

UCLA

UCLA Previously Published Works

Title

NR-Router+: Enhanced Non-Regular Electrode Routing With Optimal Pin Selection for Electrowetting-on-Dielectric Chips

Permalink

<https://escholarship.org/uc/item/35w7326s>

Journal

IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, PP(99)

ISSN

0278-0070

Authors

Pan, Youlin

Liu, Genggeng

Huang, Xing

et al.

Publication Date

2024

DOI

10.1109/tcad.2024.3381070

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

NR-Router+: Enhanced Non-Regular Electrode Routing with Optimal Pin Selection for Electrowetting-on-Dielectric Chips

Youlin Pan, Genggeng Liu, Xing Huang, Zipeng Li, Hsin-Chuan Huang, Chi-Chun Liang, Qining Wang, Chang-Jin Kim, and Tsung-Yi Ho

Abstract—With the advances in microfluidics, electrowetting-on-dielectric (EWOD) chips have widely been applied to various biological and chemical laboratory protocols. Glass-based EWOD chips with non-regular electrodes are proposed, which allow more reliable droplet operations and facilitate the integration of optical sensors for many biochemical applications. Furthermore, non-regular electrode designs are utilized in EWOD chips, e.g., interdigitated electrodes for more reliable droplet manipulation, custom shaped electrodes for specific applications like concentric heating, etc. However, due to the technical challenges of fabricating multi-layer interconnection on the glass substrate, e.g., unreliable process and high cost, both control electrodes and wires are fabricated with a single-layer configuration, which poses significant challenges to pin selection for non-regular electrodes. In this paper, we propose a minimum-cost flow-based routing algorithm called NR-Router+ that features efficient and robust routing for single-layer EWOD chips with non-regular electrodes. To the best of our knowledge, this is the first work that overcomes the aforementioned challenges. We construct a minimum-cost flow algorithm to generate optimal routing paths followed by a light-weight model to handle flow capacity. A grid reduction strategy is proposed to reduce the computational overhead. Additionally, a flow collocation algorithm based on integer linear programming is presented to efficiently prevent wire overlapping. Experimental results show that NR-Router+ achieves 100% routability while minimizing wirelength with shorter run time. Moreover, NR-Router+ can generate mask files feasible for manufacturing via adjustments of design parameters, thus demonstrating its robustness and efficiency.

I. INTRODUCTION

The use of Electrowetting-on-dielectric (EWOD) as an actuation mechanism for digital microfluidics has shown great

A preliminary version of this paper was published in the Proceedings of Asia and South Pacific Design Automation Conference (ASP-DAC), 2022 [1].

Y. L. Pan is with the School of Computer Science, Northwestern Polytechnical University, Xi'an, 710072, China, and also with the College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China (email: kzy.pan@gmail.com).

G. G. Liu is with the College of Computer and Data Science, Fuzhou University, Fuzhou 350116, China (e-mail: liugenggeng@fzu.edu.cn).

X. Huang is with the School of Computer Science, Northwestern Polytechnical University, Xi'an, 710072, China (e-mail: xing.huang@nwpu.edu.cn).

Z. Li is with the Silicon Engineering Group, Apple Inc., Cupertino, CA, USA (email: nyfranklee@gmail.com).

H.-C. Huang and C.-C. Liang are with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan (email: free44458@gmail.com; gy5204301@gmail.com).

Q. N. Wang and C.-J. Kim are with Mechanical and Aerospace Engineering Department, University of California, LA, USA (email: wangqining265@gmail.com; cjkim@ucla.edu).

T.-Y. Ho is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: tyho@cse.cuhk.edu.hk).

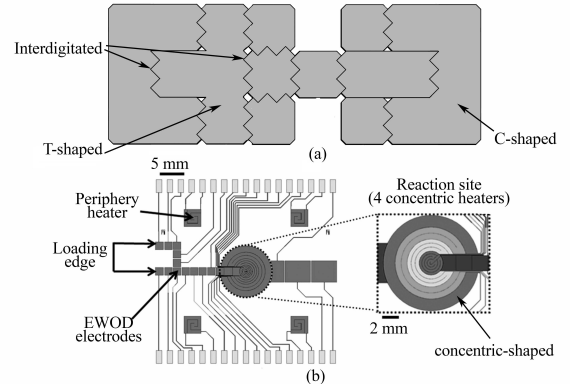


Fig. 1. (a) The C-shaped, T-shaped, and interdigitated electrodes. (b) The synthesis of molecular probes using concentric-shaped electrodes to heat droplets [2].

promise. It has enabled the development of numerous lab-on-chip platforms for biomedical and biochemical applications [3]. Based on the actuation mechanism of EWOD, electrodes are connected with contact pads, through which voltage is applied to electrodes to manipulate droplets. By manipulating individual droplets using electrical signals, EWOD chips automate a variety of microfluidics protocols and drastically reduce sample consumption and experiment time [4].

With the advancement of EWOD, various substrate materials (e.g., glass, paper, PCB, etc.) and electrode configurations are developed to meet miscellaneous requirements in diverse applications [5]. Glass-based EWOD chips with non-regular electrodes (i.e., any shape other than square) are one of the widely adopted EWOD chip configurations [6]–[8]. With smooth surface topography, high-resolution electrode, and transparent substrate, glass-based EWOD chips provide reliable droplet operation and facilitate integration for optical modules (e.g., sensors, cameras, etc.), thus significantly extending its application scenarios and assisting in its correction of error [9]–[11]. On the other hand, non-regular electrodes are extensively implemented in various EWOD chips. For example, C-shaped and T-shaped electrodes are often utilized to dispense droplets from a reservoir as shown in Fig. 1(a) [12]. Stripped electrodes together with sector electrodes and arrow-shaped electrodes are proposed to form “L-junction and “Y-junction on EWOD chips to facilitate fast and efficient droplet dispensing and splitting [13]. Furthermore, interdigitated or jagged electrodes are employed to enhance precise control of droplet volume and prevent the pinning effect in droplet motion [6], [14], [15], as shown in Fig. 1(a). It is also noteworthy that electrodes with specific non-regular shapes are necessary to achieve certain functions. For example, a

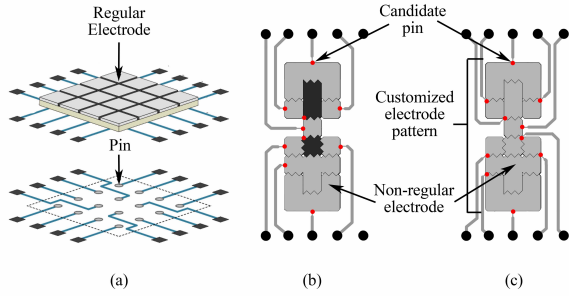


Fig. 2. (a) The pins are placed at the center of regular electrodes [22]. (b) Routing failure with unreasonable candidate pin selection. (c) Routing result using NR-Router+ with the optimal candidate pin selection.

concentric-shaped multifunctional electrode is designed for heating and temperature sensing in the on-chip synthesis of molecular probes [2], as shown in Fig. 1(b).

However, lack of reliable and economical methods for fabricating multi-layer interconnection on a glass substrate, most glass-based EWOD chips use a single-layer configuration with both regular and non-regular electrodes [16]. Hence, a large number of electrodes result in significant challenges for wire routing on a limited area in EWOD chip design. Moreover, because of the difficulty of pin selection, the incorporation of non-regular electrodes further complicates the routing problem for single-layer EWOD chips. Consequently, despite the aforementioned immense utilities of single-layer EWOD chips with non-regular electrodes, there is no existing solution to perform automatic wire routing. In traditional chip design, designers draw wires to connect electrodes with contact pads manually. As chip integration grows, designers suffer from the increasingly burdensome task of routing, and this barrier limits the development of the EWOD community [17]–[21]. Thus, there is a pressing need for an automatic routing tool for single-layer EWOD chips with non-regular electrodes.

Thus far, most existing tools are designed to handle routing problems for PCB-based EWOD chips [22]–[24]. With the multi-layer configuration, wires are allowed to go through holes on the PCB without considering the obstacles of electrodes which is the primary concern to route single-layer EWOD chips. To solve a similar routing problem for paper-based EWOD chips, Wang et al. [25] proposed the minimum cost flow algorithm based on routing grids. However, these tools cannot solve the routing problem for single-layer EWOD chips with non-regular electrodes. The pin selection problem is out of consideration in these tools as the pins are located in the center of electrodes, as shown in Fig. 2(a). For each non-regular electrode, a candidate pin needs to be chosen to route a wire. Unlike regular electrodes, the shape of the non-regular electrode determines the number of pins on it. Thus, the location of candidate pins should be designed for non-regular electrodes, as shown in Fig. 2(b). In addition, to select the optimal candidate pin of each electrode for 100% routability, it is necessary to consider simultaneously all possible routing wires and pins, which makes the problem more complicated. To efficiently solve these problems, we propose the NR-Router+ in this paper, a minimum-cost flow-based routing approach. To the best of our knowledge, NR-Router+ is the

first approach that realizes accurate routing design in single-layer EWOD chips with non-regular electrodes, as shown in Fig. 2(c). The main contributions are summarized as follows:

- We propose a minimum-cost flow-based routing approach, which performs efficient and robust routing design for the single-layer EWOD chips with non-regular electrodes while achieving 100% routability and minimized total wirelength.
- We propose an efficient pin selection method that overcomes the bottleneck of pin selection in non-regular electrodes with limited routing resources.
- We propose a light-weight model that can accurately compute the number of wires passing through the orthogonal and diagonal directions of a tile.
- We propose a grid reduction strategy that compresses multiple neighboring grids on the routing mesh into a set of super grids without reducing the number of potential optimal pseudo nodes on electrodes, thereby significantly reducing the complexity of design tasks. Moreover, the correctness of the proposed super grid model is theoretically proved.
- We propose an integer linear programming (ILP)-based flow collocation algorithm that determines how wires pass through the super grids, tiles, and connection regions between electrode area and contact area in a global manner, thus generating crossing-avoidance routing solutions with minimized total wirelengths.
- We validate the proposed method using six synthetic benchmarks and two biochemical applications. Experimental results confirm the robustness and efficiency of our algorithm. Moreover, mask files feasible for manufacturing can be generated automatically.

The remainder of this paper is organized as follows. Section II describes the problem formulation. Section III provides the details of NR-Router+. Section IV demonstrates experimental results and Section V concludes this paper.

II. PROBLEM FORMULATION

In this section, we first introduce the regular electrodes and non-regular electrodes used in EWOD chips. Then, we discuss the importance of candidate pin selection. Finally, the problem solved in this paper is formulated.

For most general-purpose single-layer EWOD chips, regular electrodes are designed in a square shape to efficiently utilize the routing area [26]. Each edge has a pin through which the electrode and contact pad are connected. In contrast, non-regular electrodes are defined as an electrode with any shape other than square. Unlike regular electrodes, the number of pins on a non-regular electrode is determined by the shape and length of its edges.

The number of pins on non-regular electrodes varies, which complexes the problem of candidate pin selections. Any point on the electrode can be connected to the wire as a pin. There are many pins on an electrode that can be chosen to route a wire. If a pin is chosen, we call it a *Candidate Pin*. The selection of candidate pins may influence not only the routability but also the wirelength. If the pin closest to

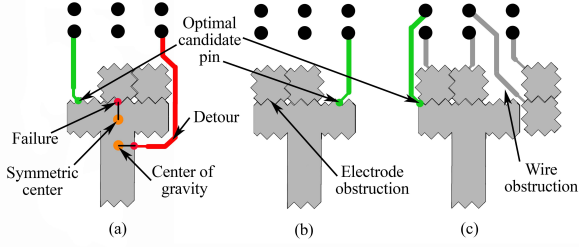


Fig. 3. (a) Detour and failure happen. (b)(c) Electrode and wire obstruction change the optimal candidate pin selection.

the symmetric center or center of gravity of the non-regular electrode is chosen, a detour or failure might happen, as shown in Fig. 3(a). Fig. 3(b) and (c) show the cases that the optimal candidate pins are changed by electrode and wire obstructions. Unfortunately, the wires are generated after candidate pins are determined. Therefore, in order to achieve 100% routability and shorter wirelength, all possible pins and routing wires must be considered simultaneously during candidate pin selection. As a result, the selection of candidate pins becomes much more difficult in non-regular electrodes.

To connect wires between electrodes and contact pads in single-layer EWOD chips with non-regular electrodes, this routing problem can be formulated as follows:

- **Input:** 1) The locations of contact pads and electrodes, 2) the shape of electrodes, and 3) chip specification.
- **Output:** 1) The paths that connect electrodes with any shapes to contact pads and 2) a feasible mask file for manufacturing.
- **Objective:** Maximize the routability while minimizing the total wirelength.

III. DETAILS OF THE PROPOSED METHOD

This section presents our proposed wire routing algorithm, NR-Router+, which is based on minimum cost flow. We begin by providing an overview of the algorithm, followed by a description of the routing area division and the definitions of tile and grid. We then introduce the concept of the pseudo node and discuss the lightweight model and the grid reduction strategy, both of which reduce the computational overhead of our approach. Next, we explain how the corresponding minimum cost flow algorithm is constructed and prove the correctness of the proposed super grid model. Finally, we present the flow collocation algorithm based on ILP, which ensures that the wires do not overlap.

A. Proposed Approach

Although [25], [27] proposed two minimum cost flow algorithms based on routing meshes, these methods cannot be applied to EWOD chips with non-regular electrodes. Thus, we construct a minimum cost flow algorithm based on a light-weight model and a grid reduction strategy, which can generate optimal routing results with appropriate selections of candidate pins. Fig. 4 shows the overall flow of NR-Router+. In the flow network construction stage, we first construct the original routing meshes based on the given design parameters. The constructed routing meshes are then simplified using the proposed grid reduction strategy. Based on the simplified

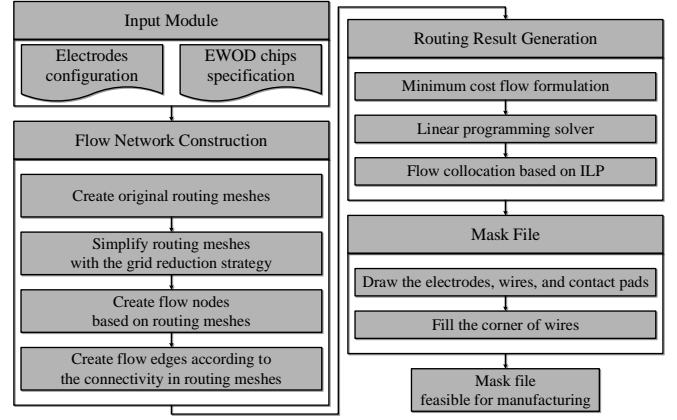


Fig. 4. The overall flow of NR-Router+.

routing meshes, a complete flow network including all flow nodes and flow edges is constructed efficiently. Afterward, a minimum cost flow algorithm is formulated based on the constructed flow network and the routing result is generated using an ILP-based flow collocation algorithm. Finally, we convert the routing result into a mask file composed of all electrodes, contact pads, and routed wires, which is feasible for manufacturing by adjusting design parameters.

B. Tile and Grid

To generate optimal routing results, we divide the routing area into the following two regions as shown in Fig. 5(a):

- **Contact Area** is located at the top and bottom of the chip, which contains all the contact pads.
- **Electrode Area** is located in the middle of the chip, which contains all the electrodes.

In Fig. 5(b), NR-Router+ adopts two routing meshes with different sizes to partition the contact area and the electrode area, respectively. Specifically, the contact area is partitioned into multiple squares of the same size by the contact pad array, where each square is called a *Tile* and each corner of a tile corresponds to a contact pad. Similarly, the electrode area can be partitioned into a mesh by horizontal and vertical lines with uniform spacing. *Grids* are formed by adjacent horizontal and vertical lines, and *Grid Points* are intersections between these horizontal and vertical lines. Note that the width of the grid w_g is a user-defined design parameter that affects the routability and complexity of the algorithm. To ensure sufficient space between wires, especially between parallel 45° wires, w_g should satisfy

$$w_g \geq \frac{w_l + s_l}{\sin 45^\circ}, \quad (1)$$

where w_l and s_l are the wire width and the spacing between wires, respectively.

C. Pseudo Node

In this subsection, we introduce the concept of *Pseudo Node* to select appropriate candidate pins on electrodes. We consider each electrode as a set composed of pseudo nodes. To obtain pseudo nodes on electrodes, the following two strategies are tried: 1) Considering the closest grid points inside as

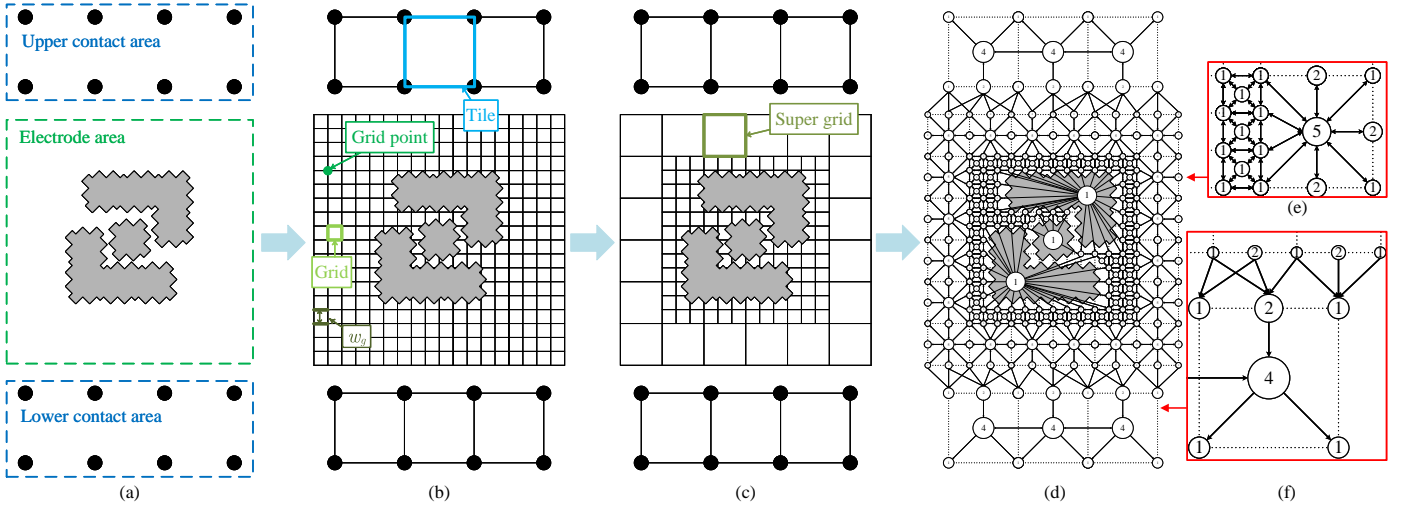


Fig. 5. The procedure of flow network construction. (a) The routing area division of an EWOD chip. (b) The constructed routing meshes. (c) The routing meshes after performing the grid reduction strategy. (d) The constructed flow network. (e) The enlarged connections between a super grid and an original grid. (f) The enlarged connections between an electrode area and a contact area.

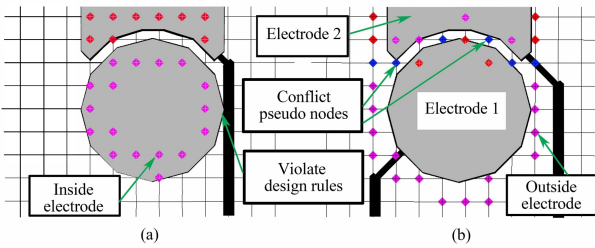


Fig. 6. The purple nodes are pseudo nodes of electrode 1, and the red nodes are pseudo nodes of electrode 2. (a) The gap between wire and electrode violates design rules. (b) The pseudo nodes conflict happens.

electrode as pseudo nodes, which leads to routing results that violate design rules as shown in Fig. 6(a) and 2) considering the closest grid points outside an electrode as pseudo nodes, which leads to conflicts between pseudo nodes as shown in Fig. 6(b). As a result, we compromise on choosing the closest grid point outside each electrode edge as pseudo nodes. If there is a conflict between multiple electrodes' pseudo nodes, and a conflict pseudo node has adjacent grid points outside the electrodes, it will become the pseudo hub and will be excluded from the set of pseudo nodes. The grid points inside the electrodes adjacent to the pseudo hubs are selected as the new pseudo nodes. We add an undirected edge with capacity 1 and unit flow cost 0 to connect pseudo hubs and adjacent new pseudo nodes. Additionally, pseudo nodes record the nearest point on the electrode. For each electrode, we remove the pseudo nodes covered by neighboring electrodes and the ones that are in conflict as shown in Fig. 7(a). Fig. 7(b) shows an example of removing unnecessary pseudo nodes.

D. Light-Weight Model

In previous minimum cost flow algorithms [28], [29], flow capacity is defined as the number of wires that can pass through a tile. Flow capacities are usually categorized into *Orthogonal Capacity* (O-cap) and *Diagonal Capacity* (D-cap). O-cap limits the number of wires through four edges of the tile, while D-cap limits the number of wires through two diagonals of the tile. Unlike the escape routing problem in integrated circuits [27], the contact pads are located on the side of the

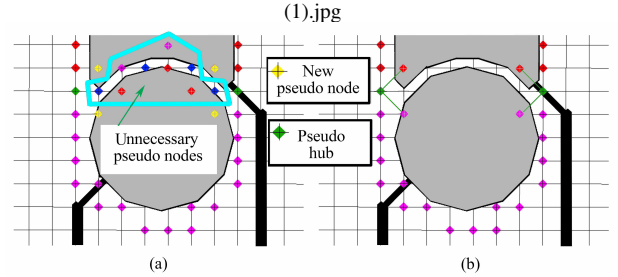


Fig. 7. (a) The unnecessary pseudo nodes. (b) Removing unnecessary pseudo nodes.

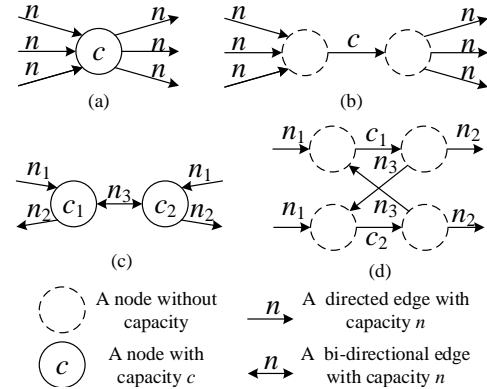


Fig. 8. (a) A node with capacity c . (b) The flow model of (a) with nodes without capacity. (c) The bi-directional edge between two nodes with capacity. (d) The flow model of (c) with nodes without capacity.

EWOD chip. Therefore, in the lower contact area, the wires all start at the upper boundary of the region, while in the upper contact area, the wires all start at the lower boundary of the region. Based on this condition, we propose a light-weight model that can handle the O-cap and D-cap correctly and efficiently. To facilitate method illustration, we introduce the node with capacity. As shown in Fig. 8(a) and 8(b), a node with capacity c can be represented by two nodes without capacity and a directed edge with capacity c . In addition, a bi-directional edge between two nodes with capacity can be realized by the way shown in Fig. 8(c) and 8(d).

The light-weight model considers a tile as a node with

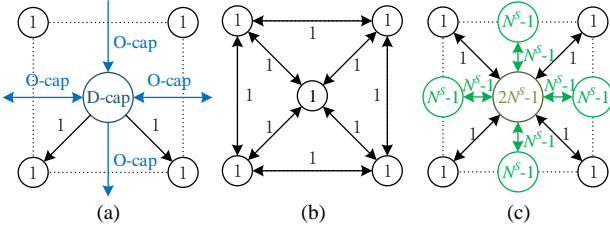


Fig. 9. (a) The light-weight model of a tile located in the lower contact area. (b) The flow model of a grid. (c) The flow model of a super grid.

capacity D-cap called *tile node*. In the lower contact area, for example, a tile node is connected to the horizontally adjacent tile nodes by two bi-directional edges with capacity O-cap and to the vertically adjacent tile node by two directed edges with capacity O-cap. In addition, the tile node is connected to the lower-left and lower-right contact pad nodes by directed edges with capacity 1. Fig. 9(a) shows the light-weight model. The tiles in the upper contact area have a similar configuration. We add a *Boundary Node* to each tile adjacent to the electrode area, with capacity O-cap, to limit the number of wires from the electrode area, as shown in Fig. 5(f), where O-cap = 2 and D-cap = 4. Each boundary node is connected to the corresponding tile node by a directed edge with capacity O-cap and unit flow cost $w_t/2$, where w_t is the width of the tile.

E. Grid Reduction Strategy

We consider a grid as a node with capacity 1 called *grid node* and a grid point as a node with capacity 1 called *grid point node*. Unlike the corners of a tile, which are occupied by contact pads, wires can pass through the corners of a grid. As shown in Fig. 9(b), a grid node is connected to the grid point nodes on the corners by bi-directional edges with capacity 1. To prevent wire overlapping, we do not connect adjacent grid nodes. Instead, adjacent grid point nodes are connected by a bi-directional edge of capacity 1.

As mentioned before, we use pseudo nodes to select the optimal candidate pins on electrodes. Note that the number of pseudo nodes on an electrode is closely related to the size of the grids. Specifically, as the feature size of a single grid reduces, the total number of grids on the routing plane will be increased significantly. As a result, a large number of pseudo nodes can be allocated to electrodes, thereby improving the routability of the chip. On the other hand, however, more nodes will be introduced into the flow network as the increase in the number of grids, leading to extra computational overhead in solving the minimum cost flow problem.

To reduce the computational overhead while improving the solution quality of routing design, we further propose a grid reduction strategy, which compresses multiple neighboring grids on the routing mesh into a set of super grids without reducing the number of potential optimal pseudo nodes on electrodes. Algorithm 1 shows the details of the proposed grid reduction strategy, where the size of the super grid is denoted as N^S ($N^S \geq 2$). We start from the lower-left corner of the electrode area and check whether there are electrodes in the adjacent $N^S \times N^S$ grids one by one. If electrodes exist in the adjacent $N^S \times N^S$ grids, the region is skipped. If there is no

Algorithm 1 Grid Reduction Strategy

Input: All the grids in the electrode area, the number of rows N_{row} and the number of columns N_{col} of the grids, the size of the super grid N^S .

Output: Grids and super grids after reduction.

```

1: for  $j \leftarrow 1$  to  $\lfloor N_{\text{row}}/N^S \rfloor$  do
2:   for  $i \leftarrow 1$  to  $\lfloor N_{\text{col}}/N^S \rfloor$  do
3:     reduce_flag  $\leftarrow 1$ ;
4:      $G_{\text{tmp}} \leftarrow \emptyset$ 
5:     for  $x \leftarrow (i-1) \times N^S + 1$  to  $i \times N^S$  do
6:       for  $y \leftarrow (j-1) \times N^S + 1$  to  $j \times N^S$  do
7:         Add the grid in column  $x$  and row  $y$  to  $G_{\text{tmp}}$ ;
8:         if Electrodes exist inside the grid in column  $x$ 
           and row  $y$  then
9:           reduce_flag  $\leftarrow 0$ ;
10:        if reduce_flag = 1 then
11:          Replace all the grids in  $G_{\text{tmp}}$  with a super grid;
```

electrode in the adjacent $N^S \times N^S$ grids, we replace all the grids and grid points in the region with a super grid, except for the grid points on the four corners of the region and in the common edge of a super grid and a grid. The result of the reduction is shown in Fig. 5(c).

Each super grid contains five flow nodes with capacity, as shown in Fig. 9(c). Four of these nodes, called *edge nodes*, are on the four edges of the super grid and are used to control the flows through each edge of the super grid, while the remaining one node, called the *central node*, is located inside the super grid and is used to control the flows passing through the super grid. The grid points located on the four corners of the super grid are called the *corner nodes* of the super grid. Note that neighboring super grids share edge nodes on common edges. And the grid points located on the common edge of a super grid and a grid are retained, as shown in Fig. 5(e), where $N^S = 3$. The flow nodes corresponding to these grid points are connected to the central node of the super grid with a bi-directional edge with capacity 1 and unit flow cost $w_g \times N^S/2$.

Since the number of wires crossing each edge of the super grid does not exceed $N^S - 1$, we give each edge node an $N^S - 1$ capacity. In addition, since the number of wires through the super grid is at most $2 \times N^S - 1$, we give the central node a capacity of $2 \times N^S - 1$. We use four bi-directional edges with capacity $N^S - 1$ to connect the central node to each edge node, and use the other four bi-directional edges with capacity 1 to connect the central node to each corner node. The incoming flow (outgoing flow) of the central node is called the incoming flow (outgoing flow) of the super grid. The correctness of super grid model is proved in Section III-G.

F. Minimum Cost Flow

Based on the above discussion, we construct a flow network $G(V, E)$ where each edge $(u, v) \in E$ has capacity $C(u, v)$, flow $F(u, v)$ and unit flow cost $\Gamma(u, v)$. The corresponding flow network is constructed as follows:

- A super source node S and a super target node T are added into V , with capacity ∞ and unit flow cost 0.
- For each electrode, an electrode node ψ_i is added into V , with capacity 1 and unit flow cost 0.

Algorithm 2 Connect(\mathcal{A}, \mathcal{B})

Input: The set of starting nodes \mathcal{A} , the set of ending nodes \mathcal{B} .

Output: A set of edges $\mathcal{E}_{\mathcal{A}, \mathcal{B}}$ from nodes in \mathcal{A} to nodes in \mathcal{B} .

```

1:  $\mathcal{E}_{\mathcal{A}, \mathcal{B}} \leftarrow \emptyset$ ;
2: Push the elements in  $\mathcal{A}$  and  $\mathcal{B}$  into the stack  $S_{\mathcal{A}}$  and  $S_{\mathcal{B}}$ ,
   respectively, from right to left according to the position
   of their center points;
3:  $b \leftarrow S_{\mathcal{B}}.\text{pop}()$ ;
4:  $b' \leftarrow \emptyset$ ;
5:  $a \leftarrow S_{\mathcal{A}}.\text{pop}()$ ;
6: while  $S_{\mathcal{A}} \neq \emptyset$  do
7:   if  $b = \emptyset$  then
8:     Add a directed edge into  $\mathcal{E}_{\mathcal{A}, \mathcal{B}}$  from  $a$  to  $b'$ , with
     capacity  $\text{Capacity}(a)$  and unit flow cost  $X(a, b')$ ;
     \ \  $\text{Capacity}(a)$  is the capacity of  $a$ ;  $X(a, b')$  is the
     horizontal distance between  $a$  and  $b'$ ;
9:      $a \leftarrow S_{\mathcal{A}}.\text{pop}()$ ;
10:   else if  $a$  is at the left of  $b$  then
11:     Add a directed edge into  $\mathcal{E}_{\mathcal{A}, \mathcal{B}}$  from  $a$  to  $b$ , with
     capacity  $\text{Capacity}(a)$  and unit flow cost  $X(a, b)$ ;
12:     if  $b' \neq \emptyset$  then
13:       Add a directed edge into  $\mathcal{E}_{\mathcal{A}, \mathcal{B}}$  from  $a$  to  $b'$ , with
       capacity  $\text{Capacity}(a)$  and unit flow cost  $X(a, b')$ ;
14:      $a \leftarrow S_{\mathcal{A}}.\text{pop}()$ ;
15:   else
16:      $b' \leftarrow b$ ;
17:      $b \leftarrow S_{\mathcal{B}}.\text{pop}()$ ;
18: return  $\mathcal{E}_{\mathcal{A}, \mathcal{B}}$ 

```

- For each electrode, a set of pseudo nodes are added into V , with capacity 1 and unit flow cost 0.
- For each contact pad, a contact pad node ϵ_j is added into V , with capacity 1 and unit flow cost 0.
- For each grid point not covered by electrodes, a grid point node is added into V , with capacity 1 and unit flow cost 0.
- For each grid not covered by electrodes, a grid node is added into V , with capacity 1 and unit flow cost 0.
- For each tile, a tile node is added into V , with capacity D-cap and unit flow cost 0.
- For each tile adjacent to the electrode area, a boundary node is added into V , with capacity O-cap and unit flow cost 0.
- For each super grid, four edge nodes are added into V , with capacity $N^S - 1$ and unit flow cost 0, and a central node is added into V , with capacity $2 \times N^S - 1$ and unit flow cost 0.
- A directed edge is added into E for each $(S, \psi_i), (\epsilon_j, T)$, with capacity 1 and unit flow cost 0.
- A directed edge is added into E between electrode nodes and corresponding pseudo nodes, with capacity 1 and unit flow cost 0.
- The directed edges are added into E between pseudo nodes and adjacent grid nodes or grid point nodes, between pseudo hubs and adjacent grid nodes or grid point nodes, with capacity 1 and unit flow cost 0.
- The directed edges are added into E between tile nodes

and adjacent contact pad nodes, with capacity 1 and unit flow cost $\sqrt{2}w_t/2$.

- The directed edges obtain by Connect(A, B) and Connect(A', B') are added into E , where Connect(\mathcal{A}, \mathcal{B}) is used to connect the electrode area to the contact area, as shown in Algorithm 2. A and A' are the sets of edge nodes of super grids and grid point nodes adjacent to the upper contact area and the lower contact area, respectively. B and B' are the sets of contact pad nodes and boundary nodes adjacent to the upper boundary and the lower boundary of the electrode area, respectively.
- A directed edge is added into E between each boundary node and corresponding tile node, with capacity O-cap and unit flow cost $w_t/2$.
- A directed edge is added into E between each pair of vertically adjacent tile nodes, with capacity O-cap and unit flow cost w_t .
- A bi-directional edge is added into E between each pair of horizontally adjacent tile nodes, with capacity O-cap and unit flow cost w_t .
- A bi-directional edge is added into E between each corner node and the central node in the same super grid, with capacity 1 and unit flow cost $w_g \times N^S/2$.
- A bi-directional edge is added into E between each edge node and the central node in the same super grid, with capacity $N^S - 1$ and unit flow cost $w_g \times N^S/2$.
- A bi-directional edge is added into E between each grid node and each adjacent grid point node, with capacity 1 and unit flow cost $w_g/2$.
- For each grid, a bi-directional edge is added into E between each pair of adjacent grid point nodes, with capacity 1 and unit flow cost w_g .

According to the above steps, the flow network shown in Fig. 5(d) is constructed with $N^S = 3$, O-cap = 2, and D-cap = 4. Note that the super source and super target in Fig. 5(d) are hidden, and they are respectively connected to the electrode nodes and contact pad nodes. We can obtain the minimum cost flow by minimizing $\sum_{(u,v) \in E} \Gamma(u,v) \times F(u,v)$ on the constructed flow network.

G. Correctness of the Proposed Super Grid Model

To demonstrate the correctness and effectiveness of the proposed super grid model, we use $f(g_i^s)$ to represent the total number of flows that pass through a super grid $g_i^s \in G^s$, where G^s is the set of super grids on the routing plane. Then, we have the follow lemma:

Lemma 1: In the minimum cost flows corresponding to the flow network constructed in Section III-F, a quantity of $f(g_i^s)$ flows that pass through a super grid can be transformed into $f(g_i^s)$ disjoint paths on the corresponding routing grid before compression, where the size of the routing grid is $N^S \times N^S$ and $f(g_i^s) \leq 2 \times N^S - 1$.

To prove Lemma 1, we first discuss three properties of the minimum cost flows on the super grid.

Property 1: In a super grid, the incoming flow and outgoing flow cannot share an edge node.

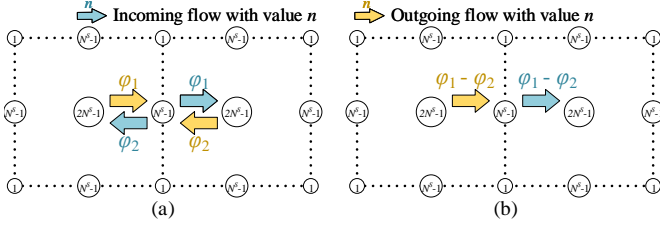


Fig. 10. By offsetting the incoming flow and outgoing flow each other, we can reduce the total cost without affecting the value of total flows. (a) Before the change. (b) After the change. ($\varphi_1 \geq \varphi_2$)

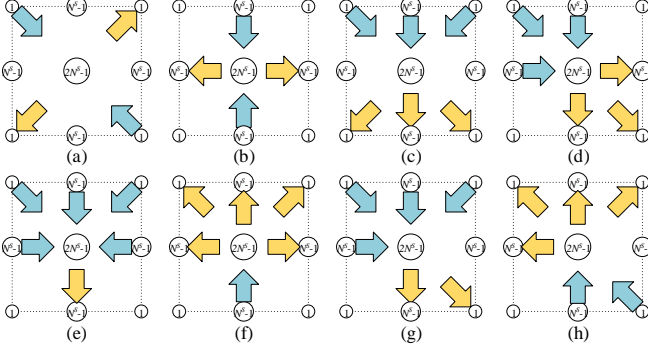


Fig. 12. (a)-(h) The eight flow configurations.

Property 2: In a super grid, if an outgoing flow points to a corner node, then no incoming flow can come from the two edge nodes adjacent to that corner node.

Property 3: In a super grid, if an incoming flow comes from a corner node, then no outgoing flow can point to the two edge nodes adjacent to that corner node.

For Property 1, we assume there are incoming and outgoing flows that share the same edge node in the minimum cost flows through the left super grid, as shown in Fig. 10(a). We assume the values of the outgoing and incoming flow of the left super grid are φ_1 and φ_2 , respectively. Without loss of generality, let $\varphi_1 \geq \varphi_2$. We can reduce the total cost by offsetting the incoming flow and outgoing flow each other, without affecting the other flows, as shown in Fig. 10(b). This contradicts our assumption that the flow through the super grid already has minimum cost. Therefore, Property 1 is true.

For Property 2, we assume that in the minimum cost flows through the left super grid, an outgoing flow points to a corner node, and an incoming flow comes from the edge nodes adjacent to the corner node, as shown in Fig. 11(a). We can reduce the total cost by shifting the outgoing flow pointing to the corner node to the neighboring super grid, without affecting the other flows, as shown in Fig. 11(b). This contradicts our assumption that the flow through the super grid already has minimum cost. Property 2 holds true, and the same reasoning applies to Property 3.

According to Property 1, Property 2 and Property 3, we can enumerate the eight possible configurations of the incoming/outgoing flows through the super grid, as shown in Fig. 12. Notice that the value of flows shown in Fig. 12 can be zero. By means of rotation or symmetry, we can extend these eight configurations to all possible flow configurations. Therefore, our goal is converted into demonstrating that Lemma 1 is true in all eight flow configurations.

For illustration, we use $f_{in}(g_i^s, U)$, $f_{in}(g_i^s, D)$, $f_{in}(g_i^s, L)$, and $f_{in}(g_i^s, R)$ to represent the values of the incoming flows

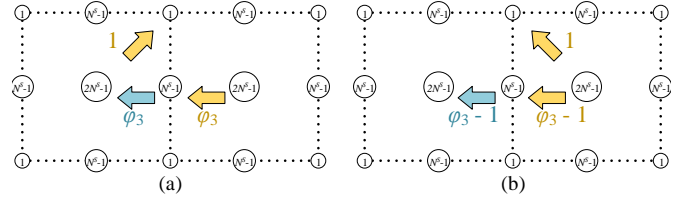


Fig. 11. By shifting the outgoing flow pointing to the corner node to the neighboring super grid, we can reduce the total cost without affecting the value of total flows. (a) Before the change. (b) After the change.

from the upper, lower, left, and right edge nodes of g_i^s , respectively. $f_{out}(g_i^s, U)$, $f_{out}(g_i^s, D)$, $f_{out}(g_i^s, L)$, and $f_{out}(g_i^s, R)$ are the values of the outgoing flows to the upper, lower, left, and right edge nodes of g_i^s , respectively. $f_{in}(g_i^s, UL)$, $f_{in}(g_i^s, UR)$, $f_{in}(g_i^s, DL)$, and $f_{in}(g_i^s, DR)$ are the values of the incoming flows from the upper-left, upper-right, lower-left, and lower-right corner nodes of g_i^s , respectively. $f_{out}(g_i^s, UL)$, $f_{out}(g_i^s, UR)$, $f_{out}(g_i^s, DL)$, and $f_{out}(g_i^s, DR)$ are the values of the outgoing flows to upper-left, upper-right, lower-left, and lower-right corner nodes of g_i^s , respectively.

1) *The First Flow Configuration:* In the first flow configuration shown in Fig. 12(a), the incoming flows come from two opposite corner nodes, and the outgoing flows point to the remaining corner nodes. We have $f(g_i^s) \leq 2$. Obviously, all flows can be transformed into disjoint paths regardless of the value of flows. Thus, Lemma 1 is true in the first flow configuration.

2) *The Second Flow Configuration:* In the second flow configuration shown in Fig. 12(b), the incoming flows come from two opposite edge nodes, and the outgoing flows point to the remaining edge nodes. We have $f(g_i^s) \leq 2 \times N^S - 2$. Without loss of generality, let $f_{in}(g_i^s, U) \geq f_{out}(g_i^s, R)$. We have $f_{out}(g_i^s, L) \geq f_{in}(g_i^s, D)$, since $f_{in}(g_i^s, U) + f_{in}(g_i^s, D) = f_{out}(g_i^s, L) + f_{out}(g_i^s, R)$.

In the routing grid before compression, we choose $f_{in}(g_i^s, D)$ grid points from left to right on the lower boundary as the starting points of the paths and $f_{out}(g_i^s, L)$ grid points from lower to upper on the left boundary as the ending points of the paths. We can construct $f_{in}(g_i^s, D)$ parallel paths from the lower boundary to the left boundary in the area framed with a red line at the lower left of Fig. 13(a). And there are $f_{out}(g_i^s, L) - f_{in}(g_i^s, D)$ ending points remaining on the left boundary.

Similarly, we choose $f_{in}(g_i^s, U)$ grid points from right to left on the upper boundary as the starting point of the paths and $f_{out}(g_i^s, R)$ grid points from upper to lower on the right boundary as the ending point of the paths. We can construct $f_{out}(g_i^s, R)$ parallel paths from the upper boundary to the right boundary in the area framed with a red line on the upper right of Fig. 13(a). And there are $f_{in}(g_i^s, U) - f_{out}(g_i^s, R)$ starting points remaining on the upper boundary.

For the remaining $f_{out}(g_i^s, L) - f_{in}(g_i^s, D)$ ending points on the left boundary and the remaining $f_{in}(g_i^s, U) - f_{out}(g_i^s, R)$ starting points on the upper boundary, we can construct $f_{out}(g_i^s, L) - f_{in}(g_i^s, D)$ parallel paths in the area framed with a green line on the upper left of Fig. 13(a). We construct a total of $f_{in}(g_i^s, D) + f_{out}(g_i^s, R) + f_{out}(g_i^s, L) - f_{in}(g_i^s, D) = f(g_i^s)$ disjoint paths, thus Lemma 1 is true in the second flow

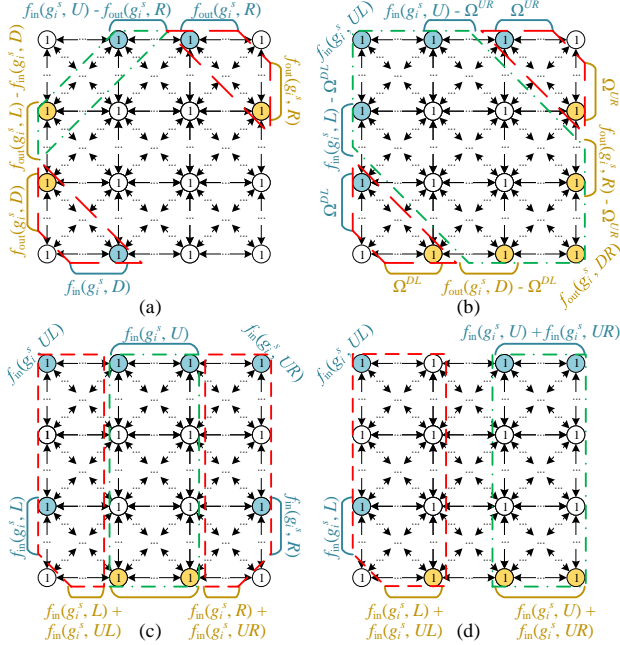


Fig. 13. (a)-(d) The divisions of the routing grid before compression for the second, fourth, fifth and seventh flow configurations, respectively. The blue numbers indicate the number of starting points and the yellow numbers indicate the number of ending points.

configuration.

We introduce Menger's theorem [30] to prove Lemma 1 is true. Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and two sets of nodes \mathcal{C} and \mathcal{D} , $\mathcal{C}, \mathcal{D} \subset \mathcal{V}$. A third set of nodes $\mathcal{W} \subset \mathcal{V}$ separates \mathcal{C} from \mathcal{D} if every path from a node in \mathcal{C} to a node in \mathcal{D} contains a node from \mathcal{W} . $\mathcal{P}(\mathcal{G}, \mathcal{C}, \mathcal{D})$ is the largest number of disjoint paths with starting node in \mathcal{A} , an ending node in \mathcal{B} , and no internal nodes in \mathcal{C} or \mathcal{D} . $\mathcal{K}(\mathcal{G}, \mathcal{C}, \mathcal{D})$ is the smallest number of nodes in a set that separates \mathcal{C} from \mathcal{D} .

Lemma 2 (Menger's Theorem): $\mathcal{P}(\mathcal{G}, \mathcal{C}, \mathcal{D}) = \mathcal{K}(\mathcal{G}, \mathcal{C}, \mathcal{D})$.

3) *The Third Flow Configuration:* In the third flow configuration shown in Fig. 12(c), the incoming flows come from an edge node and two corner nodes adjacent to that edge node, and outgoing flows point to the opposite edge node and the remaining corner nodes. We have $f(g_i^s) \leq N^S + 1$.

In the routing grid before compression, we choose $f(g_i^s)$ grid points on the upper boundary, the upper-left corner, and the upper-right corner as starting points and $f(g_i^s)$ grid points on the lower boundary, the lower-left corner, and the lower-right corner as ending points. We assume that the set of grid points \mathcal{W}^S can separate the starting points from the ending points. Obviously, the smallest number of grid points in \mathcal{W}^S is equal $\min(N^S + 1, f(g_i^s))$. Since $f(g_i^s) \leq N^S + 1$, $\min(N^S + 1, f(g_i^s)) = f(g_i^s)$. According to Lemma 2, we can construct $f(g_i^s)$ disjoint paths in the routing grid before compression, i.e., Lemma 1 is true in the third flow configuration.

4) *The Fourth Flow Configuration:* In the fourth flow configuration shown in Fig. 12(d), the incoming flows come from a corner node and two edge nodes adjacent to that corner node, and the outgoing flows point to the opposite corner node and the remaining edge nodes. We have $f(g_i^s) \leq 2 \times N^S - 1$. For illustration, we use Ω^{DL} and Ω^{UR} to represent $\min(f_{in}(g_i^s, U), f_{out}(g_i^s, R))$ and $\min(f_{in}(g_i^s, L), f_{out}(g_i^s, D))$, respectively.

In the routing grid before compression, we choose $f_{in}(g_i^s, L)$ grid points from lower to upper on the left boundary as the starting points of the paths and $f_{out}(g_i^s, D)$ grid points from left to right on the lower boundary as the ending points of the paths. Obviously, we can construct Ω^{DL} parallel paths from the lower boundary to the left boundary in the area framed with a red line at the lower left of Fig. 13(b). And there are $f_{in}(g_i^s, L) - \Omega^{DL}$ starting points remaining on the left boundary and $f_{out}(g_i^s, D) - \Omega^{DL}$ ending points remaining on the lower boundary.

Similarly, we choose $f_{in}(g_i^s, U)$ grid points from right to left on the upper boundary as the starting point of the paths and $f_{out}(g_i^s, R)$ grid points from upper to lower on the right boundary as the ending point of the paths. Obviously, we can construct Ω^{UR} parallel paths from the upper boundary to the right boundary in the area framed with a red line on the upper right of Fig. 13(b). And there are $f_{in}(g_i^s, U) - \Omega^{UR}$ starting points remaining on the upper boundary and $f_{out}(g_i^s, R) - \Omega^{UR}$ ending points remaining on the right boundary.

In the area framed with green line of Fig. 13(b), there are the remaining $f(g_i^s) - \Omega^{DL} - \Omega^{UR}$ starting points on the upper left and the $f(g_i^s) - \Omega^{DL} - \Omega^{UR}$ ending points on the lower right. Obviously, the smallest number of grid points in \mathcal{W}^S is equal $\min(2 \times N^S - 1 - \Omega^{DL} - \Omega^{UR}, f(g_i^s) - \Omega^{DL} - \Omega^{UR})$. Since $f(g_i^s) \leq 2 \times N^S - 1$, $\min(2 \times N^S - 1 - \Omega^{DL} - \Omega^{UR}, f(g_i^s) - \Omega^{DL} - \Omega^{UR}) = f(g_i^s) - \Omega^{DL} - \Omega^{UR}$. According to Lemma 2, we can construct at most $f(g_i^s) - \Omega^{DL} - \Omega^{UR}$ disjoint paths in the area framed with a green line of Fig. 13(b). We construct a total of $\Omega^{DL} + \Omega^{UR} + f(g_i^s) - \Omega^{DL} - \Omega^{UR} = f(g_i^s)$ disjoint paths, thus Lemma 1 is true in the fourth flow configuration.

5) *The Fifth Flow Configuration:* In the fifth flow configuration shown in Fig. 12(e), the outgoing flow points to the lower edge node, and the incoming flows come from the edge nodes and corner nodes that are not adjacent to the lower edge node. We have $f(g_i^s) \leq N^S - 1$.

In the routing grid before compression, we choose $f_{in}(g_i^s, L) + f_{in}(g_i^s, UL)$ grid points from left to right on the lower boundary as the ending points of the paths and choose $f_{in}(g_i^s, L)$ grid points from lower to upper on the left boundary and $f_{in}(g_i^s, UL)$ upper-left grid point as the starting points of the paths. We can construct $f_{in}(g_i^s, L) + f_{in}(g_i^s, UL)$ disjoint paths from the lower boundary to the left boundary and the upper-left corner in the area framed with a red line at the left of Fig. 13(c).

Similarly, we choose $f_{in}(g_i^s, R) + f_{in}(g_i^s, UR)$ grid points from right to left on the lower boundary as the ending points of the paths and choose $f_{in}(g_i^s, R)$ grid points from lower to upper on the right boundary and $f_{in}(g_i^s, UR)$ upper-right grid point as the starting points of the paths. We can construct $f_{in}(g_i^s, R) + f_{in}(g_i^s, UR)$ disjoint paths from the lower boundary to the right boundary and the upper-right corner in the area framed with a red line at the right of Fig. 13(c).

In the area framed with a green line at Fig. 13(c), we choose $f_{in}(g_i^s, U)$ grid points from right to left on the upper boundary as starting points and $f_{in}(g_i^s, U)$ grid points from right to left on the lower boundary as ending points. According to Lemma 2, we can construct $f_{in}(g_i^s, U)$ disjoint paths in the

area framed with a green line of Fig. 13(c). We construct a total of $f_{\text{in}}(g_i^s, L) + f_{\text{in}}(g_i^s, UL) + f_{\text{in}}(g_i^s, R) + f_{\text{in}}(g_i^s, UR) + f_{\text{in}}(g_i^s, U) = f(g_i^s)$ disjoint paths, thus Lemma 1 is true in the fifth flow configuration.

6) *The Sixth Flow Configuration:* In the sixth flow configuration shown in Fig. 12(f), the incoming flow comes from the lower edge node, and the outgoing flows point to the edge nodes and corner nodes that are not adjacent to the lower edge node. We have $f(g_i^s) \leq N^S - 1$. We can prove that Lemma 1 also holds in the sixth flow configuration in a way similar to the proof of the fifth flow configuration.

7) *The Seventh Flow Configuration:* In the seventh flow configuration shown in Fig. 12(g), the outgoing flows point to the lower edge node and the lower-right corner node, and the incoming flows come from the edge nodes and corner nodes that are not adjacent to these nodes. We have $f(g_i^s) \leq N^S$.

In the routing grid before compression, we choose $f_{\text{in}}(g_i^s, L) + f_{\text{in}}(g_i^s, UL)$ grid points from left to right on the lower boundary as the ending points of the paths and choose $f_{\text{in}}(g_i^s, L)$ grid points from lower to upper on the left boundary and $f_{\text{in}}(g_i^s, UL)$ upper-left grid point as the starting points of the paths. We can construct $f_{\text{in}}(g_i^s, L) + f_{\text{in}}(g_i^s, UL)$ disjoint paths from the lower boundary to the left boundary and the upper-left corner in the area framed with a red line of Fig. 13(d).

In the area framed with a green line of Fig. 13(d), we choose $f_{\text{in}}(g_i^s, U) + f_{\text{in}}(g_i^s, UR)$ grid points from right to left on the lower boundary as the ending points of the paths and $f_{\text{in}}(g_i^s, U) + f_{\text{in}}(g_i^s, UR)$ grid points from right to left on the upper boundary as the starting points of the paths. According to Lemma 2, we can construct $f_{\text{in}}(g_i^s, U) + f_{\text{in}}(g_i^s, UR)$ disjoint paths in the area framed with a green line of Fig. 13(c). We construct a total of $f_{\text{in}}(g_i^s, L) + f_{\text{in}}(g_i^s, UL) + f_{\text{in}}(g_i^s, U) + f_{\text{in}}(g_i^s, UR) = f(g_i^s)$ disjoint paths, thus Lemma 1 is true in the seventh flow configuration.

8) *The Eighth Flow Configuration:* In the eighth flow configuration shown in Fig. 12(h), the incoming flows come from the lower edge node and the lower-right corner node, and the outgoing flows point to the edge nodes and corner nodes that are not adjacent to these nodes. We have $f(g_i^s) \leq N^S$. We can prove that Lemma 1 also holds in the eighth flow configuration in a way similar to the proof of the Seventh flow configuration.

In conclusion, Lemma 1 holds in all flow configurations.

H. Flow Collocation

After obtaining the maximum flow, detailed routing needs to be performed to decide how wires pass through the super grids, tiles, and connection regions between electrode area and contact area, thus generating the final routing results. Note that wire crossings need to be avoided in the detailed routing to ensure the correctness of control-signal prorogation. Moreover, compared with tiles whose corners are already occupied by pins, super grids allow flows to pass through their edges. If two flows are allocated a shared edge between adjacent super grids, a wire crossing will be generated. Thus, the routing design regarding super grids needs to be considered

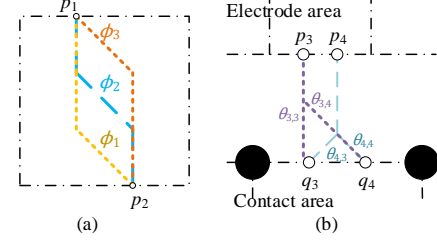


Fig. 14. The diagram of the routing track. (a) Routing tracks between p_1 and p_2 . (b) Routing tracks connecting the electrode area to the contact area.

systematically with new techniques. Based on Lemma 1, we further propose an ILP-based flow collocation algorithm to deal with the detailed routing considering super grids.

To determine the starting and ending points of wire segments that cross a super grid, we recover the grid points on the four sides of the super grid. We use the 0-1 variables λ_i^S and λ_i^E to represent whether the grid point p_i is the start or ending point of the wire segments, respectively. p_i cannot be simultaneously a starting point and an ending point, which can be constrained as

$$\lambda_i^S + \lambda_i^E \leq 1, \forall p_i \in P(g_h^s), \forall g_h^s \in G^s, \quad (2)$$

where $P(g_h^s)$ is the set of grid points located on the boundary of the super grid $g_h^s \in G^s$.

For a super grid $g_h^s \in G^s$, the value of the incoming flow from a edge node is equal to the number of starting points on the corresponding edge. Similarly, the value of outgoing flow to a edge node is equal to the number of ending points on the corresponding edge. Thus, we have the following constraints

$$\sum_{p_i \in P^\Delta(g_h^s)} \lambda_i^S = f_{\text{in}}(g_h^s, \Delta), \forall g_h^s \in G^s, \forall \Delta \in \{U, D, L, R\}, \quad (3)$$

$$\sum_{p_i \in P^\Delta(g_h^s)} \lambda_i^E = f_{\text{out}}(g_h^s, \Delta), \forall g_h^s \in G^s, \forall \Delta \in \{U, D, L, R\}, \quad (4)$$

where $P^U(g_h^s)$, $P^D(g_h^s)$, $P^L(g_h^s)$, and $P^R(g_h^s)$ are the sets of grid points located at the upper, lower, left, and right edge of g_h^s , respectively.

For a super grid $g_h^s \in G^s$, if the value of the incoming flow from a corner node is not 0, the corresponding grid point is the starting point of the wire segment. Similarly, if the value of the outgoing flow to a corner node is not 0, the corresponding grid point is the ending point of the wire segment. Thus, we have

$$\lambda_i^S = f_{\text{in}}(g_h^s, \Delta), \quad \forall g_h^s \in G^s, \forall \Delta \in \{UL, UR, DL, DR\} \wedge p_i = P^\Delta(g_h^s), \quad (5)$$

$$\lambda_i^E = f_{\text{out}}(g_h^s, \Delta), \quad \forall g_h^s \in G^s, \forall \Delta \in \{UL, UR, DL, DR\} \wedge p_i = P^\Delta(g_h^s), \quad (6)$$

where $P^{UL}(g_h^s)$, $P^{UR}(g_h^s)$, $P^{DL}(g_h^s)$, and $P^{DR}(g_h^s)$ are the grid points located in the upper-left, upper-right, lower-left, and lower-right of g_h^s , respectively.

There is one or more ways to connect any two grid points on the boundary of g_h^s . We name the possible routing paths as *Routing Tracks*. As shown in Fig. 14(a), there are three routing tracks from p_1 to p_2 . If p_i is the starting (or ending) point of

the wire segments, there is only one routing track starting (or ending) at p_i that is occupied by a wire. Thus, we have

$$\sum_{p_j \in P(g_h^s)} \sum_{\phi_k \in \Phi(p_i, p_j)} \mu_k = \lambda_i^S, \forall g_h^s \in G^s, \quad (7)$$

$$\sum_{p_i \in P(g_h^s)} \sum_{\phi_k \in \Phi(p_i, p_j)} \mu_k = \lambda_j^E, \forall g_h^s \in G^s, \quad (8)$$

where $\Phi(p_i, p_j)$ is a set of all routing tracks that start at p_i and end at p_j . μ_k is a 0-1 variable representing whether $\phi_k \in \Phi(p_i, p_j)$ is occupied by a wire or not.

Since the shape of g_h^s is fixed as a $n \times n$ grid, all possible routing tracks can be established for the constructed super grids. To avoid wire crossings, we use a 0-1 variable $O(\phi_k, \phi_{k'})$ to represent whether a pair of routing tracks ϕ_k and $\phi_{k'}$ overlap with each other. When ϕ_k and $\phi_{k'}$ are in the same super grid and $O(\phi_k, \phi_{k'}) = 1$, ϕ_k and $\phi_{k'}$ cannot be occupied by wires at the same time. Thus, we have

$$\begin{aligned} \mu_k + \mu_{k'} &\leq 2 - O(\phi_k, \phi_{k'}), \\ \forall \phi_k \in \Phi(p_i, p_j), \forall \phi_{k'} \in \Phi(p_{i'}, p_{j'}), \\ \forall p_i, p_j, p_{i'}, p_{j'} \in P(g_h^s), \forall g_h^s \in G^s. \end{aligned} \quad (9)$$

Similarly, when routing tracks ϕ_k and $\phi_{k'}$ are in adjacent super grids and $O(\phi_k, \phi_{k'}) = 1$, we have

$$\begin{aligned} \mu_k + \mu_{k'} &\leq 2 - O(\phi_k, \phi_{k'}), \\ \forall \phi_k \in \Phi(p_i, p_j), \forall \phi_{k'} \in \Phi(p_{i'}, p_{j'}), \forall g_h^s \in G^s, \\ \forall g_{h'}^s \in AC(g_h^s), \forall p_{i'}, p_{j'} \in P(g_{h'}^s), \forall p_i, p_j \in P(g_h^s), \end{aligned} \quad (10)$$

where $AC(g_h^s)$ is the set of adjacent super grids of g_h^s .

We model the flow assignment problem for the contact area in a similar way. For any tile t_h , we first add O-cap points, called *Tile Points*, on each edge to serve as the starting or ending points of the wire segments. In addition, we also treat the contact pad on each corner of the tile as a tile point to serve as the ending point of the wire segments. The 0-1 variables ξ_i^S and ξ_i^E are introduced to represent whether the tile point q_i is the starting or ending point of the wire segments, respectively. q_i cannot be simultaneously the start and ending point, which can be constrained as

$$\xi_i^S + \xi_i^E \leq 1, \forall q_i \in Q(t_h), \forall t_h \in T^D \cup T^U, \quad (11)$$

where $Q(t_h)$ is the set of tile points located on the boundary of t_h . T^D and T^U are the sets of all tiles located in the upper and lower contact area, respectively.

The value of the incoming (outgoing) flow through an edge of the tile and the number of starting (ending) points on the corresponding edge of the tile can be constrained as

$$\begin{aligned} \sum_{q_i \in Q^\Delta(t_h)} \xi_i^S &= f_{\text{in}}(t_h, \Delta), \\ \forall t_h \in T^D \cup T^U, \forall \Delta \in \{U, D, L, R\}, \end{aligned} \quad (12)$$

$$\begin{aligned} \sum_{q_i \in Q^\Delta(t_h)} \xi_i^E &= f_{\text{out}}(t_h, \Delta), \\ \forall t_h \in T^D \cup T^U, \forall \Delta \in \{U, D, L, R\}, \end{aligned} \quad (13)$$

where $Q^U(t_h)$, $Q^D(t_h)$, $Q^L(t_h)$, and $Q^R(t_h)$ are the tile points located at the upper, lower, left, and right edge of t_h , respectively.

The upper-left, upper-right, lower-left, and lower-right tile points of t_h can only be used as the ending point of the wire segment. If the the upper-left, upper-right, lower-left, or lower-right outgoing flow value of t_h is not 0, the corresponding tile points is the ending point of the wire segment. Thus, we have

$$\begin{aligned} \xi_i^E &= f_{\text{out}}(t_h, \Delta), \\ \forall t_h \in T^D \cup T^U, \forall \Delta \in \{DL, DR, UL, UR\} \wedge q_i = Q^\Delta(t_h), \end{aligned} \quad (14)$$

where $Q^{UL}(t_h)$, $Q^{UR}(t_h)$, $Q^{DL}(t_h)$, and $Q^{DR}(t_h)$ are the tile points located in the upper-left, upper-right, lower-left, and lower-right corners of t_h , respectively.

We can also build corresponding routing tracks for each tile. If q_i is the starting (or ending) point of the wire segments, there is only one routing track starting (or ending) at q_i that is occupied by a wire. Thus, we have

$$\sum_{q_j \in Q(t_h)} \sum_{\phi_k \in \Phi(q_i, q_j)} \nu_k = \xi_i^S, \forall t_h \in T^D \cup T^U, \quad (15)$$

$$\sum_{q_i \in Q(t_h)} \sum_{\phi_k \in \Phi(q_i, q_j)} \nu_k = \xi_j^E, \forall t_h \in T^D \cup T^U, \quad (16)$$

where $\Phi(q_i, q_j)$ is the set of routing tracks that start at q_i and end at q_j . ν_k is a 0-1 variable representing whether $\phi_k \in \Phi(q_i, q_j)$ is occupied by a wire or not.

For any two routing tracks ϕ_k and $\phi_{k'}$ in the same tile, if $O(\phi_k, \phi_{k'}) = 1$, they cannot be occupied by wires at the same time. Thus, we have

$$\begin{aligned} \nu_k + \nu_{k'} &\leq 2 - O(\phi_k, \phi_{k'}), \\ \forall \phi_k \in \Phi(q_i, q_j), \forall \phi_{k'} \in \Phi(q_{i'}, q_{j'}), \\ \forall q_i, q_j, q_{i'}, q_{j'} \in Q(t_h), \forall t_h \in T^D \cup T^U. \end{aligned} \quad (17)$$

A group of routing tracks is constructed to connect the electrode area to the contact area. The routing tracks, that connect the electrode area to the lower contact area, start at a grid point on the lower boundary of the electrode area and end at a tile point on the upper boundary of the lower contact area, as shown in Fig. 14(b). We use $\theta_{i,j}$ to represent the routing track starting at p_i and ending at q_j . $\rho_{i,j}$ is a 0-1 variable that represents whether $\theta_{i,j}$ is occupied by a wire or not. Thus, we have

$$\sum_{q_j \in Q(T^D, U)} \rho_{i,j} = \lambda_i^E, \forall p_i \in P(G^s, D), \quad (18)$$

$$\sum_{p_i \in P(G^s, D)} \rho_{i,j} = \xi_j^S, \forall q_j \in Q(T^D, U), \quad (19)$$

where $P(G^s, D)$ is the set of grid points located at the lower boundary of the electrode area. $Q(T^D, U)$ is the set of tile points located at the upper boundary of lower contact area.

To avoid wire crossings, we further use a 0-1 variable $O(\rho_{i,j}, \rho_{i',j'})$ to represent whether tracks $\theta_{i,j}$ and $\theta_{i',j'}$ overlap with each other. Then, we have

$$\begin{aligned} \rho_{i,j} + \rho_{i',j'} &\leq 2 - O(\rho_{i,j}, \rho_{i',j'}), \\ \forall p_i, p_{i'} \in P(G^s, D), \forall q_j, q_{j'} \in Q(T^D, U). \end{aligned} \quad (20)$$

TABLE I
DETAILS OF BENCHMARKS USED IN THE EXPERIMENTS

Benchmark (N_E , S_E (mm ²), S_P (mm ²), N_P)		
Test Case 1 (5, 3883, 852, 256)	Test Case 2 (7, 3883, 852, 256)	Test Case 3 (13, 3883, 852, 256)
Test Case 4 (23, 3883, 852, 256)	Test Case 5 (44, 3883, 852, 256)	Test Case 6 (88, 3883, 852, 256)
Test Case 7 (200, 7766, 1703, 512)	Test Case 8 (250, 7766, 1703, 512)	Test Case 9 (300, 7766, 1703, 512)
Test Case 10 (350, 7766, 1703, 512)	Test Case 11 (400, 15533, 3406, 1024)	Test Case 12 (600, 15533, 3406, 1024)
Dilution Chip 1 (34, 3883, 852, 216)		Dilution Chip 2 (100, 3883, 852, 216)

Similarly, the routing tracks, that connect the electrode area to the upper contact area, start at a grid point on the upper boundary of the electrode area and end at a tile point on the lower boundary of the upper contact area. We have the following constraints

$$\sum_{q_j \in Q(T^U, D)} \rho_{i,j} = \lambda_i^E, \forall p_i \in P(G^s, U), \quad (21)$$

$$\sum_{p_i \in P(G^s, U)} \rho_{i,j} = \xi_j^S, \forall q_j \in Q(T^U, D), \quad (22)$$

$$\rho_{i,j} + \rho_{i',j'} \leq 2 - O(\rho_{i,j}, \rho_{i',j'}), \quad (23)$$

$$\forall p_i, p_{i'} \in P(G^s, U), \forall q_j, q_{j'} \in Q(T^U, D),$$

where $P(G^s, U)$ is the set of grid points located at the upper boundary of the electrode area. $Q(T^U, D)$ is the set of tile points located at the lower boundary of upper contact area.

Finally, optimized routing path with minimized wire length and non-intersection of wires can be generated by solving the following optimization problem

$$\begin{aligned} & \text{Minimize} \quad \sum_{p_i \in P(G^s, D)} \sum_{q_i \in Q(T^D, U)} \rho_{i,j} \times L(\theta_{i,j}) \\ & + \sum_{p_i \in P(G^s, U)} \sum_{q_i \in Q(T^U, D)} \rho_{i,j} \times L(\theta_{i,j}) \\ & + \sum_{g_h^s \in G^s} \sum_{p_i, p_j \in P(g_h^s)} \sum_{\phi_k \in \Phi(p_i, p_j)} \mu_k \times L(\phi_k) \\ & + \sum_{t_h \in T^U \cap T^D} \sum_{q_i, q_j \in Q(t_h)} \sum_{\phi_k \in \Phi(q_i, q_j)} \nu_k \times L(\phi_k) \end{aligned} \quad (24)$$

$$\text{subject to (2) - (23)}, \quad (25)$$

where $L(\theta_{i,j})$ and $L(\phi_k)$ are the length of $\theta_{i,j}$ and ϕ_k , respectively.

IV. EXPERIMENTAL RESULTS

NR-Router+ was implemented in Python and tested on a PC with 3.4 GHz CPU and 24GB memory. The minimum cost flow and the ILP problem formulated in Section III are solved by the OR-Tools [31]. We used 14 benchmarks to verify the proposed method as shown in Table I. Parameters N_E , S_E , S_P , and N_P in Table I are the number of electrodes, the size of the electrode area, the size of the contact area, and the number of contact pads, respectively. The two dilution EWOD chips are real-life biochemical applications, and the other 12 synthetic benchmarks were generated using our developed

design interface. These benchmarks contain a number of electrodes with various shapes. Among them, test cases 7-12 use larger chip sizes to accommodate more electrodes and pad contacts. Moreover, the parameters used in the experiments are set as follows: $w_l = 100 \mu\text{m}$, $s_l = 5 \mu\text{m}$, $w_g = 150 \mu\text{m}$, $N^S = 3$, O-cap = 2, and D-cap = 4.

A. Validation of the Proposed NR-Router+

Since there is no existing work targeting the problem solved in this paper, we implemented an exhaustive algorithm and an A* algorithm in the preliminary version for comparison [1]. We ran the three algorithms on the aforementioned benchmarks, including two dilution EWOD chips and 12 synthetic benchmarks. Table II shows the experimental results, where N. Average provides the average ratio between the comparison method and NR-Router+ across all the benchmarks. It can be seen from Table II that all the algorithms achieve 100% routability in the benchmarks.

Compared with the exhaustive algorithm, it can be seen from Table II that NR-Router+ achieves better results in both total wirelength and CPU time. We analyze this result for the following reasons: 1) The exhaustive algorithm performs routing design based on a routing mesh using the proximity principle between electrode area and contact area. In contrast, NR-Router+ further expands connection options between electrodes and contact pads in a fine-grained level using the ILP-based flow collocation algorithm, thus achieving shorter wirelength and 2) since the exhaustive algorithm needs to enumerate all possible combinations of candidate pins on the routing mesh, it takes a relatively long time to evaluate a large number of feasible solutions.

In addition, compared with the A* algorithm, NR-Router+ achieves an average reduction of 50.97% regarding the wirelength and a 11.6X speedup of CUP time on average. Although the A* algorithm shows excellent performance in finding a single shortest path between nodes, the optimal routing paths considering all the pin-pad pairs in a global manner cannot be generated. In particular, for benchmarks with a large number of electrodes, since the previously routed paths will become obstacles to the subsequent routing designs, the time consumed for path computation is increased significantly. In contrast, since the grids occupied by electrodes are removed from the constructed flow network, the computation overhead of NR-Router+ is reduced significantly with the increase in the number of electrodes, thus demonstrating its better scalability.

B. Comparison Between NR-Router [1] and NR-Router+

To further verify the effectiveness of NR-Router+, this section presents comparisons between NR-Router+ and the previous version of NR-Router [1]. Due to the adoption of several new techniques, e.g., the grid reduction strategy and the ILP-based flow collocation algorithm, key indicators such as the computation efficiency and the scalability of NR-Router can be improved systematically. Similarly, We ran both algorithms on the same PC using the aforementioned benchmarks. Table III shows the corresponding comparison results with respect to wirelength and CPU time.

TABLE II
WIRELENGTH AND CPU TIME OF NR-ROUTER+ AND TWO BASLINES SOLUTIONS

Benchmark		Test Case 1	Test Case 2	Test Case 3	Test Case 4	Test Case 5	Test Case 6	Dilution Chip 1	Dilution Chip 2	N. Average
Exhaustive Algorithm [1]	Wirelength (μm)	89374	129836	173946	539274	1304753	2183741	593762	2243131	1.0143
	CPU Time (s)	>1 hour	>1 hour	>1 hour	>1 hour	>1 hour	>1 hour	>1 hour	>1 hour	>100
A* Algorithm [1]	Wirelength (μm)	128475	182736	397364	1984654	2794854	4395862	2594832	3418597	2.4048
	CPU Time (s)	0.2374	0.5920	1.0943	1.4284	2.5573	3.6716	2.4343	3.5392	11.5847
NR-Router+	Wirelength (μm)	89124	127278	173863	536977	1292989	2172773	556634	2230076	1.0000
	CPU Time (s)	0.1660	0.1992	0.1946	0.2062	0.1417	0.1912	0.1280	0.1819	1.0000

TABLE III
WIRELENGTH AND CPU TIME OF NR-ROUTER AND NR-ROUTER+

Benchmark		Test Case 1	Test Case 2	Test Case 3	Test Case 4	Test Case 5	Test Case 6	Dilution Chip 1	N. Average
Wirelength (μm)	NR-Router	89374	129836	173946	539274	1304753	2183741	593762	
	NR-Router+	89124	127278	173863	536977	1292989	2172773	556634	
	Ratio	1.0028	1.0201	1.0005	1.0043	1.0091	1.0050	1.0667	
Benchmark		Test Case 7	Test Case 8	Test Case 9	Test Case 10	Test Case 11	Test Case 12	Dilution Chip 2	1.0155
Wirelength (μm)	NR-Router	3490540	4601247	5581822	6204942	7286310	11108725	2243131	
	NR-Router+	3460042	4513620	5448888	6060968	7250295	10881327	2230076	
	Ratio	1.0088	1.0194	1.0244	1.0238	1.0050	1.0209	1.0059	
Benchmark		Test Case 1	Test Case 2	Test Case 3	Test Case 4	Test Case 5	Test Case 6	Dilution Chip 1	N. Average
CPU Time (s)	NR-Router	0.4624	0.4981	0.5244	0.5221	0.6483	0.6739	0.5937	
	NR-Router+	0.1660	0.1992	0.1946	0.2062	0.1417	0.1912	0.1280	
	Ratio	2.7855	2.5005	2.6948	2.5320	4.5752	3.5246	4.6383	
Benchmark		Test Case 7	Test Case 8	Test Case 9	Test Case 10	Test Case 11	Test Case 12	Dilution Chip 2	2.6150
CPU Time (s)	NR-Router	2.9418	4.0645	6.3019	8.9143	7.8237	15.9363	0.6894	
	NR-Router+	2.2283	3.1239	3.8083	4.0152	5.9103	9.1132	0.1819	
	Ratio	1.3202	1.3011	1.6548	2.2201	1.3237	1.7487	3.7900	

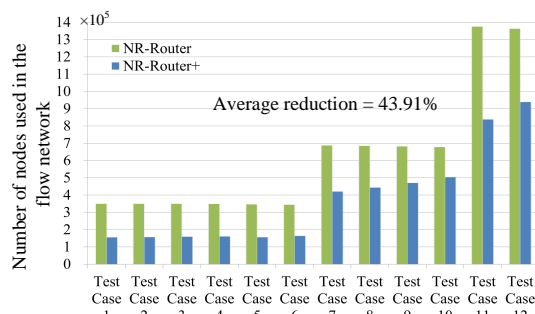


Fig. 15. Comparison results between NR-Router and NR-Router+ with respect to the number of nodes used in the flow network.

Compared with NR-Router, the computational efficiency is also improved by NR-Router+ across all the benchmarks. It can be seen that the maximum speedup reaches up to 4.6X with an average speedup of 2.6X. This is mainly because the introduction of the grid reduction strategy simplifies the flow network significantly. As shown in Fig. 15 and Fig. 16, NR-Router+ achieves a 43.91% and a 66.17% reduction on average in terms of the number of nodes and edges used in the flow network compared with NR-Router, respectively. Moreover, it can be seen from Table III that NR-Router+ achieves a 0.05%-6.25% reduction of wirelength with an average reduction of 1.50%. This is mainly because NR-Router+ constructed a more detailed connection network between the electrode area and the contact area.

Furthermore, the simplification of the flow network also reduces the times of solver calling significantly in the proposed method. As shown in Table IV, compared with NR-Router, the time for solver calling as a percentage of the total CPU time in NR-Router+ is reduced significantly from 56.14% to 38.67% on average. Note that column Time of NR-Router+ in Table IV includes the calling times of both the minimum

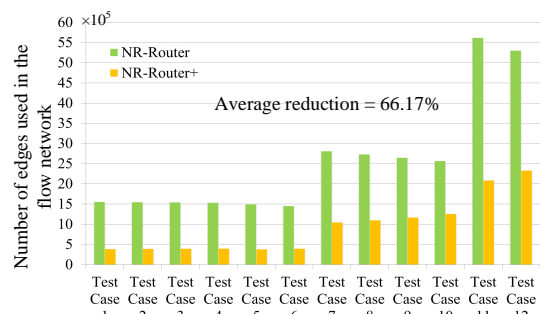


Fig. 16. Comparison results between NR-Router and NR-Router+ with respect to the number of edges used in the flow network.

cost flow algorithm and the flow collocation method. It can be seen that the proposed flow collocation method can generate the optimal solutions within a short period of time across all the benchmarks. This result, once again, demonstrates the robustness and scalability of the proposed NR-Router+. Finally, Fig. 17–Fig. 19 show the routing solutions of a synthetic EWOD chip and two real-life dilution EWOD chips, respectively. It can be seen that all the routing paths are constructed correctly on a single layer without any violation of design rules, and these solutions can be used directly for chip manufacturing.

V. CONCLUSIONS

We have presented NR-Router+, the first automation method that can realize accurate routing design of single-layer EWOD chips with non-regular electrodes. In NR-Router+, a minimum-cost flow algorithm followed by a light-weight model is proposed to efficiently construct optimal routing paths between electrodes and contact pads. Moreover, several techniques such as a grid reduction strategy and a flow allocation

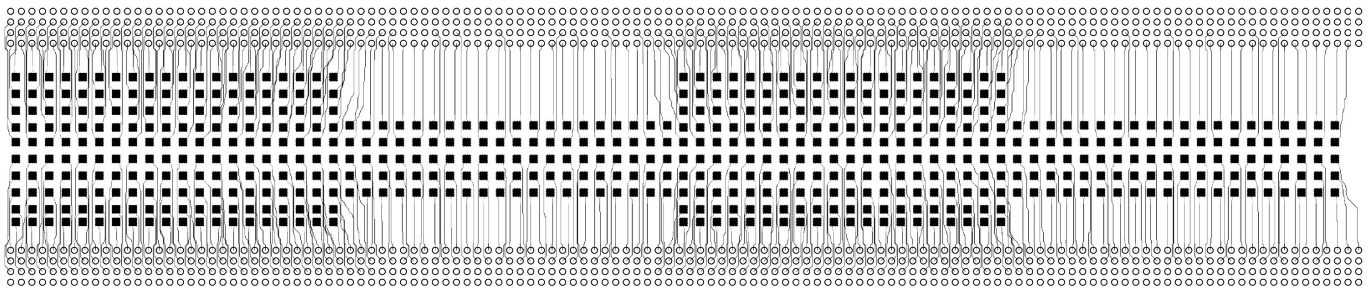


Fig. 17. The routing result of test case 12.

TABLE IV
CPU TIME OF SOLVER CALLING IN NR-ROUTER AND NR-ROUTER+

Benchmark	NR-Router		NR-Router+		
	Time (s)	Percentage (%)	Time (s) [#]	Percentage (%)	Imp (%)
Test Case 1	0.1641	35.49	0.0321 + 0.0234	33.43	5.80
Test Case 2	0.2096	42.08	0.0505 + 0.0287	39.76	5.51
Test Case 3	0.2491	47.50	0.0440 + 0.0326	39.36	17.14
Test Case 4	0.3034	58.11	0.0614 + 0.0425	50.39	13.29
Test Case 5	0.5360	82.68	0.0579 + 0.0255	58.86	28.81
Test Case 6	0.4973	73.79	0.0676 + 0.0443	58.53	20.68
Test Case 7	1.5659	53.23	0.3756 + 0.1981	25.75	51.63
Test Case 8	1.9664	48.38	0.6846 + 0.3612	33.48	30.80
Test Case 9	3.6921	58.59	0.8316 + 0.3705	31.57	46.12
Test Case 10	6.0297	67.64	0.5472 + 0.3757	22.99	66.01
Test Case 11	3.5883	45.86	1.8189 + 0.3525	36.74	19.89
Test Case 12	6.1521	38.60	1.8648 + 0.8194	29.45	35.70
Dilution Chip 1	0.3303	55.63	0.0266 + 0.0191	35.70	35.83
Dilution Chip 2	0.5403	78.37	0.0469 + 0.0357	45.41	42.06
Average		56.14		38.67	29.09

[#] The execution time of the minimum cost flow algorithm and the flow collocation method.

algorithm are proposed to further improve the overall performance of NR-Router+. Mask files feasible for manufacturing can also be generated automatically. Experimental results on both real-life chip applications and synthetic benchmarks have confirmed the effectiveness of NR-Router+.

REFERENCES

- [1] H.-C. Huang, C.-C. Liang, Q. Wang, X. Huang, T.-Y. Ho, and C.-J. Kim, "Nr-router: Non-regular electrode routing with optimal pin selection for electrowetting-on-dielectric chips," in *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2022, pp. 56–61.
- [2] P. Y. Keng, S. Chen, H. Ding, S. Sadeghi, G. J. Shah, A. Dooraghi, M. E. Phelps, N. Satyamurthy, A. F. Chatzivoannou, C.-J. Kim *et al.*, "Microchemical synthesis of molecular probes on an electronic microfluidic device," *Proceedings of the National Academy of Sciences*, vol. 109, no. 3, pp. 690–695, 2012.
- [3] H.-H. Shen, S.-K. Fan, C.-J. Kim, and D.-J. Yao, "Ewod microfluidic systems for biomedical applications," *Microfluidics and Nanofluidics*, vol. 16, pp. 965–987, 2014.
- [4] W. C. Nelson and C.-J. Kim, "Droplet actuation by electrowetting-on-dielectric (ewod): A review," *Journal of Adhesion Science and Technology*, vol. 26, no. 12-17, pp. 1747–1771, 2012.
- [5] E. Samiei, M. Tabrizian, and M. Hoorfar, "A review of digital microfluidics as portable platforms for lab-on a-chip applications," *Lab on a Chip*, vol. 16, no. 13, pp. 2376–2396, 2016.
- [6] J. Chen, Y. Yu, J. Li, Y. Lai, and J. Zhou, "Size-variable droplet actuation by interdigitated electrowetting electrode," *Applied Physics Letters*, vol. 101, no. 23, p. 234102, 2012.
- [7] X. Rui, S. Song, W. Wang, and J. Zhou, "Applications of electrowetting-on-dielectric (ewod) technology for droplet digital pcr," *Biomicrofluidics*, vol. 14, no. 6, p. 061503, 2020.
- [8] A. N. Davis, K. Samlali, J. B. Kapadia, J. Perreault, S. C. Shih, and N. Khanna, "Digital microfluidics chips for the execution and real-time monitoring of multiple ribozymatic cleavage reactions," *ACS omega*, vol. 6, no. 35, pp. 22514–22524, 2021.
- [9] L. Malic, T. Veres, and M. Tabrizian, "Nanostructured digital microfluidics for enhanced surface plasmon resonance imaging," *Biosensors and Bioelectronics*, vol. 26, no. 5, pp. 2053–2059, 2011.
- [10] C. Dong, L. Liu, H. Liu, W. Guo, X. Huang, S. Lian, X. Liu, and T.-Y. Ho, "A survey of dmfb security: State-of-the-art attack and defense," in *2020 21st International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2020, pp. 14–20.
- [11] W. Guo, S. Lian, C. Dong, Z. Chen, and X. Huang, "A survey on security of digital microfluidic biochips: Technology, attack, and defense," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 27, no. 4, pp. 1–33, 2022.
- [12] N. J. B. Nikapitiya, S. M. You, and H. Moon, "Droplet dispensing and splitting by electrowetting on dielectric digital microfluidics," in *2014 IEEE 27th International Conference on Micro Electro Mechanical Systems (MEMS)*. IEEE, 2014, pp. 955–958.
- [13] N. J. B. Nikapitiya, M. M. Nahar, and H. Moon, "Accurate, consistent, and fast droplet splitting and dispensing in electrowetting on dielectric microfluidics," *Micro and Nano Systems Letters*, vol. 5, p. 24, 2017.
- [14] J. Berthier and C. Peponnet, "A model for the determination of the dimensions of dents for jagged electrodes in electrowetting on dielectric microsystems," *Biomicrofluidics*, vol. 1, no. 1, p. 014104, 2007.
- [15] X. Xu, L. Sun, L. Chen, Z. Zhou, J. Xiao, and Y. Zhang, "Electrowetting on dielectric device with crescent electrodes for reliable and low-voltage droplet manipulation," *Biomicrofluidics*, vol. 8, no. 6, p. 064107, 2014.
- [16] J. Li, S. Chen, and C.-J. Kim, "Low-cost and low-topography fabrication of multilayer interconnections for microfluidic devices," *Journal of Micromechanics and Microengineering*, vol. 30, no. 7, p. 077001, 2020.
- [17] J. Li and C.-J. Kim, "Cybermanufacturing ecosystem for expanding electrowetting community," in *Abstract Book of the 11th International Conference on Electrowetting and Drop Dynamics on Functionalized Surfaces*, 2018.
- [18] C.-J. Kim, "Democratizing ewod digital microfluidics by cybermanufacturing," in *Proceedings of IEEE International Conference on Nano/Micro Engineered and Molecular Systems (IEEE-NEMS 2020)*. IEEE, 2021, p. 14.
- [19] —, "Democratizing ewod digital microfluidics," in *Proceedings of International Conference on Solid-State Sensors and Actuators (Transducers 2021)*. IEEE, 2021, p. 4.
- [20] X. Huang, C.-C. Liang, J. Li, T.-Y. Ho, and C.-J. Kim, "Open-source incubation ecosystem for digital microfluidics status and roadmap," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2019, pp. 1–6.
- [21] L. Zhang, Z. Li, X. Huang, and K. Chakrabarty, "Enhanced built-in self-diagnosis and self-repair techniques for daisy-chain design in meda digital microfluidic biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023.
- [22] T.-W. Huang, S.-Y. Yeh, and T.-Y. Ho, "A network-flow based pin-count aware routing algorithm for broadcast-addressing ewod chips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 12, pp. 1786–1799, 2011.
- [23] T.-W. Huang and T.-Y. Ho, "A fast routability-and performance-driven droplet routing algorithm for digital microfluidic biochips," in *2009 IEEE International Conference on Computer Design*. IEEE, 2009, pp. 445–450.
- [24] —, "A two-stage ilp-based droplet routing algorithm for pin-constrained digital microfluidic biochips," in *Proceedings of the 19th international symposium on Physical design*, 2010, pp. 201–208.
- [25] Q. Wang, Z. Li, H. Cheong, O.-S. Kwon, H. Yao, T.-Y. Ho, K. Shin, B. Li, U. Schlichtmann, and Y. Cai, "Control-fluidic codesign for paper-based digital microfluidic biochips," in *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. ACM, 2016, pp. 1–8.
- [26] T.-W. Huang, T.-Y. Ho, and K. Chakrabarty, "Reliability-oriented broadcast electrode-addressing for pin-constrained digital microfluidic biochips," in *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2011, pp. 448–455.
- [27] T. Yan and M. D. Wong, "A correct network flow model for escape routing," in *Proceedings of the 46th Annual Design Automation Conference*, 2009, pp. 332–335.

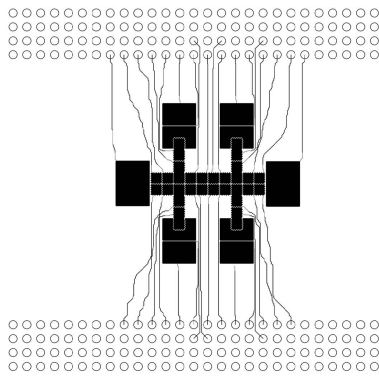


Fig. 18. The routing results of dilution chip 1.

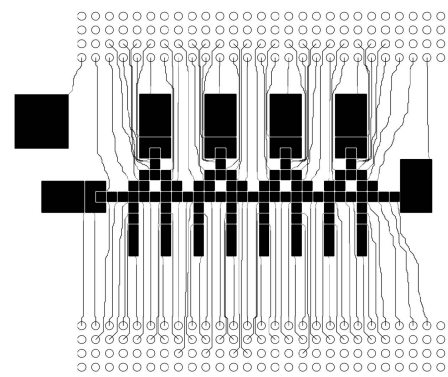


Fig. 19. The routing results of dilution chip 2.

- [28] W.-T. Chan, F. Y. Chin, and H.-F. Ting, "A faster algorithm for finding disjoint paths in grids," in *Algorithms and Computation: 10th International Symposium, ISAAC99 Chennai, India, December 16–18, 1999 Proceedings*. Springer, 2000, pp. 393–402.
- [29] J.-W. Fang and Y.-W. Chang, "Area-i/o flip-chip routing for chip-package co-design," in *2008 IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 2008, pp. 518–522.
- [30] K. Menger, "Zur allgemeinen kurventheorie," *Fundamenta Mathematicae*, vol. 10, no. 1, pp. 96–115, 1927.
- [31] Or-tools. [Online]. Available: <https://developers.google.com/optimization/>.