steps based on design rules and surrounding obstacles, improving efficiency while ensuring sufficient search space for optimal paths. Shape-based obstacle detection provides precise and efficient real-time obstacle detection for the adaptive step-size search algorithm, ensuring strict DRC compliance during routing. Finally, routing optimization accelerates the routing process by using cost guidance, and employs re-routing strategies to minimize DRC violations. Together, these components enable MegaRoute to route large-scale PCBs with high routability, efficiency, and strict DRC compliance.

A. Adaptive Step-Size Search

Routing each PCB net can be decomposed into several two-terminal path search problems [13]. This section introduces five exploration methods specifically designed for PCB routing to achieve precision and efficient path searches, as shown in Figure 4. The algorithm explores new nodes with an adaptive step size from the start node $\mathbf{n}_{\text{start}} = (x_{\text{start}}, y_{\text{start}}, z_{\text{start}})$ and creates edges between the current node $\mathbf{n}_{\text{cur}} = (x_{\text{cur}}, y_{\text{cur}}, z_{\text{cur}})$ and newly discovered nodes until the end node $\mathbf{n}_{\text{end}} = (x_{\text{end}}, y_{\text{end}}, z_{\text{end}})$ is reached. The x and y coordinates represent positions within the same PCB layer, while z denotes the layer index. Each exploration method for each step is detailed below:

1) *Obstacle-Aware Exploration*: Obstacle-aware exploration adjusts the step size δ based on the current net's wire width rules and the presence of obstacles detected in real-time as described in Section IV-B. The step size decreases near obstacles and increases in clear spaces to expedite the search process.

The step size δ is determined as follows:

$$\delta = \begin{cases} w_{\text{net}}, & \text{if } d_{\text{obstacle}} \leq \beta \times w_{\text{net}} \\ \beta \times w_{\text{net}}, & \text{if } d_{\text{obstacle}} > \beta \times w_{\text{net}} \end{cases} \quad (2)$$

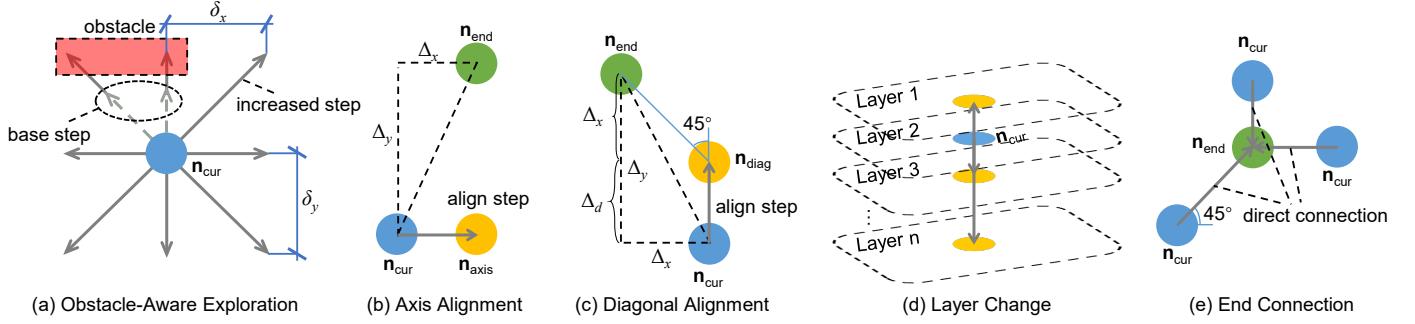


Fig. 4. Illustration of the exploration methods of adaptive step-size search algorithm.

where β is a hyperparameter, w_{net} is the current net's wire width, and d_{obstacle} is the distance to the nearest obstacle. By adjusting δ based on real-time obstacle detection, the algorithm balances efficiency and precision. The set of extended nodes $N_{\text{o-a}}$ is defined as:

$$N_{\text{o-a}} = \{(x_{\text{cur}} + \delta_x, y_{\text{cur}} + \delta_y) \mid (\delta_x, \delta_y) \in \mathcal{C}\} \quad (3)$$

where $\mathcal{C} = \{-\delta, 0, \delta\}^2 \setminus \{(0, 0)\}$. The set $\{-\delta, 0, \delta\}^2$ represents the Cartesian product, including all combinations of step sizes for x and y .

2) *Axis Alignment*: Axis alignment aligns the current node with the end node along the x or y axis, depending on which direction has the smaller distance. This avoids scenarios where the step size does not precisely reach the end node. Let n_{end} be the end node and $\Delta x = x_{\text{end}} - x_{\text{cur}}$, $\Delta y = y_{\text{end}} - y_{\text{cur}}$. The alignment node n_{axis} is defined as:

$$n_{\text{axis}} = \begin{cases} (x_{\text{cur}} + \Delta x, y_{\text{cur}}) & \text{if } |\Delta x| < |\Delta y| \\ (x_{\text{cur}}, y_{\text{cur}} + \Delta y) & \text{if } |\Delta y| \leq |\Delta x| \end{cases} \quad (4)$$

This alignment is performed only if both $|\Delta x|$ and $|\Delta y|$ are greater than zero.

3) *Diagonal Alignment*: Diagonal alignment makes fine adjustments based on the difference between the x and y components of the end node direction. Let $\Delta d = ||\Delta x| - |\Delta y||$. The diagonal alignment node n_{diag} is defined as:

$$n_{\text{diag}} = \begin{cases} (x_{\text{cur}}, y_{\text{cur}} + \Delta d \cdot \text{sign}(\Delta y)) & \text{if } |\Delta x| < |\Delta y| \\ (x_{\text{cur}} + \Delta d \cdot \text{sign}(\Delta x), y_{\text{cur}}) & \text{if } |\Delta y| < |\Delta x| \end{cases} \quad (5)$$

This alignment is performed only if Δd is greater than zero and less than the step size.

4) *Layer Change*: Layer change exploration enables transitions between different PCB layers. Let Z be the set of all layers. The layer change nodes are defined as:

$$N_{\text{layer_change}} = \{(x_{\text{cur}}, y_{\text{cur}}, z) \mid z \in Z, z \neq z_{\text{cur}}\} \quad (6)$$

5) *End Connection*: A direct connection is made to the end node when the current node is within a distance smaller than the step size and aligned on the x or y axis or diagonally with the end node. The end connection node $n_{\text{end_conn}}$ is defined as:

$$n_{\text{end_conn}} = \begin{cases} (x_{\text{cur}} + \Delta x, y_{\text{cur}}) & \text{if } y_{\text{cur}} = y_{\text{end}} \\ (x_{\text{cur}}, y_{\text{cur}} + \Delta y) & \text{if } x_{\text{cur}} = x_{\text{end}} \\ (x_{\text{cur}} + \Delta x, y_{\text{cur}} + \Delta y) & \text{if } |\Delta x| = |\Delta y| \end{cases} \quad (7)$$

This connection is performed only if the current node is on the same layer as the end node and the distance to the end node is smaller than the step size.

By using these exploration methods to dynamically adjust the search space, MegaRoute avoids fixed grids to improve precision and efficiency, while maintaining sufficient search space for optimal paths even for complex and large-scale PCBs.

B. Shape-based Obstacle Detection

This section details the obstacles encountered in PCB routing and the shape-based distance calculations used to detect the obstacles in the search process, ensuring adherence to design rules.

1) *Types of Obstacles*: In PCB layout, obstacles can be categorized and represented by geometric objects as follows:

- **Pins**: Polygons or circles, pins define the connection nodes for components.
- **Boundaries**: Lines, boundaries define the edges of the PCB and other restricted areas.
- **Wires**: Lines, wires form the electrical connections between pins and other components within the same layer.
- **Vias**: Circles, vias allow connections between different layers of the PCB.

Each shape element is associated with several PCB layers and a specific net. Only shapes from different nets but the same layer are considered obstacles to each other.

2) *Obstacle Detection in Adaptive Step-Size Search*: In the adaptive step-size search process, obstacle detection ensures compliance with clearance rules by calculating distances between various geometric shapes on the PCB layout. During same-layer explorations—such as obstacle-aware exploration, axis alignment, diagonal alignment, or end connection—distances between new wires and existing elements (wires, vias, pins, boundaries) are computed to ensure that the required clearance is maintained. In layer-change scenarios, the via's multi-layer penetration requires that its distance to other shapes on all layers be calculated, ensuring clearance compliance across layers.

The specific calculations depend on the shapes involved. For circular shapes like vias or pins, the distance between centers is calculated by subtracting the sum of their radii. For interactions between circles and lines, the perpendicular distance from the circle's center to the line is calculated, minus the circle's radius and half the line's width. In line-to-line cases, the minimum distance between line segments is calculated, subtracting half the width of each line. For polygons, the minimum distance from polygon edges to other shapes is calculated.

C. Routing Optimization

In this section, we describe several optimization techniques designed to accelerate the search process, enhance overall routability and DRC compliance, and refine paths for improved electrical performance.

1) *Cost Guidance*: We propose a cost-based priority search strategy to guide the process by prioritizing the exploration of nodes with the lowest routing costs, reducing unnecessary computations. This method combines multiple routing costs to improve efficiency without compromising solution quality. The calculation of each routing cost is as follows:

- **Length Cost**: We use A*-like historical and estimated costs [21], where the historical cost is the wire length from the start node to the current node, and the estimated cost using the 135° angle constraint distance is computed as follows:

$$h(n_{\text{cur}}, n_{\text{end}}) = \Delta_{\max} + (\sqrt{2} - 1) \times \Delta_{\min}, \quad (8)$$

where Δ_{\max} and Δ_{\min} are the maximum and minimum of $|\Delta x|$ and $|\Delta y|$, respectively.

- **DRC Cost**: Based on shape-based obstacle detection, this cost reflects the number of DRC violations, heavily weighted to avoid them.
- **Layer Change Cost**: Additional cost for layer change explorations, typically proportional to the via diameter to avoid excessive drilling.

Additionally, to prevent the number of search nodes from growing explosively in dense obstacle regions, we set a density constraint for exploration nodes. Specifically, if an existing node is within a threshold distance d_{\min} of a new node, the new node will be ignored. This ensures a sparse distribution of exploration nodes and avoids unnecessary computations.

2) *DRC Minimization*: We observed that the sequential nature of net routing often leads to inefficient use of space. Later-routed nets must navigate around earlier-routed ones, resulting in excessive wire lengths and via counts if insufficient space is reserved. Inspired by negotiation-based methods [22], we divide the routing process into multiple iterations. We gradually increase the DRC cost weight from low to high using a logistic function. For each iteration, the weight α^i is calculated as:

$$\alpha^i = L(f(2\mu) - f(0))^{-1} (f(i) - f(0)) \quad (9)$$

where $f(x) = (1 + e^{-k(x-\mu)})^{-1}$, i is the current iteration number, L is the maximum value, k is the slope parameter, and μ is the midpoint.

V. EXPERIMENTS

A. Setup

MegaRoute is implemented in C++ and compiled with mingw64, running on a Windows 11 workstation with 64GB memory and an i9-13900K processor. We use the EasyEDA Pro project format [20] for testing by exporting data inputs for other methods and commercial tools through its conversion tool. Performance metrics are evaluated using EasyEDA. The adaptive step parameter is $\beta = 5$, density threshold d_{\min} is 1 mil, and for DRC minimization, we set $L = 200$, $k = 0.4$, $\mu = 10$, with a maximum of 20 iterations. We collect a PCB dataset of 101 real-world PCB designs from the open-source hardware platform OSHWHub¹. The designs vary in scale, with net counts from 7 to 1564, layer counts from 2 to 6, pin counts from 14 to 3161, connection counts from 6 to 2326, and areas from 42 to 37845 mm². We divide the dataset into two groups based on pin count, with 1000 pins as the threshold: 14-963 pins (93 cases) and 1011-3161 pins (8 cases). This division aims to evaluate the performance at different PCB scales.

B. Comparison Methods

We compare the performance with 8 state-of-the-art PCB routing methods. These include 4 shape-based methods: Altium Designer [19], Allegro X [18], ELECTRA [17], and EasyEDA [20], as well as 4 grid-based methods: Freerouting [15], Lin's Algorithm [16], He's Algorithm [14], and FanoutNet [13]. As Lin's Algorithm [16] and He's Algorithm [14] are not publicly available, we have implemented these methods ourselves for the comparison. Notably, Altium Designer [19], Allegro X [18], and ELECTRA [17] are currently the most advanced commercial routing tools globally.

C. Evaluation Metrics

We evaluate each routing method based on following metrics: *Routability* (ratio of successfully routed connections), *DRC* (design rule check violations), *WL* (total wire length), *VC* (total via count), *meanWL* (mean wire length per routed connection), *meanVC* (mean via count per routed connection), and *Runtime* (total routing time). For large-scale PCBs with more than 1000 pins, a 24-hour runtime limit is imposed to manage excessive computation times.

D. Performance Comparisons and Analysis

The performance comparison between MegaRoute and other state-of-the-art methods is shown in Table I.

- **Routability & DRC Compliance**: MegaRoute consistently achieves over 98% routability with zero DRC violations across PCBs of varying sizes, outperforming other methods. Grid-based methods like Freerouting and FanoutNet show reduced routability and more DRC violations on larger PCBs. Shape-based methods and commercial tools, such as Allegro X and Altium Designer, offer excellent routability but higher DRC violations compared to MegaRoute. This demonstrates MegaRoute's improvements in both routability and DRC compliance.

¹OSHWHub open source hardware platform: <https://oshwhub.com/>

TABLE I
PERFORMANCE COMPARISONS FOR DIFFERENT PCB ROUTING METHODS.

| Method | Routability | DRC | meanWL (mm) | meanVC | WL (mm) | VC | Runtime (min) |
|-----------------------------|---------------|----------|-------------|--------|------------|------|---------------|
| Pin Count < 1000 | | | | | | | |
| Altium Designer (2024) [19] | 0.9768 | 16 | 11.0143 | 0.3299 | 3082.2093 | 92 | 0.4970 |
| Allegro X (2024) [18] | 0.9378 | 29 | 11.2441 | 0.4186 | 2952.4030 | 109 | 0.2410 |
| ELECTRA (2024) [17] | 0.8982 | 37 | 15.2119 | 0.4934 | 3506.2589 | 112 | 5.5027 |
| EasyEDA (2024) [20] | 0.5940 | 11 | 20.0243 | 0.8056 | 1304.2043 | 54 | 0.9585 |
| Freerouting (2023) [15] | 0.9430 | 13 | 9.8172 | 0.3606 | 2729.9455 | 100 | 18.3703 |
| FanoutNet (2023) [13] | 0.9331 | 29 | 8.6597 | 0.6248 | 2224.9598 | 163 | 8.3916 |
| Lin's (2021) [16] | 0.9245 | 19 | 9.2243 | 1.2558 | 1878.6550 | 237 | 6.3855 |
| He's (2023) [14] | 0.8969 | 23 | 8.3019 | 1.8330 | 1690.7901 | 356 | 31.9275 |
| MegaRoute (ours) | 0.9885 | 0 | 9.4917 | 0.4609 | 2718.5300 | 133 | 3.0827 |
| Pin Count ≥ 1000 | | | | | | | |
| Altium Designer (2024) [19] | 0.9941 | 52 | 13.3009 | 0.3753 | 12523.9648 | 431 | 6.1478 |
| Allegro X (2024) [18] | 0.9894 | 26 | 12.8357 | 0.3973 | 12049.2910 | 421 | 1.2458 |
| ELECTRA (2024) [17] | 0.9623 | 66 | 20.3863 | 0.3926 | 22544.1129 | 395 | 202.1396 |
| EasyEDA (2024) [20] | 0.5734 | 69 | 9.8953 | 0.4574 | 5452.3798 | 231 | 8.3431 |
| Freerouting (2023) [15] | 0.9657 | 51 | 11.7774 | 0.3592 | 10470.1995 | 357 | >24-hours |
| FanoutNet (2023) [13] | 0.9390 | 98 | 10.4682 | 0.6531 | 8665.8335 | 576 | >24-hours |
| Lin's (2021) [16] | 0.9126 | 69 | 12.2551 | 0.9623 | 9882.5633 | 839 | >24-hours |
| He's (2023) [14] | 0.8864 | 84 | 11.3436 | 1.2993 | 8857.8680 | 1088 | >24-hours |
| MegaRoute (ours) | 0.9943 | 0 | 11.2119 | 0.4151 | 10493.3533 | 453 | 46.3188 |

- Efficiency:** MegaRoute shows comparable efficiency to commercial tools on PCBs with fewer than 1000 pins. On larger PCBs with over 1000 pins, while MegaRoute is slower than Allegro X and Altium Designer, it still significantly outperforms other methods. This indicates that MegaRoute maintains high efficiency in improving both routability and DRC compliance.

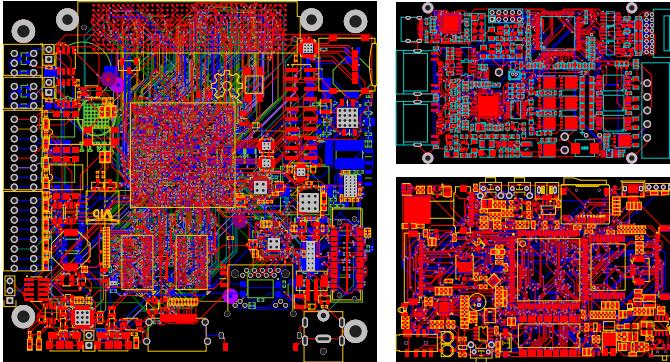


Fig. 5. Case study of mainboards.

E. Case Study

Figure 5 shows the visual results of MegaRoute on three large-scale PCB designs: Xilinx Zynq XC7Z030 mainboard, AF Motor Dev Board, and STM32F767 Full Screen Dev Board. It can be seen that MegaRoute achieves stable routing on large, high-density, and complex PCBs, with the routing results adhering to the 135° routing rule.

F. Ablation Study

To evaluate the impact of each component in MegaRoute, we conduct an ablation study by individually removing the

TABLE II
ABLATION STUDY

| Method | Routability | Runtime (min) |
|--|---------------|---------------|
| w/o Adaptive Step-Size ($\delta = 1$ mil) | 0.9861 | 12.3002 |
| w/o Cost Guidance | 0.9747 | 10.7732 |
| w/o DRC Minimization | 0.9092 | 1.3748 |
| MegaRoute | 0.9890 | 6.5073 |

adaptive step-size, cost guidance, and DRC minimization features, as shown in Table II. The results, which represent the average performance across all cases, indicate that omitting any component results in varying degrees of performance degradation. Specifically, the absence of cost guidance notably affects efficiency, while removing DRC minimization significantly impacts routability. These findings highlight the importance and effectiveness of each design element in our approach.

VI. CONCLUSION

In this paper, we introduce MegaRoute, an universal routing algorithm designed for large-scale PCBs, achieving both high routability and efficiency while maintaining strict DRC compliance. Our extensive experiments with hundreds of real-world PCBs demonstrate that MegaRoute outperforms various commercial tools and state-of-the-art methods. Future work will focus on enhancing performance further and addressing more complex design rules such as differential pairs.

VII. ACKNOWLEDGEMENTS

This work is partially supported by the National Natural Science foundation of China under Grant No. 62441233, 62002106, Huawei Terminal Co., Ltd, Yi Wu, Shejun Sun, and Shenzhen JLC Electronics Co., Ltd.

REFERENCES

- [1] T. Yan, Q. Ma, and M. D. Wong, “Advances in pcb routing,” *IPSJ Transactions on System and LSI Design Methodology*, vol. 5, pp. 14–22, 2012.
- [2] T. Yan and M. D. F. Wong, “Recent research development in pcb layout,” in *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2010, pp. 398–403.
- [3] Y. Goh, D. Jung, G. Hwang, and J.-M. Chung, “Consumer electronics product manufacturing time reduction and optimization using ai-based pcb and vlsi circuit designing,” *IEEE Transactions on Consumer Electronics*, 2023.
- [4] C. Chen, D. Lin, R. Wei, Q. Liu, Z. Zhu, and J. Chen, “Efficient global optimization for large scaled ordered escape routing,” in *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, 2023, pp. 535–540.
- [5] J.-T. Yan, “Ordered escape routing via assignment of routability-driven detouring paths,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 13, no. 4, pp. 451–464, 2023.
- [6] Q. Liu, Q. Tang, J. Chen, C. Chen, Z. Zhu, H. He, J. Chen, and Y.-W. Chang, “Disjoint-path and golden-pin based irregular pcb routing with complex constraints,” in *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2023, pp. 1–6.
- [7] S.-T. Lin, H.-H. Wang, C.-Y. Kuo, Y. Chen, and Y.-L. Li, “A complete pcb routing methodology with concurrent hierarchical routing,” in *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2021, pp. 1141–1146.
- [8] J.-T. Yan, “Single-layer obstacle-aware multiple-bus routing considering simultaneous escape length,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 12, no. 6, pp. 902–915, 2022.
- [9] C.-H. Hsu, S.-C. Hung, H. Chen, F.-K. Sun, and Y.-W. Chang, “A dag-based algorithm for obstacle-aware topology-matching on-track bus routing,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 3, pp. 533–546, 2021.
- [10] J.-T. Yan, “Bus assignment considering flexible escape routing for layer minimization in pcb designs,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 8, pp. 2699–2713, 2022.
- [11] ———, “Fuzzy-clustering-based circular topological via minimization in pcb designs,” *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 5, pp. 1023–1036, 2021.
- [12] ———, “Layer assignment of buses and nets with via-count constraint in high-speed pcb designs,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 3, pp. 512–525, 2019.
- [13] H. Li, J. Zhang, N. Xu, and M. Liu, “Fanoutnet: a neuralized pcb fanout automation method using deep reinforcement learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8554–8561.
- [14] Y. He, H. Li, G. Luo, and F. S. Bao, “Two-stage pcb routing using polygon-based dynamic partitioning and mcts,” in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2023, pp. 1–2.
- [15] FreeRouting, “FreeRouting: Advanced PCB Auto-Router,” <https://github.com/freerouting/freerouting>, 2023, version 1.9.0.
- [16] T.-C. Lin, D. Merrill, Y.-Y. Wu, C. Holtz, and C.-K. Cheng, “A unified printed circuit board routing algorithm with complicated constraints and differential pairs,” in *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, 2021, pp. 170–175.
- [17] Konekt, “ELECTRA,” <https://konekt.com/>, 2024, version 8.10.
- [18] Cadence Design Systems, “Cadence Allegro,” <https://www.cadence.com/>, 2024, version 17.4.
- [19] Altium, “Altium Designer,” <https://www.altium.com/>, 2024, version 24.0.
- [20] EasyEDA, “EasyEDA Pro Edition,” <https://easyeda.com/>, 2024, version 2.1.25.
- [21] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [22] L. McMurchie and C. Ebeling, “Pathfinder: A negotiation-based performance-driven router for fpgas,” in *Proceedings of the 1995 ACM third international symposium on Field-programmable gate arrays*, 1995, pp. 111–117.