V-GR: 3D Global Routing with Via Minimization and Multi-Strategy Rip-up and Rerouting

Ping Zhang*, Pengju Yao*, Xingquan Li†, Bei Yu‡, and Wenxing Zhu*
*Fuzhou University; †Pengcheng Laboratory, China; ‡The Chinese University of Hong Kong
{215420017, 211027184, wxzhu}@fzu.edu.cn; fzulxq@gmail.com; byu@cse.cuhk.edu.hk

2024 29th Asia and South Pacific Design Automation Conference (ASP-DAC) | 979-8-3503-9354-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ASP-DAC58780.2024.10473939

***Abstract*—In VLSI, a large number of vias may reduce manufacturability, degrade circuit performance, and increase layout area required for interconnection. In this paper, we propose a 3D global router V-GR, which considers minimizing the number of vias. V-GR uses a modified via-aware routing cost that considers the impact of wire density on the via. This cost function is more sensitive to the number of vias. Meanwhile, a novel multi-strategy rip-up & rerouting framework is developed for V-GR to solve the overflowed net, effectively optimizing wire length, overflow, and minimizing the number of vias. The proposed framework first leverages two proprietary routing techniques, namely the 3D monotonic routing and 3D 3-via-stack routing, to control the number of vias and reduce overflow. Additionally, the framework incorporates an RSMT-aware expanded source 3D maze routing algorithm to build routing paths with shorter wire length. Experimental results on the ICCAD'19 contest benchmarks show that, V-GR achieves high-quality results, reducing vias by 8% and overflow by 7.5% in the global routing phase. Moreover, to achieve a fair comparison, TritonRoute is used to conduct detailed routing, and Innovus is used to evaluate the final solution. Comparison shows that V-GR achieves 4.7% reduction in vias and 8.7% reduction in DRV, while maintaining almost the same wire length.**

***Index Terms*—global routing, via, rip-up & rerouting, maze routing.**

I. Introduction

Routing is a core stage in VLSI physical design, which is typically divided into two steps: global routing and detailed routing. In global routing, nets are routed over a coarse-grained grid graph to determine within which areas each net should be routed. In detailed routing, the routing scheme determined by global routing is used as a guide to achieve a specific routing result. A good global router should have shorter wire length, less number of vias and less amount of overflow, and at the same time can guide detailed routing to achieve high routability. Therefore, the final routing quality largely depends on the performance of global routing.

Vias are interconnections between different routing metal layers. A large number of vias can reduce manufacturing yield, cause circuit performance degradation, and increase layout area required for interconnections [1], [2]. In VLSI physical design, meeting the DFM (Design for Manufacturability) constraints is essential, and these constraints often include strict requirements regarding vias. As a result, it becomes crucial to minimize the number of vias during the VLSI physical design process while ensuring that the routability of the design is not compromised. Global routing captures precise pin and congestion information, and thereby enables direct consideration of the via minimization issue.

Most academic global routers use pattern routing to obtain initial solution quickly. However, the lack of candidate scheme for pattern routing results in significantly high overflow for the initial solution. Hence rip-up & rerouting is usually employed with various strategies to solve overflowed nets. A well-designed rip-up & rerouting framework can effectively optimize the wire length, as well as address the issues of vias and overflow present in the initial solution. However, existing rip-up and rerouting techniques do not fully consider via minimization. In this paper, we shall present a 3D global router with via minimization and multi-strategy rip-up and rerouting.

This work was supported by the National Natural Science Foundation of China under Grant 62174033.
* Corresponding author: wxzhu@fzu.edu.cn

A. Related Work

There are two main categories for global routing: two-dimensional (2D) global routing followed by layer assignment, and three-dimensional (3D) global routing. 2D method compresses routing layers onto a 2D plane, builds a 2D routing graph and routes on the 2D plane followed with layer assignment. While, 3D approach routes directly on multiple metal layers. These global routers pay more attention to routability and minimizing the wire length without overflow, but the number of vias is not fully considered.

In order to balance runtime and routing performance, most global routers belong to the first category, such as NTHU-Route 2.0 [3], NCTU-GR 2.0 [4], SPRoute 2.0 [5] and FastRoute 4.0 [2]. They first decompose each multi-pin net into two-pin nets, and then use pattern routing or monotonic routing to quickly obtain an initial solution. Then, maze routing is used to rip up & reroute congested nets to optimize the initial routing result. To reduce the number of vias, FastRoute 4.0 [2] used 3-bend routing and maze routing in sequence instead of only using the maze routing. After rip-up & rerouting, layer assignment was performed and the result was restored to 3D grid graph, which determines the final number of vias.

During 2D routing, only the number of turns is known on a routing path, hence we cannot obtain accurate via information before layer assignment, which may lead to an increase in the number of final vias. Therefore, one significant challenge of 2D global routing is that, it cannot effectively control the number of vias. In contrast, another routing strategy, 3D global routing, can effectively deal with this issue.

3D global routing routes nets directly on a 3D grid graph. FGR [6] used maze routing to directly rip up & reroute nets, based on the discrete Lagrangian cost framework. GRIP [7] determined 3D routing candidate patterns for each net in advance, and then used ILP for optimal selection. MGR [8] used pattern routing and layer assignment to obtain a 3D initial solution, and then adopted 3D maze routing to rip up & reroute the nets in congestion areas. CUGR [9] used 3D pattern routing based on dynamic programming to obtain an initial routing, and used multi-level 3D maze routing for rip-up and rerouting to obtain a final global routing solution.

Note that, the rip-up & rerouting stages of 3D global routers utilize a single-strategy rip-up & rerouting framework, employing solely the maze routing algorithm [8], [9], [10], [11], [12]. Although this approach effectively reduces overflow, it may lead to an increase in the number of generated vias. Therefore, it would be desirable to propose a novel rip-up and rerouting scheme which can minimize the number of vias while optimizing wirelength and overflow.

B. Our Contributions

By considering the pros and cons of existing 2D and 3D global routing algorithms, this work proposes a novel via-aware 3D global router V-GR to handle via minimization in 3D space throughout the entire global routing flow. The main contributions of this work are summarized as follows:

- We propose a modified via-aware routing cost function that takes into account the effect of wire density on the via. This function can guide the proposed global router to generate a global routing solution with fewer vias.

- We propose a novel multi-strategy rip-up & rerouting framework, comprising two stages: local and global rip-up & rerouting. This framework effectively optimizes overflow while minimizing increases in wire length and the number of vias.
- During the local rip-up & rerouting stage, a 3D monotonic routing algorithm is proposed to generate a routing path in a bounding-box, with the aim of reducing overflow and the number of vias, while maintaining an optimized wire length. During the global rip-up & rerouting stage, a 3D 3-via-stack routing algorithm and an RSMT-aware expanded source 3D maze routing algorithm are proposed to reduce overflow while minimizing the increase in the number of vias and wire length.
- Compared with the CUGR, experimental results show that V-GR achieves a comprehensive improvement in the global routing stage. For generating detailed routing solutions, we employ TritonRoute. Subsequently, we evaluate these solutions using Innovus, which reports a 4.7% reduction in vias and an 8.7% reduction in DRV, while maintaining almost the same wire length.

The rest of the paper is organized as follows. Section II introduces the global routing problem and the existing cost scheme. Section III presents detailed design flow and the modified via-aware routing cost used in our global routing. Section IV describes our multi-strategy rip-up & rerouting framework, and its internal core algorithms. Experimental results are provided in Section V, and Section VI concludes our work.

II. Preliminaries

In this section, we introduce the 3D global routing problem and the cost scheme that will be modified and used in our global routing.

A. 3D Global Routing Problem

In global routing, the chip area is partitioned into an array of rectangular area cells called G-cells, and modeled as a 3D grid graph $G(V, E)$ with the set of nodes V and the set of edges E, as shown in Fig. 1(c). Each G-cell corresponds to a node $v \in V$ in the grid graph. Between two adjacent G-cells in the same metal layer, there exists a global edge which corresponds to an edge $e \in E$ in the grid graph. The capacity $cap(e)$ of a grid edge indicates the maximum number of wires that can use the grid edge, and the demand of an edge, $d(e)$, denotes the number of tracks currently passed through by wires. The resource of an edge, $r(e)$, is the portion of the capacity that can still be utilized while routing. Furthermore, the overflow of an edge represents the amount of exceeding demand.

The global routing problem is: given a grid graph and a set of nets composed of pins, find a routing scheme for every net to connect its pins such that the total overflow is minimized. Additionally, minimizing the wire length and the number of vias are also important goals.

Fig. 1(a) shows a two-pin net with its feasible 2D global routing path. Figs. 1(b) and 1(c) give two different scenarios that may be obtained when the routing path is restored to 3D. They respectively generate 5 and 3 vias under the same wire length. Since 2D routing only knows how many turns there are in the path, it cannot accurately obtain the number of vias, which leads to different numbers of vias in the 3D routing paths with the same 2D routing result. Therefore, in order to effectively realize minimizing the number of vias, the proposed algorithms in this work are all in 3D space.

B. Cost Scheme in Existing Global Routing

In global routing, the costs of edges have significant impact on the pathfinding process. Therefore, a cost scheme must be able to accurately reflect the resource quantities of edges and vias in the grid graph and can effectively guide the routing, which is essential for global routing. CUGR [9] has designed a carefully considered cost scheme that takes into account wire length, as well as the possibility of overflow. This

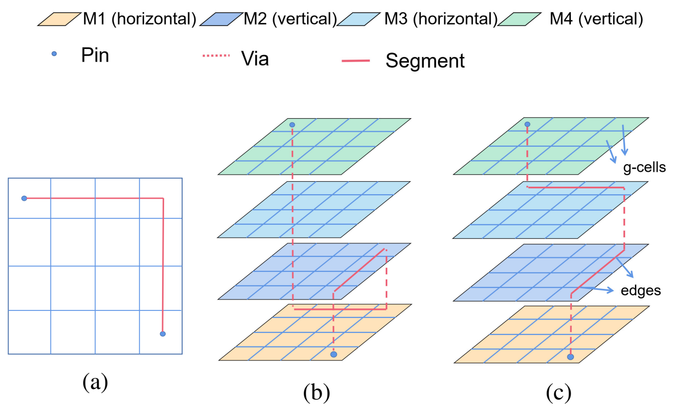


Fig. 1: Two 3D global routing results with the same 2D routing path.

scheme minimizes both the wire length and the number of vias, while ensuring that overflow does not occur.

A connection path of a net consists of both edges and vias. In CUGR [9], the total cost of a path P is:

$$cost(P) = \sum_{e(u,v) \in P} cost_w(u,v) + \sum_{via(u,u') \in P} cost_v(u,u'),$$

where $cost_w(u,v)$ is the cost of wire edge (u,v), and $cost_v(u,u')$ is the cost of via (u,u'). Further, the wire edge cost is computed by the wire length cost and the congestion cost as follows:

$$\begin{aligned} cost_w(u,v) &= wl(u,v) + eo(u,v) \times \lg(u,v), \\ eo(u,v) &= wl(u,v) \times \frac{d(u,v)}{c(u,v)} \times uoc, \\ \lg(u,v) &= (1.0 + \exp(slope \times r(u,v)))^{-1}, \end{aligned}$$

where $wl(u,v)$ is the wire length of the wire edge (u,v), $eo(u,v)$ is the expected overflow cost, $\lg(u,v)$ is the sigmoid function of $r(u,v)$, and uoc is a given constant which is the unit length overflow cost. $slope$ is an adjustable parameter in the sigmoid function, which defines sensitivity to overflow of the global router. Calculation of the via cost is similar to that of the wire edge:

$$\begin{aligned} cost_v(u,u') &= uvc \times (1 + \lg(u) + \lg(u')), \\ \lg(u) &= (1.0 + \exp(slope \times r(u)))^{-1}, \end{aligned}$$

where uvc gives the unit via cost, $r(u)$ is the half of the total resource of two edges connecting u.

The above cost scheme will be adopted in this work. However, we found that an edge with higher wire density has a higher probability of producing vias, so we will modify the via cost for our work.

III. Flow of our Global Router and Routing Cost

In this section, we first introduce the detailed design flow of the proposed global router V-GR. After that, we introduce the modified via-aware routing cost used in our global routing.

A. Flow of our Global Router

The proposed global router V-GR consists of two parts: initial routing and multi-strategy rip-up & rerouting, as shown in Fig. 2. For the initial routing, we first use FLUTE [13] to generate an RSMT for each net, which decomposes each multi-pin net into a set of two-pin nets. Subsequently, we apply 3D pattern routing [9] with the cost scheme presented in Subsection III-B to route all nets and obtain an initial 3D global routing solution. The multi-strategy rip-up & rerouting is divided into local and global rip-up & rerouting stages. Each stage performs rip-up & rerouting of nets in ascending order of net area.

The local rip-up & rerouting is first executed using the 3D monotonic routing proposed in Subsection IV-A once to reduce the total overflow

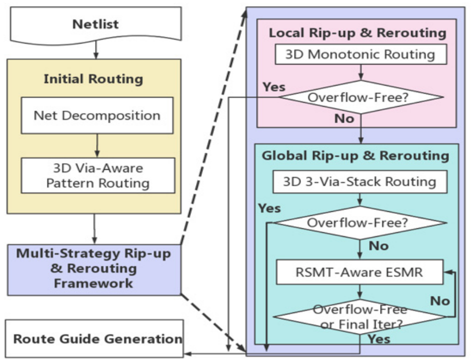


Fig. 2: Design flow of our global router V-GR.

while reducing the number of vias and maintaining the wire length. Then in global rip-up & rerouting, we first perform the 3D 3-via-stack routing proposed in Subsection IV-B once, which is capable of detouring, controlling the number of vias, and can reduce congestion similar to maze routing. And then, we iteratively use the RSMT-aware expanded source 3D maze routing (RSMT-aware ESMR) proposed in Subsection IV-B to reroute the remaining congested nets until the termination condition is satisfied, while reducing overflow and avoiding excessive wire length generated.

B. Modified Via-Aware Routing Cost Function

Several variations of cost functions have been discussed in previous works. NTHU-Route 2.0 [4] proposed a history-based cost function. A probability-based cost scheme was introduced in CUGR [9]. FGR [6] developed a discrete Lagrange multiplier cost framework. Although these cost schemes perform well, their via penalties are based on a constant cost function, or, the cost of via may decrease over time. In order to better minimize the number of vias and overcome their shortcoming, we propose a dynamic via-aware cost function.

Rectangular Uniform wire Density (RUDY) [14] is commonly used in routability prediction, in which the wire density of a net is defined as the wire length per unit area of its bounding-box. Instead of RUDY using the half-perimeter wirelength, we adopt the length of the RSMT for an estimation of the wire density:

$$wd_n(u,v) = \begin{cases} \frac{rl}{area}, & (u,v) \in bdb(n); \\ 0, & (u,v) \notin bdb(n), \end{cases}$$

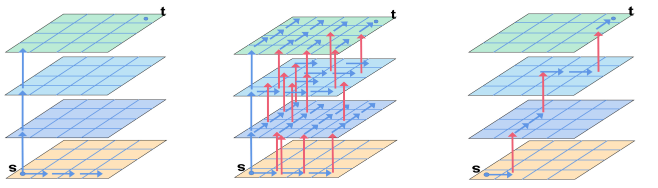
where $wd_n(u,v)$ denotes the wire density of net n on edge $e(u,v)$, $bdb(n)$ is the bounding-box of net n, and rl is the RSMT length of the net n. Additionally, $area$ is the area of the bounding-box of the net.

The wire density of an edge is defined as the sum of the wire densities of all nets N along that edge: $wd_N(u,v) = \sum_{n\in N} wd_n(u,v)$. An edge with higher wire density consumes more routing capacity. When the routing capacity of an edge on the current layer is exhausted, the tendency is to use the corresponding edges of other layers, which leads to an increase in the number of vias.

Increasing the via cost will reduce the use of vias, but will also increase the overflow. Therefore, we balance the increase in overflow with the decrease in the number of vias, while taking into account the impact of wire density on vias. Based on the CUGR cost scheme, we modify it and define the via cost given as:

$$cost_v(u,u') = \frac{\gamma \times uvc \times (1 + \lg(u) + \lg(u'))}{(1.0 + \exp(wd_N(u) + wd_N(u'))^{-1}},$$

where γ is a given parameter, $wd_N(u)$ represents the half of the sum of wire densities of two edges connecting u. In this cost function, the cost of a via is neither a fixed constant nor a value that decreases over time, but can be dynamically adjusted according to the wire density and the degree of congestion.



(a) (b) (c)

Fig. 3: 3D monotonic routing. (a) The $d(u)$ of grid point u with the same x or y or z-coordinate as point s is determined, and the arrow of each point indicates its predecessor node. (b) Predecessor nodes of all points inside 3D bounding-box are determined. (c) 3D monotonic path is obtained by backtracking.

IV. Multi-Strategy Rip-up & Rerouting Framework

This section introduces our multi-strategy rip-up & rerouting framework, which comprises two distinct stages: local rip-up & rerouting and global rip-up & rerouting. Each stage employs different routing algorithms to effectively handle the overflowed nets within their respective regions meanwhile optimizing wirelength and the number of vias.

A. Local Rip-up & Rerouting

After the initial routing, there are still many overflowed nets to be dealt with. We propose a 3D monotonic routing extended from 2D monotonic routing [15], aiming to reduce overflow and minimizing the number of vias without compromising on wire length.

3D monotonic routing works in the bounding-box of a net and requires that, during its routing process, the next path node to be found must satisfy a decrease or maintenance of the 3D Manhattan distance. As a result, for a two-pin net, this algorithm exhibits the characteristic that each grid point can only have one to three predecessor nodes. Algorithm 1 is the pseudo-code for the 3D monotonic routing algorithm. The source and sink points have coordinates $(x1, y1, z1)$ and $(x2, y2, z2)$, respectively. $cost(u,v)$ represents the routing cost of grid edge or via, while $d(u)$ represents the minimum cost from the source to the point u inside the 3D bounding-box. $prev(u)$ refers to the predecessor node of u.

Algorithm 1 3D Monotonic Routing

```
Input: cost array $d$, source $s(x1, y1, z1)$, sink $t(x2, y2, z2)$.
Output: resulting path.
 1: $d(s) = 0$, $prev(s) = null$;
 2: for $x = x1 + 1$ to $x2$ do
 3:     $left = (x-1, y1, z1)$, $cur = (x, y1, z1)$;
 4:     $d(cur) = d(left) + cost(left, cur), prev(cur) = left$;
 5: end for
 6: for $y = y1 + 1$ to $y2$ do
 7:     $after = (x1, y-1, z1)$, $cur = (x1, y, z1)$;
 8:     $d(cur) = d(after) + cost(after, cur), prev(cur) = after$;
 9: end for
10: for $z = z1 + 1$ to $z2$ do
11:     $down = (x1, y1, z-1)$, $cur = (x1, y1, z)$;
12:     $d(cur) = d(down) + cost(down, cur), prev(cur) = down$;
13: end for
14: for $z = z1$ to $z2$ do
15:     for $x = x1$ to $x2$ do
16:         for $y = y1$ to $y2$ do
17:             $left = (x-1, y, z)$, $cur = (x, y, z)$;
18:             $after = (x, y-1, z)$, $down = (x, y, z-1)$;
19:             $d(cur) = \min(d(left) + cost(left, cur),$
                  $d(after) + cost(after, cur), d(down) + cost(down, cur))$;
20:             $prev(cur) = min\_location(left, after, down)$;
21:         end for
22:     end for
23: end for
24: trace back:path.
```

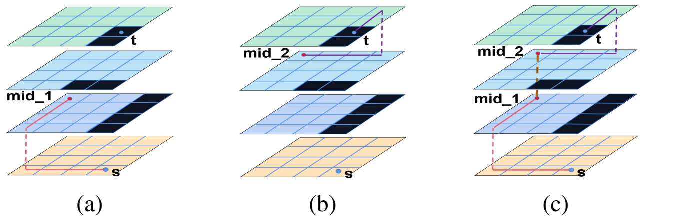


Fig. 4: 3D 3-via-stack routing. (a) Connect s to point mid_1 using L-shape pattern routing. (b) Connect t to point mid_2 using L-shape pattern routing. (c) Connect mid_1 and mid_2 using a via-stack.

In Algorithm 1, lines 1 to 13 calculate the minimum cost for the points with the same x or y or z-coordinate as point s (Fig. 3(a)). In lines 14 to 23, a dynamic programming algorithm is used to compute the minimum cost of each point in sequence at each layer (Fig. 3(b)). In line 24, the algorithm searches for the predecessor node from the sink until it encounters the source, and finally determines the minimum cost 3D monotonic path (Fig. 3(c)). The time complexity of the 3D monotonic routing algorithm is $O(|B|)$, where $|B|$ represents the volume of the 3D bounding-box.

B. Global Rip-Up & Rerouting

Since the local rip-up & rerouting is executed inside a 3D bounding-box, there exist nets still with overflow. To address this issue, we perform two techniques globally, namely 3D 3-via-stack routing and RSMT-aware ESMR. They both reduce overflow, but 3D 3-via-stack routing focuses on adding as few vias as possible, while RSMT-aware ESMR increases wire length as less as possible.

1) 3D 3-Via-Stack Routing: The 3D 3-via-stack routing completes the connection of two pins. It has up to three stack of vias, and has stronger congestion reduction capability than 3D pattern routing. Moreover, its advantage over 3D monotonic routing and 3D maze routing is that it may produce fewer vias and is faster.

Fig. 4 provides an illustration of the 3D 3-via-stack routing. In the figure, the black area represent congested edges. Obviously, 3D pattern routing and 3D monotonic routing cannot get a routing path without overflow within the congested 3D bounding-box, while the 3D 3-via-stack routing obtains a routing path without overflow, and produces as few vias as possible.

Algorithm 2 3D 3-Via-Stack Routing

```
Input: $s(x1, y1, z1)$, $t(x2, y2, z2)$, 3D expanded bounding-box $B$.
Output: best 3D 3-via-stack path.
1: $Cost\_best = +\infty$;
2: for $x = B.minx$ to $B.maxx$ do
3:     for $y = B.miny$ to $B.maxy$ do
4:         for $z1 = B.minz$ to $B.maxz$ do
5:             $mid\_1 = (x, y, z1)$;
6:             for $z2 = B.minz$ to $B.maxz$ do
7:                 $mid\_2 = (x, y, z2)$;
8:                 $cost\_sum = cost\_L\_path(mid\_1, s)$
                   $+cost\_L\_path(mid\_2, t)$
                   $+cost\_via(mid\_1, mid\_2)$;
9:                 if $cost\_sum < Cost\_best$ then
10:                    update best 3D 3-via-stack path;
11:                end if
12:            end for
13:        end for
14:    end for
15: end for
```

The pseudo-code of the 3D 3-via-stack routing is given in Algorithm 2. $B.minx$ and $B.maxx$ represent the min-max value of the x-coordinate of the 3D expanded bounding-box B, respectively. The same meaning applies to $B.miny$ and $B.maxy$; $B.minz$ and $B.maxz$. $cost\_L\_path(p1, p2)$ and $cost\_via(p1, p2)$ denote the minimum cost

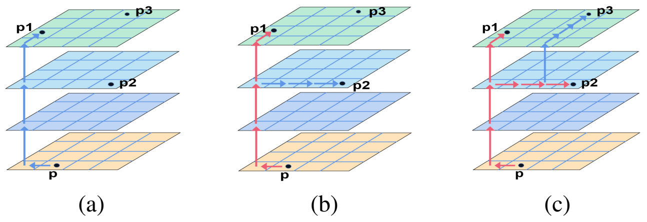


Fig. 5: RSMT-aware ESMR. (a) Connect pin p1. (b) Connect pin p2. (3) Connect pin p3.

and via cost of connecting $p1$ and $p2$ using 3D L-shape pattern routing, respectively. A 3D 3-via-stack path consists of three parts: two 3D L-shape paths and a stack of vias, like using $s \rightarrow mid\_1$, $mid\_2 \rightarrow t$, and $mid\_1 \rightarrow mid\_2$. The 3D 3-via-stack routing is faster than 3D maze routing and offers good congestion reduction. We use it before 3D maze routing to reduce the number of overflowed nets, resulting in lower total overflow and via counts.

2) RSMT-Aware ESMR: After completing the 3D 3-via-stack routing algorithm, only a small number of nets have congestion, and we need to use 3D maze routing to process these nets. Traditional 3D maze routing usually uses 3-terminal maze routing, but it does not perform well in terms of wire length. Hence, we propose ESMR, which integrates RSMT awareness to achieve RSMT-aware ESMR, aiming to avoid the generation of excessively long wire length.

ESMR: The main idea of ESMR is that, the finished routing tree edge in a net can be reused. Our set of source and sink changes dynamically and is not restricted to any two points. As shown in Fig. 5, red line represents the set of sources, while blue line represents a routing path to a pin that is routed in the current iteration. Suppose we need to complete the routing of a net N. First, a pin is selected as the source, and the remaining pins as the sinks. Then, ESMR is utilized to find the minimum cost path to connect the source to any sink (Fig. 5(a)). In this case, points on the path formed by already connected pins are regarded as sources, and the remaining unconnected pins are still regarded as sinks. This process is repeated (Figs. 5(b)(c)) until all pins are connected.

ESMR is presented in Algorithm 3. Each node contains information about its predecessor node and the cost from the source to the current node. $visit$ indicates the node status. Q represents a priority queue sorted by cost, and $need$ is the remaining pins that need to be connected. The function $expand\_source$ adds nodes on the connection path to the source set. Lines 2 to 15 take a node from Q. The adjacent directions of node p are traversed sequentially. If the adjacent node p' is a pin, which means that the shortest path is found, then p' is marked as "visited", and $need$ will be reduced by one. Further, add each node on the shortest path to the source set. If p' is not a pin, then update the cost of p', and add p' to Q. When the $need$ is equal to 0, the routing of this net is finished. In line 16, the final routing path of this net can be obtained by backtracking the predecessor nodes.

RSMT-Aware ESMR: The RSMT-aware routing scheme proposed in NCTU-GR 2.0 [4] enables the router to find a routing solution with a shorter wire length. However, it only used the information of flat segments in RSMT and did not involve non-flat segments. To reduce the wire length of ESMR, we utilize information from the entire RSMT and propose the RSMT-aware ESMR.

First, we get an RSMT by FLUTE [13] for a multi-pin net N. Then, we mark each G-cell traversed by the RSMT and also mark each G-cell in every layer with the same x, y coordinates as the endpoints of the RSMT. In the 3D grid graph, each edge connects two adjacent G-cells, and we modify the cost of edge based on the number of times the connected G-cells have been marked. The cost of edge is defined as:

$$cost(e) = \begin{cases} wl(e) * \frac{C}{2} + eo(e) \times \lg(e), & \text{e is marked twice;} \\ wl(e) * C + eo(e) \times \lg(e), & \text{e is marked once,} \end{cases}$$

Algorithm 3 Expanded Source 3D Maze Routing

```
Input: net, cost array $d$.
Output: resulting path.
 1: $Q.push(s)$, $visit = \{s\}$, $need = n - 1$;
 2: while $Q! = empty$ do
 3:     $p = Q.pop()$;
 4:     for $p'$ in all directions of $p$ do
 5:         if $p'$ is a pin and is not in $visit$ then
 6:             $need - -$, $visit = visit \cup \{p'\}$;
 7:             $expand\_source(p')$;
 8:         else
 9:             $Q.push(p')$;
10:         end if
11:     end for
12:     if $need == 0$ then
13:         $break$;
14:     end if
15: end while
16: $traceback : path$.
```

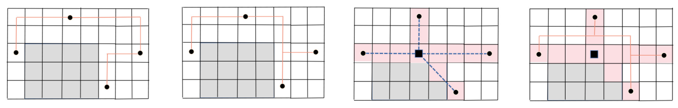


Fig. 6: Comparison of different maze algorithms. (a) Traditional maze routing. (b) ESMR. (c) RMST for Net N. (d) RSMT-aware ESMR.

where C is an adjustable parameter. In the process of RSMT-aware ESMR, the path selection tends to select the edges with marked G-cells rather than other edges.

Fig. 6 compares the routing effects of the traditional maze routing, ESMR, and RSMT-aware ESMR. In the figure, black dots represent pins, gray shaded areas indicate obstacles or congestion, and the rest are routable. The paths generated by each algorithm are shown in orange lines. Fig. 6(a) demonstrates that the traditional maze routing generates a routing with the total wire length of 15, as it lacks the ability to reuse the routing result that has already been completed. In contrast, ESMR is capable of reusing the routing result, resulting in a shorter path and a reduced total wire length of 13 (Fig. 6(b)). Fig. 6(c) depicts the RSMT for this net, where the black cell represents the Steiner point and the pink areas represent marked G-cells. Obviously, RSMT-aware ESMR further reduces the total wire length to 12 (Fig. 6(d)).

V. Experimental Results

The proposed global router V-GR is implemented in C/C++ language, and tested on the benchmarks of ICCAD19 contest [16] and evaluated using the methodology from [16]. All experiments are conducted on a 64-bit Linux workstation with an Intel Core 2.4 GHz CPU and 32 GB memory. Moreover, the obtained global routing solutions are used as routing guide for Dr. CU 2.0 [17] or TritonRoute [18] to generate detailed routing solutions, and subsequently Cadence Innovus [19] is used to evaluate detailed routing solutions and report scores.

A. *Effectiveness of 3D Monotonic Routing and 3D 3-Via-Stack Routing*

In this subsection, in order to evaluate the effectiveness of the 3D monotonic routing and 3D 3-via-stack routing algorithms, we conduct two sets of experiments: one using 3D monotonic routing, 3D 3-via-stack routing and the RSMT-aware ESMR, and another using only the RSMT-aware ESMR. The comparison results are presented in TABLE I. The results show that the router with 3D monotonic routing and 3D 3-via-stack routing achieves effectiveness in vias and overflow reduction. On average, 6.5% of vias and 7.1% of overflow are reduced, while the wire length is almost in the same level.

TABLE I: Comparison of the Results of V-GR with and without 3D Monotonic Routing and 3D 3-Via-Stack Routing Algorithms.

Benchmarks	Ours without the Algorithms			Ours with the Algorithms		
	Wire Length	#Via	Overflow	Wire Length	#Via	Overflow
18test5	**27019100**	816721	0	27029600	**761579**	0
18test5m	**27329100**	804629	**3216**	27343000	**759084**	3221
18test8	**64225300**	2062620	0	64235200	**1895530**	0
18test8m	**63155300**	1970615	5593	63195000	**1855570**	**5500**
18test10	**66635400**	2194560	0	66720800	**2019910**	0
18test10m	**69689400**	2117990	1495	69695800	**1979430**	**645**
19test7	**118296000**	3218160	0	118377000	**3041740**	0
19test7m	**106591000**	3234286	4318	106508000	3034040	**3989**
19test8	181935000	5594214	0	**181836000**	**5213620**	0
19test8m	**179116000**	5553435	6986	179229000	**5263920**	**6551**
19test9	273311000	9126440	60	**273237000**	**8617980**	**0**
19test9m	**271199000**	9438193	**3689**	271278000	**8853840**	3782
Avg.	**120708466**	3844321	2113	120723700	**3608020**	**1974**
Norm.	**99.9%**	106.5%	107.1%	100%	**100%**	**100%**

TABLE II: Comparison of the Results of RSMT-Aware ESMR and Traditional Maze Routing.

Benchmarks	Traditional MR		RSMT-Aware ESMR	
	Wire Length	#Via	Wire Length	#Via
18test5	27326925	766148	**27029600**	**761579**
18test5m	27589087	762879	**27343000**	**759084**
18test8	64819740	1905955	**64235200**	**1895530**
18test8m	63763755	1864840	**63195000**	**1855570**
18test10	67287926	2032231	**66720800**	**2019910**
18test10m	70323062	1989327	**69695800**	**1979430**
19test7	119442393	3056948	**118377000**	**3041740**
19test7m	107466572	3049210	**106508000**	**3034040**
19test8	183472524	5239688	**181836000**	**5213620**
19test8m	180842061	5290239	**179229000**	**5263920**
19test9	275696133	8661069	**273237000**	**8617980**
19test9m	273719502	8898109	**271278000**	**8853840**
Avg.	121812473	3626386	**120723700**	**3608020**
Norm.	100.9%	100.5%	**100%**	**100%**

B. *Effectiveness of RSMT-Aware ESMR*

TABLE II is to demonstrate the effectiveness of the RSMT-aware ESMR in reducing the wire length, where the second and third columns list the wire length and the number of vias of traditional maze routing, respectively. While, the fourth and fifth columns present the corresponding values by RSMT-aware ESMR. For these benchmarks, both routing algorithms perform similarly in terms of overflow reduction. However, RSMT-aware ESMR outperforms traditional maze routing on reducing wire length by 0.9% and vias by 0.5%. This is because RSMT-aware ESMR is able to consciously route paths with shorter wire length on the basis of RSMT.

C. *Comparison with the State-of-the-Art*

We compare the performance of V-GR with CUGR in TABLE III. We perform the two global routers on the same machine, and all benchmarks are run in an eight-threaded environment. It is essential to note that, the 'DR Score' in TABLE III reflects the result reported by Innovus after utilizing TritonRoute due to its superior quality compared to Dr. CU 2.0. The global routing results show that, our algorithm achieves an average reduction of approximately 8.0% on the number of vias and 7.5% on overflow, while maintaining the same level of wire length. Additionally, the 'DR Score' is improved by 1.2%.

TABLE IV lists the detailed results of all components of the 'DR Score'. To demonstrate the superiority of TritonRoute, we compare the data reported by Innovus after using CUGR with Dr. CU 2.0 and CUGR with TritonRoute. The results reveal that CUGR with TritonRoute yields better wire length, #via, and DRVs, compared to CUGR with Dr. CU 2.0, however with higher occurrence of non-preferred routing usage. Since we aim for solutions with minimized wire lengths, #via, and DRVs, we select TritonRoute as our detailed router. The results indicate a reduction of 4.7% in the number of vias and a reduction of 8.7% in the number of design rule violations, while other evaluation indicators remain almost the same. This demonstrates that V-GR can find a routing scheme with fewer vias and less overflow while maintaining almost the same wire length.

TABLE III: Routing Performance Improvement over CUGR.

Benchmarks	GR Wire Length		GR #Via		GR Overflow		GR CPU(s)		DR Score	
	CUGR	V-GR	CUGR	V-GR	CUGR	V-GR	CUGR	V-GR	CUGR	V-GR
18test5	**26997000**	27029600	855742	**761579**	0	0	33.6	**25.8**	15609059	**15557607**
18test5m	27915200	**27343000**	802643	**759084**	**3201**	3221	32.0	**29.0**	15902609	**15580356**
18test8	64380000	**64235200**	2174530	**1895530**	0	0	105.5	**95.4**	37831617	**37076170**
18test8m	64697100	**63195000**	1955940	**1855570**	5587	**5500**	122.0	**91.0**	36973756	**36268485**
18test10	66778500	**66720800**	2308290	**2019910**	0	0	123.2	**90.6**	39640768	**39218365**
18test10m	72840100	**69695800**	2200010	**1979430**	1575	**645**	218.0	**109.0**	41514624	**41342376**
19test7	118701000	**118377000**	3124640	**3041740**	0	0	243.2	**236.4**	78239626	**77623552**
19test7m	106977000	**106508000**	3070990	**3034040**	4332	**3989**	221.0	**195.0**	72627528	**71846934**
19test8	181912000	**181836000**	5748110	**5213620**	0	0	208.4	**198.7**	120660152	**119610521**
19test8m	**177674000**	179229000	5599140	**5263920**	7061	**6551**	**388.0**	405.0	118730552	**116784975**
19test9	274113000	**273237000**	9598230	**8617980**	75	**0**	335.5	**284.8**	185445043	**184104089**
19test9m	**267622000**	271278000	9342640	**8853840**	**3652**	3782	455.0	**418.0**	183776050	**180753328**
Avg.	120883908	**120723700**	3898408	**3608020**	2123	**1974**	207.1	**181.5**	78912615	**77980563**
Norm.	100.13%	**100%**	108.05%	**100%**	107.58%	**100%**	114.08%	**100%**	101.20%	**100%**

TABLE IV: Decomposed Detailed Routing Score Improvement over CUGR.

Benchmarks	Wire Length			#Via			Non-preferred Usage			Design Rule Violations		
	CUGR + Dr. CU 2.0	CUGR + TritonRoute	V-GR + TritonRoute	CUGR + Dr. CU 2.0	CUGR + TritonRoute	V-GR + TritonRoute	CUGR + Dr. CU 2.0	CUGR + TritonRoute	V-GR + TritonRoute	CUGR + Dr. CU 2.0	CUGR + TritonRoute	V-GR + TritonRoute
18test5	13755664	13701011	**13717353**	1854248	1687442	**1614934**	**159196**	306749	225320	335018	**0**	**0**
18test5m	14001665	13891767	**13702732**	1843696	1699774	**1643168**	**120374**	311067	234455	280744	**0**	**0**
18test8	32980410	32778214	**32526979**	4827046	4399240	**4040724**	**254658**	654162	508467	140621	**0**	**0**
18test8m	32407308	32099091	**31651745**	4522450	4112254	**4033852**	**326961**	762411	582889	238783	**0**	**0**
18test10	34054967	33887670	**33874618**	4991064	4617196	**4345714**	**885532**	1135902	998033	662037	**0**	**0**
18test10m	35711640	35370581	**35303348**	4899802	4545166	**4516894**	**1244888**	1541692	1468608	1634185	57183	**53525**
19test7	61002413	60490812	**60272448**	16276308	15155620	**14488772**	**1423304**	2593194	2839479	9779552	**0**	**0**
19test7m	54858539	54237117	**54147005**	16386864	15275220	**14518402**	**1604677**	3115191	3181526	9933662	**0**	**0**
19test8	93471561	**92736584**	92963254	26290792	25323176	**23799936**	**1290976**	2600391	2847331	7466225	**0**	**0**
19test8m	91154538	90262544	**89888979**	25999844	25141952	**23858036**	**899052**	2983055	3037959	8345362	**0**	**0**
19test9	141254192	140047181	**139666849**	42973076	41053404	**39620000**	**2181813**	4169857	4817240	15091717	**0**	**0**
19test9m	138028089	136440857	**135856004**	43195156	41605100	**39795704**	**2474472**	5729092	5101619	15487862	1000	**0**
Avg.	61890082	61328619	**61130942**	16171695	15384628	**14689678**	**1072158**	2158563	2153577	2894823	2423.5	**2229.5**
Norm.	101.24%	100.32%	**100%**	110.09%	104.73%	**100%**	**49.79%**	100.23%	100%	129841.80%	108.70%	**100%**

VI. Conclusion

In this paper, we have presented a 3D global router V-GR that focuses on reducing the number of vias and overflow, while optimizing wire length. A modified via-aware routing cost that takes into account the effect of wire density on the via was designed to optimize the number of vias. Additionally, a novel multi-strategy rip-up & rerouting framework was developed for V-GR to solve the congested net. This framework uses our proposed 3D monotonic routing algorithm locally to reduce the number of vias and overflow, and then uses proprietary 3D 3-via-stack routing and RSMT-aware ESMR globally to rip up & reroute congested nets while ensuring the lowest possible growth in wire length and the number of vias. Evaluation on the ICCAD19 contest benchmarks demonstrates that, V-GR achieves fewer vias and design rule violations than CUGR while maintaining almost the same level of wire length.

References

[1] T.-H. Lee and T.-C. Wang, "Congestion-constrained layer assignment for via minimization in global routing," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 9, pp. 1643–1656, 2008.

[2] Y. Xu, Y. Zhang, and C. Chu, "FastRoute 4.0: Global router with efficient via minimization," in Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference, 2009, pp. 576–581.

[3] Y.-J. Chang, Y.-T. Lee, and T.-C. Wang, "NTHU-Route 2.0: A fast and stable global router," in Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2008, pp. 338–343.

[4] W.-H. Liu, W.-C. Kao, Y.-L. Li, and K.-Y. Chao, "NCTU-GR 2.0: Multithreaded collision-aware global routing with bounded-length maze routing," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 5, pp. 709–722, 2013.

[5] J. He, U. Agarwal, Y. Yang, R. Manohar, and K. Pingali, "SPRoute 2.0: A detailed-routability-driven deterministic parallel global router with soft capacity," in Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference, 2022, pp. 586–591.

[6] J. A. Roy and I. L. Markov, "High-performance routing at the nanometer scale", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 6, pp. 1066-1077, 2008.

[7] T.-H. Wu, A. Davoodi, and J. T. Linderoth, "GRIP: Scalable 3D global routing using integer programming," in Proceedings of ACM/IEEE Design Automation Conference, 2009, pp. 320–325.

[8] Y. Xu and C. Chu, "MGR: Multi-level global router," in Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2011, pp. 250–255.

[9] J. Liu, C.-W. Pui, F. Wang, and E. F. Young, "CUGR: Detailed-routability-driven 3D global routing with probabilistic resource model," in Proceedings of ACM/IEEE Design Automation Conference, 2020, pp. 1–6.

[10] S. Lin, J. Liu, and M. D. Wong, "GAMER: GPU accelerated maze routing," in Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2021, pp. 1-8.

[11] S. Liu, P. Liao, R. Zhang, Z. Chen, W. Lv, Y. Lin, and B. Yu, "FastGR: Global routing on CPU-GPU with heterogeneous task graph scheduler," in Proceedings of IEEE/ACM Design Automation and Test in Europe, 2022, pp. 760–765.

[12] Shiju Lin and Martin D. F. Wong. "Superfast full-scale CPU-accelerated global routing," In Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2022, pp, 1–8.

[13] C. Chu and Y.-C. Wong, "FLUTE: Fast lookup table based rectilinear Steiner minimal tree algorithm for VLSI design," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 1, pp. 70–83, 2007.

[14] P. Spindler and F. M. Johannes, "Fast and accurate routing demand estimation for efficient routability-driven placement," in Proceedings of IEEE/ACM Design Automation and Test in Europe, 2007, pp. 1226-1231.

[15] M. Pan and C. Chu, "FastRoute 2.0: A high-quality and efficient global router," in Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference, 2007, pp. 250–255.

[16] S. Dolgov, A. Volkov, L. Wang, and B. Xu, "2019 CAD Contest: Lef/def based global routing," in Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2019, pp. 1–4.

[17] H. Li, G. Chen, B. Jiang, J. Chen, and E. F. Young, "Dr. Cu 2.0: A scalable detailed routing framework with correct-by-construction design rule satisfaction," in Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2019, pp. 1–7.

[18] A. B. Kahng, L. Wang and B. Xu, "TritonRoute: The open-source detailed router", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 40, no. 3, pp. 547-559. 2021.

[19] "Cadence innovus implementation system." https://www.cadence.com.