

# An ILP-based Routing Algorithm for Pin-Constrained EWOD Chips With Obstacle Avoidance

Jia-Wen Chang, Sheng-Han Yeh, Tsung-Wei Huang, and Tsung-Yi Ho, *Senior Member, IEEE*

**Abstract**—Electrowetting-on-dielectric (EWOD) chips have become the most popular actuators, particularly for droplet-based digital microfluidic biochip (DMFB) systems. In order to enable the electrical manipulations, wire routing is a key problem in designing EWOD chips. Unlike traditional very-large-scale-integration (VLSI) routing problems, in addition to routing-path establishment on signal pins, the pin-constrained EWOD-chip routing problem must address the issue of signal sharing for pin-count reduction under a practical constraint posed by a limited pin-count supply. Moreover, EWOD-chip designs might incur several obstacles in the routing region due to embedded devices for specific fluidic protocols. However, no existing work considers the EWOD-chip routing with obstacles and, therefore, lots of manual design efforts are involved. To remedy this insufficiency, we propose in this paper the first routing algorithm for pin-constrained EWOD chips with obstacle avoidance. The proposed algorithm, based on effective integer-linear-programming (ILP) formulation as well as efficient routing framework, can achieve high routability with a low design complexity. Experimental results based on real-life chips with obstacles demonstrate the high routability of proposed algorithm for pin-constrained EWOD chips with obstacle avoidance.

**Index Terms**—Digital microfluidic biochips, integer linear programming, pin-constrained, routing.

## I. INTRODUCTION

**E**LECTROWETTING-on-dielectric (EWOD) chips have emerged as the most widely used actuators particularly for droplet-based digital microfluidic (DMF) platforms [17]. EWOD chips enable the electrical manipulations of droplets on

Manuscript received August 5, 2012; revised December 21, 2012 and March 31, 2013; accepted June 5, 2013. Date of current version October 16, 2013. The work of T.-Y. Ho was supported in part by the Taiwan National Science Council under Grant NSC 101-2220-E-006-016 and Grant NSC 101-2628-E-006-018-MY3, and the Ministry of Education, Taiwan, under the NCKU Aim for the Top University Project Promoting Academic Excellence and Developing World Class Research Centers. A preliminary version of this paper was presented and nominated for the Best Paper Award at the IEEE/ACM 17<sup>th</sup> Asia and South Pacific Design Automation Conference, Sydney, Australia, in January 2012. This paper was recommended by Associate Editor Y. Chen.

J.-W. Chang, S.-H. Yeh, and T.-Y. Ho are with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: jwchang@eda.csie.ncku.edu.tw; hardyyeh@eda.csie.ncku.edu.tw; tyho@csie.ncku.edu.tw).

T.-W. Huang is with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA (e-mail: twhuang@eda.csie.ncku.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2013.2269767

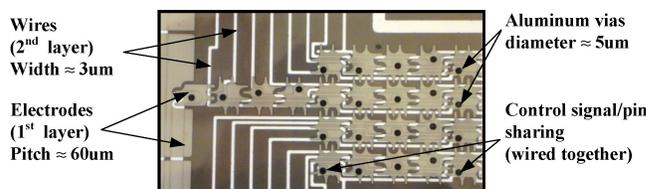


Fig. 1. Schematic view of an EWOD chip.

a two-dimensional (2-D) microfluidic array. This approach offers advantages of flexibility, accuracy, parallel processing, and automated controls [11]. These advantages are increasing the practicality of applications on miniaturized DMF platforms, including immunoassays, DNA sequencing, and point-of-care diagnosis [18].

The general diagram of a typical EWOD chips is schematically presented in Fig. 1 [15], [17]. This chip comprises two major layers of 2-D electrodes patterned in the first layer and conduction wires routed in the second layer, as well as an interinsulator of silicon dioxide for via hole patterning. Through these electrical components, the external controller drives these electrodes by assigning time-varying actuation voltage to generate electrohydrodynamic force. Hence, droplet manipulations can be performed in a reconfigurable manner as a result of EWOD phenomenon [5], [16].

Typically, the regular design flow of EWOD chips consists of three major stages; electrode-addressing, routing (i.e., wire routing), and fabrication [15]. This paper focuses on automated designs of electrode-addressing and routing, which are two key stages that determine the manufacturing complexity and fabrication cost [11]. Electrode-addressing is the method through which electrodes are addressed with control pins to identify input signals. Early EWOD-chip designs relied on direct addressing [10], in which each electrode is directly addressed with an independent control pin. This addressing scheme maximizes the flexibility of electrode controls. However, because the control pins are actuated by an external controller that supplies a limited number of signal ports, it is infeasible to actuate a large number of control pins, especially for high-density electrode arrays. To comply with the limited pin-count supply, pin-constrained electrode-addressing has been introduced as a solution to this problem. A prevailing approach, broadcast addressing, reduces the number of control

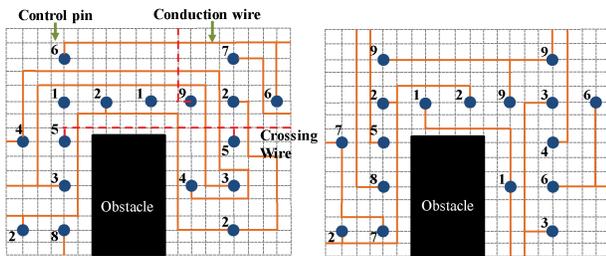


Fig. 2. Comparison of two different design methods for performing the same droplet controls. (a) Separate consideration of electrode-addressing and routing. (b) Simultaneous consideration of electrode-addressing and routing.

pins by assigning a single control pin to multiple electrodes with mutually compatible control signals [19].

After electrodes are appropriately addressed with control pins, conduction wires must be routed to establish connections between pins and signal ports. This routing problem has become more critical than ever for modern EWOD-chip designs, which must consider several routing obstacles from permanently embedded devices for specific fluidic protocols [6]. For example, a DNA sequencing chip may embed several electrophoresis devices for fast and accurate sample isolation; DNA amplification protocols require on-chip sensors to monitor temperature variation in each amplification cycle; immunoassay protocols require on-chip magnets to capture antibodies; protein or DNA analysis require on-chip electrophoresis equipments to separate and identify individual components (i.e., ions and particles) in reaction products, etc [7]. As these devices are independent from EWOD actuations, they are typically regarded as on-chip obstacles. During EWOD-chip routing, conduction wires should not be routed through these obstacles, thereby increasing the problem complexity. Therefore, current manual design methods might suffer from either poor solution quality or time-consuming human effort without the assistance of CAD tools.

Despite the interdependence between electrode-addressing and routing, most EWOD-chip design methods treat the two problems as being independent. This potential design gap becomes even more critical when the concerns of routing obstacles are involved. Specifically, if the electrode-addressing and routing with obstacle avoidance are not simultaneously considered, the feasibility and quality of resulting routing solution may be inevitably limited. For example, Fig. 2 shows two routing solutions under two different design methods that perform the same droplet controls. In the case of Fig. 2(a), which is an infeasible routing solution due to separate consideration of electrode-addressing and routing, additional postprocesses such as electrode readdressing and rerouting should be considered. As a result, the entire design effectiveness is greatly restricted. In contrast, Fig. 2(b) shows a simultaneous consideration that avoids wiring across the obstacle, thereby providing a higher solution feasibility and quality in terms of routability and wire length.

Regarding the above discussions, it is necessary to develop an integrated design automation for pin-constrained EWOD-chip designs. Consequently, this paper proposes an efficient

heuristic approach that simultaneously considers electrode-addressing and routing to achieve high design performance and solution quality.

#### A. Previous Work

To the best knowledge of the authors, there is no existing work considering the EWOD-chip routing problem with on-chip obstacles. Most related works focus on only pin-constrained electrode-addressing techniques [13], [14], [19], [22], and only one work in [12] considers the automated routing for pin-constrained EWOD chips. The state-of-the-art work in [12] adopts a two-stage technique of global routing followed by a progressive routing scheme. In global routing, a minimum set of straight routing tracks with zero detour routes are constructed to simultaneously minimize the pin-count and wire length globally. Then, the progressive routing scheme iteratively completes the addressing and routing based on these tracks. Although this method provides a solution to automated routing for pin-constrained EWOD chips, it is based on the assumption of non-existent obstacles. If on-chip obstacles block these tracks, many routing pathways are prohibitively obstructed. Therefore, the entire routing procedure might encounter severe routability problems if we attempt to apply this method to solve the obstacle-avoiding routing. Moreover, large-scale DMFB applications with many electrodes incur the demand of multilayer wire routing. The method proposed in [12] assumes single-layer routing. It is not intuitive to directly apply this method to multilayer designs without inducing lots of via and long wire length. These impediments and insufficiencies trigger the necessity of a dedicated routing algorithms for pin-constrained EWOD chips.

#### B. Our Contributions

In this paper, we propose a novel ILP-based obstacle-avoiding routing algorithm for pin-constrained EWOD-chip designs. Compared with prior design automations, our router is the first work proposed in the literature that provides integrated electrode-addressing and routing with presence of on-chip obstacles. Our contributions can be summarized as follows.

- 1) We consider the EWOD chips with the presence of obstacles and introduce a practical problem formulation of obstacle-avoiding routing for EWOD-chip designs.
- 2) We propose the first routing algorithm to solve this practical problem in EWOD-chip designs, which can relieve the current design burden of time-consuming manual optimizations.
- 3) Our algorithm, based on effective integer-linear-programming (ILP) formulation as well as efficient routing framework, solves this problem with high routability while keeping the induced design complexity minimized.
- 4) We propose the first multilayer routing algorithm to deal with large-scale EWOD-chips. Based on the proposed partition method, our algorithm can appropriately utilize the routing resource of each layer and minimize the layer usage.

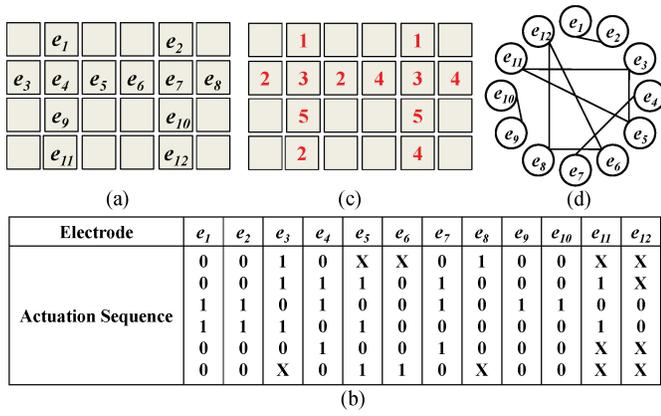


Fig. 3. (a) Electrodes used for droplet controls. (b) Control information in the form of actuation sequences. (c)–(d) Broadcast addressing result and corresponding clique-partition result in compatibility graph.

Experimental results demonstrate the effectiveness of the proposed addressing and routing algorithm. The evaluation performed on two real-life chips with obstacles and several randomly generated hard test chips show that our routing algorithm achieves high routability, whereas the extension of the previous work fails to complete any of these test cases.

The rest of this paper is organized as follows. Section II presents the electrode-addressing and routing in pin-constrained EWOD-chip designs. Section III formulates the obstacle-avoiding EWOD-chip routing problem and Sections IV and V describe the proposed algorithm to solve this practical routing problem. Sections VI and VII show the experimental results and conclude this paper, respectively.

## II. PIN-CONSTRAINED EWOD-CHIP DESIGNS

EWOD chips are typically controlled through an external controller, also referred to as function generator, which has a limited number of control-signal/pin ports. Designers working on EWOD chips should comply with a practical constraint that specifies the maximum allowable pin-count. This criterion leads to the pin-constrained EWOD-chip designs. In this section, we first discuss the prevailing electrode-addressing approach, broadcast addressing, to pin-constrained designs. Then, we discuss the routing problem with the presence of obstacles for pin-constrained EWOD chips.

### A. Broadcast Addressing

Pin-constrained design techniques have recently received much attention as they utilize a limited number of pins to control a large number of electrodes in EWOD chips. A promising solution, called broadcast addressing, has been presented in [19]. The droplet-controlling information is stored in the form of electrode actuation sequences, where each bit in a sequence represents a signal status [“1”(actuated), “0”(deactuated), or “X”(don’t-care)] of the electrode at a specific time step [19]. Examples of an electrode set and the corresponding actuation sequences are presented in Figs. 3(a) and (b). Two electrode actuation sequences are compatible, if either of the values of two bits at every time step are the same,

or the value of one bit is “X”. Specifically, we can replace the status “X” with “1” or “0” so that different actuation sequences become the identical sequence. Take electrodes  $e_3$  and  $e_5$  in Fig. 3(b) for example. By replacing “X” in the actuation sequences of  $e_3$  and  $e_5$  with one, we can obtain an identical outcome of 110101. Therefore,  $e_3$  and  $e_5$  can be addressed with the same control pin due to their mutually compatible actuation sequences. Broadcast addressing utilizes this feature to identify groups of electrodes with mutually compatible actuation sequences and assign each group a dedicated control pin. In other words, multiple electrodes in the same group share a single pin, thereby reducing the total required pin-count for electrode-addressing without affecting the bioassay operations. Researchers also model the broadcast addressing as a compatibility graph [12], [19], in which the vertex set represents the electrode set and an edge between two electrodes indicates that the corresponding actuation sequences are compatible. Therefore, the derivation of a broadcast-addressing result can be mapped to a graph problem of clique partition. Fig. 3(d) partitions the compatibility graph of Fig. 3(b) into five cliques and assigns control pins to electrodes as Fig. 3(c). With broadcast addressing, required pin-count to control a bioassay can be dramatically reduced.

### B. Obstacle-Avoiding EWOD-Chip Routing

After electrodes are addressed with control pins, conduction wires must be appropriately routed to establish the correspondence between the control pins (i.e., electrodes with the same pin must be connected with conduction wires) and the signal ports. Since signal ports of EWOD chips are generally located outside the component (i.e., defined as the 2-D electrode array) boundary, the routing problem that connects these inside terminal pins to outside signal ports is similar to the typical escape routing problem in many VLSI designs [4], [8], [21]. However, in pin-constrained EWOD-chip designs, multiple electrodes may share the same control pin, and, therefore, a single control signal may actuate multiterminal pins. To realize the electrical connections, multiterminal pins with the same control signal must be routed together, and then escape to the component boundary. This feature makes the typical escape router, which is based on the connection between two-terminal pins, unsuitable for EWOD-chip routing problem. In addition to establishing escape routing connections, modern EWOD-chip routing should address the issue of routing obstacles. To avoid signal transmission error, any likelihood of routing wires across these obstacles is prohibited. Regarding above discussions and the distinctive technology of EWOD chips from VLSI counterparts, it is desirable to develop a specialized router to handle the obstacle-avoiding EWOD-chip routing problem.

In this paper, we consider the EWOD chip built on printed-circuit-board (PCB), in which electrical conduction wires are created inexpensively using mature PCB technology [10]. We allow wires to be routed in horizontal and vertical fashions with a routing angle of 90 degree. Since conduction wires are routed beneath the electrode layer through connection vias, the EWOD-chip routing model can be specified as a 2-D pin array (Fig. 4). The routing model used in this paper is the same

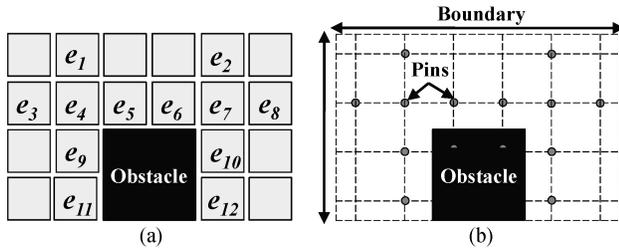


Fig. 4. Routing model. (a) Layout of DMFB. (b) Corresponding routing model of (a).

as that in [12], which is based on a uniform grid structure with a maximum number of three wires passing through adjacent pins. Besides, a multilayer arrangement of wiring connections necessitates a dedicated mechanism to route wires between layers. Accordingly, we first focus on the single-layer routing problem in Sections III and IV; and, then, a dedicated algorithm for multilayer designs is proposed in Section V.

### III. PROBLEM FORMULATION

The obstacle-avoiding routing problem for pin-constrained EWOD chips can be formulated as follows.

*Input:* A set of electrodes  $|E_e|$  used for droplet controls with the control information in the form of actuation sequences, a specified value of  $P_{max}$  indicating the maximum pin-count supported by external controller, a set of on-chip obstacles, design rules, and chip specification.

*Constraints:*

- 1) Broadcast-addressing constraint: A set of electrodes can be addressed to a single control pin if and only if their actuation sequences are mutually compatible.
- 2) Routing constraint #1: Satisfying the design rules and avoiding any likelihood of wiring through obstacles.

*Objective:* Correctly deriving an electrode-addressing result whose resulted pin-count cannot exceed the maximum allowable value  $P_{max}$  (i.e., pin constraint) and establishing a feasible routing solution.

### IV. ILP-BASED OBSTACLE-AVOIDING ROUTING ALGORITHM

The overview of our ILP-based obstacle-avoiding routing algorithm (ILP-OAR) for EWOD chips is presented in Algorithm 1. In proposed algorithm, we divide the electrodes into several groups (i.e., electrode groups). Each electrode group indicates an independent control pin. The basic idea behind our algorithm is to reduce the design complexity by dividing the entire routing problem into several manageable routing iterations. Each iteration is associated with an incrementing value  $R$ , indicating the maximum allowable grouping range of each electrode. In each iteration, we first conduct the routability-driven electrode grouping by using an effective ILP formulation (line 5). We then establish the wiring connections according to the electrode grouping result (line 6). The entire routing iterations terminate when  $R$  covers the whole chip.

#### Algorithm 1: ILP-Based Obstacle-Avoiding Routing

---

**Input** : A 2-D pin array, an electrode set  $|E_e|$ ,  $P_{max}$ , chip spec.  
 /\*  $R$ : maximum grouping range of each electrode \*/  
 /\*  $W/H$ : width/height of the chip \*/

```

1 begin
2    $R \leftarrow 0$ ;
3   while  $R \leq W + H$  do
4      $R \leftarrow R + 1$ ;
5     ILP-based routability-driven electrode grouping;
6     wire routing;
7     if resulted pin-count satisfies pin constraint then
8       break
9     end
10  end
11 end
```

---

**Output:** Arrangement of control pins and wires

---

Additionally, to derive a feasible routing solution with a minimum pin-count is undesirable in this problem because it will greatly increase computational complexity. On the other hand, to derive a feasible solution, which satisfies the pin constraint is a more desirable objective and is more practical. Therefore, we terminate our algorithm once we have found a feasible routing solution that satisfies all the constraints (lines 7–9).

In the following sections, we detail the proposed ILP-based routability-driven electrode grouping and wire routing methods.

#### A. ILP-Based Routability-Driven Electrode Grouping

In pin-constrained EWOD-chip designs, different electrode grouping results may lead to different routing solutions. An inappropriate electrode grouping result produces wiring detors, which may cause an infeasible routing solution. For example in Fig. 2, the electrode grouping result in Fig. 2(a) causes an infeasible routing solution with many detors, whereas the electrode grouping result in Fig. 2(b) leads to a feasible result. Moreover, routability also decreases in the presence of obstacles. Thus, it is necessary to incorporate the routability issue into the electrode grouping. To this end, we propose a routability-driven electrode grouping method based on ILP. The major goal is to minimize the pin-count (i.e., maximize the likelihood of pin-count reduction) to meet the pin constraint while taking routability issues into account. For each net (i.e., defined as a set of electrodes that are addressed with the same control pin), we observe that detors greatly contribute to the routability degradation. This is because detors incur a longer wiring distance, and, therefore, have a high possibility of blocking the pathway of other wires. The idea here is try to avoid unnecessary detors by utilizing the constraints of ILP formulation. We identify there are two major factors having the potential to introduce detors: net interference and obstacle crossing.

- 1) *Net interference:* Net interference describes a situation in how the routing path of a net interferes with others, particularly for the cause of detors. Nets on chip may interfere each other when conducting wire routing. A net with a longer wiring distance is more likely to block the routing paths of other wires than a shorter wiring

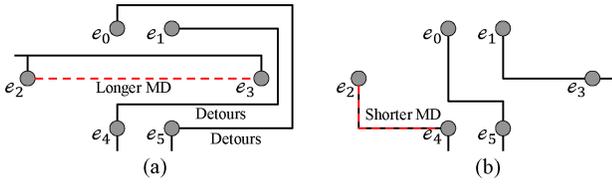


Fig. 5. Grouping result with longer Manhattan distance (MD) in (a) incurs more detours than that of shorter MD in (b).

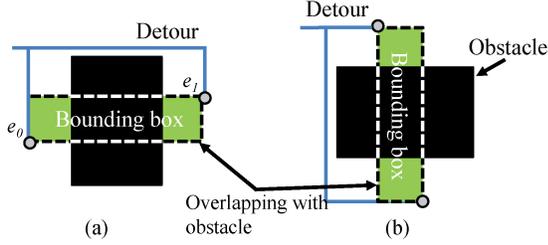


Fig. 6. Overlapping between obstacles and nets incurs detours. (a) Horizontal crossing. (b) Vertical crossing.

distance. To resolve the blocked paths, some wires might incur detours, thereby degrading the routability. In this paper, we estimate the net interference based on Manhattan distance among electrodes of nets. For example in Fig. 5(a), if we group electrode  $e_2$  and  $e_3$  together, the Manhattan distance of net  $(e_2, e_3)$  is longer than other nets. To avoid crossings, nets  $(e_0, e_5)$  and  $(e_1, e_4)$  might incur severe detours. In contrast, if we group  $e_2$  with  $e_4$  rather than  $e_3$ , we could have a shorter Manhattan distance and avoid detours when routing other nets, as illustrated in Fig. 5(b). Therefore, during electrode grouping, we manage to keep the Manhattan distance among the electrodes in each net as minimum as possible.

- 2) *Obstacle crossing*: Detours are likely to occur when the bounding box of a net overlaps with obstacles. This overlap makes wire cross obstacles if we directly conduct the shortest connection. To avoid the crossing error, wires should be detoured to avoid routing through obstacles. As illustrated in Fig. 6(a), the bounding box of net  $(e_0, e_1)$  overlaps with the obstacle horizontally, called horizontal crossing in this paper. To avoid crossing with obstacle, routing  $e_0$  and  $e_1$  must incur detours. Similarly, Fig. 6(b) shows a type of vertical crossing. These crossings induce wiring detours and consume more routing resource. Therefore, electrode grouping should prevent the bounding box of a net from horizontal/vertical crossing with obstacles.

To consider the routability issue in electrode grouping, we incorporate the above two factors into the proposed ILP formulation. There are two key strategies in this ILP formulation. The first strategy is to associate the factor net interference with a parameter  $R$  to limit the Manhattan distance among the electrodes of each net, thereby keeping that distance as minimum as possible. Based on the feature of  $R$ , we desire to obtain a feasible solution with a minimized  $R$ . Hence,  $R$

TABLE I  
NOTATIONS USED IN ILP FORMULATION

$u_i$	a 0-1 variable represents there is at least one electrode group assigned to group number $i$
$a_{ij}$	a 0-1 variable represents $p_i$ is assigned to group number $j$
$c_{ij}$	a 0-1 constant represents $p_i$ and $p_j$ are compatible (i.e., all electrodes in $p_i$ and $p_j$ are mutually compatible)
$M$	a very large constant
$NI(p_i, p_j)$	a 0-1 constant represents the Manhattan distance of any electrode pair exceeds $R$ when $p_i$ and $p_j$ are merged into one group
$OC(p_i, p_j, o_i)$	a 0-1 constant represents the bounding box of $p_j$ has horizontal/vertical crossing with obstacle $o_i$ after $p_i$ and $p_j$ are merged into one group

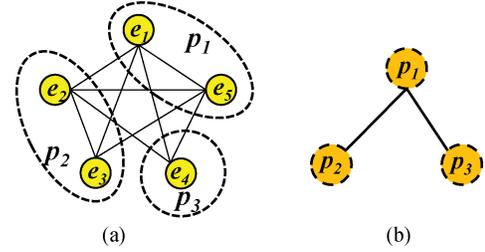


Fig. 7. Two electrode groups are compatible if and only if electrodes in both groups are mutually compatible. (a) Compatibility graph in viewpoint of electrodes. (b) Compatibility graph in viewpoint of electrode groups.

is initialized as zero and will gradually increment by one unit distance between adjacent electrodes, corresponding to each routing iteration. Limiting the maximum allowable grouping range encourages electrodes to group with other electrodes, which are nearby within  $R$ . By this strategy, the nets with long wire length could be avoided. The second strategy considers obstacle crossing by prohibiting horizontal/vertical crossing during electrode grouping. We prevent the electrodes that are in opposite direction of an obstacle from grouping together thereby avoiding detours around obstacles. The following paragraphs introduce the objective function and constraints of our ILP formulation.

- 1) *ILP Formulation*: The notations are listed in Table I. The inputs of our ILP are the set of  $n$  electrode groups  $P = \{p_1, p_2, \dots, p_n\}$  where each  $p_i \in P$  indicates an electrode group, the parameter  $R$ , the set of  $m$  obstacles  $O = \{o_1, o_2, \dots, o_m\}$ , and the compatibility graph  $G$  of electrodes. In Table I, the 0-1 constant  $c_{ij}$  is 1 if all electrodes in  $p_i$  and  $p_j$  are mutually compatible. For example in Fig. 7, electrode group  $p_1$  and  $p_2$  are compatible [Fig. 7(b)] because all electrodes (i.e.,  $e_1, e_2, e_3, e_5$ ) are mutually compatible [Fig. 7(a)]. However,  $p_2$  and  $p_3$  are incompatible since there exists an electrode pair  $(e_3, e_4)$  that is not compatible. The term  $c_{ij}$  can be used to indicate the compatibility of the current electrode groups. Based on the notations above, we presents the following formulations.

Objective function

$$\text{Minimize} : \sum_{i=1}^n u_i \quad (1)$$

subject to

$$\sum_{j=1}^n a_{ij} = 1, 1 \leq i \leq n \quad (2)$$

$$\sum_{i=1}^n a_{ij} \leq M \cdot u_j, 1 \leq j \leq n \quad (3)$$

$$a_{ik} + a_{jk} \leq 1 + c_{ij}, 1 \leq i, j, k \leq n \quad (4)$$

$$a_{ik} + a_{jk} \leq 2 - NI(p_i, p_j), \forall p_i, p_j \in P, 1 \leq i, j, k \leq n \quad (5)$$

$$a_{ik} + a_{jk} \leq 2 - OC(p_i, p_j, o_l)$$

$$\forall p_i, p_j \in P, \forall o_l \in O, 1 \leq i, j, k \leq n. \quad (6)$$

The objective of this ILP formulation is to minimize the pin-count, thereby facilitating the pin-count reduction. Note that  $n$  represents the pin-count (i.e., the number of electrode groups) at the beginning of each routing iteration, and is initialized as  $|E_e|$  in the first routing iteration. We have four major constraints in our ILP formulation.

- 1) *Grouping constraints*: We merge the electrode groups by assigning each  $p_i \in P$  with a group number (1 to  $n$ ). Electrode groups are expected to be merged into one group if they are assigned with the same number. Constraint (2) guarantees that each electrode group can only be assigned with a specific group number. Constraint (3) is used to maintain the 0-1 variable  $u_i$ .  $u_i$  is 1 if there exists at least one electrode group assigned with group number  $i$ . Otherwise, it is 0 (i.e., the group number  $i$  is not used).
- 2) *Broadcast constraints*: Constraint (4) states that for each pair of electrode group ( $p_i, p_j$ ), if  $p_i$  and  $p_j$  are assigned with the same group number  $k$ , electrodes in  $p_i$  and  $p_j$  should be mutually compatible.
- 3) *Net interference*: As mentioned before, we use the parameter  $R$  to limit the maximum grouping range of each electrode. Each electrode can only be grouped with electrodes within  $R$  distance (one unit represents the distance between adjacent electrodes), which is formulated as (5).
- 4) *Obstacle crossing*: The detours caused by horizontal crossing or vertical crossing should also be minimized. Constraint (6) achieves this goal. For each pair of electrode groups ( $p_i, p_j$ ),  $p_i$  and  $p_j$  cannot be merged together if the resulting group causes horizontal/vertical crossing.

2) *ILP Reduction*: Although the formulation above is simple, it will incur redundant solution space. Redundant solution space is caused from identical group combinations. Take Fig. 8 as an example. Suppose we currently have three electrode groups,  $p_1, p_2$ , and  $p_3$ . Only  $p_1$  and  $p_2$  are mutually compatible. When we solve ILP, we assign these groups with group numbers 1 to 3 and then minimize the number of used group numbers. Obviously, the assignment in Fig. 8(b) is an optimal assignment (i.e.,  $p_1$  and  $p_2$  are grouped together) so that the pin-count can be minimized. However, the solutions (a), (c), (d), (e), (f) in Fig. 8 are identical to Fig. 8(b) because they both turn out to be the same two group combinations,

Fig. 8. (a)–(f) Six identical pin assignment solutions.

Fig. 9. Illustration of ILP reduction using the proposed technique. Without proposed technique, all nine variables can be 0 or 1. After reduction, only four variables need to be decided.

$\{\{p_1, p_2\}, \{p_3\}\}$ . These redundant solutions increase the solving time for ILP to find the optimal solution, however, we have to keep only one of them.

Fortunately, we can prune such redundant solutions without affecting the quality of ILP result. By observing above formulation, we find that there are redundant solutions above the main diagonal of matrix  $a$ .  $p_1$  can choose group number 1, 2, or 3, causing redundant solutions. Therefore, we use (7) to eliminate those solutions above the main diagonal. Moreover, (8) is added to restrict the group number that each electrode group can be assigned with. Constraint (8) states that electrode group  $p_i$  can be assigned with group number  $j$  if the electrodes in electrode groups  $p_i$  and  $p_j$  are mutually compatible. With these constraints, the situations in Fig. 8(a), (c), (d), (e), and (f) will not exist. Considerable unnecessary solutions have been eliminated in original formulation. The redundant type of Fig. 8(c), (d), (e), and (f) can be eliminated by (7) while the redundant type of Fig. 8(a), (c), (e), and (f) can be eliminated by (8). Fig. 9 depicts the solution space reduction using the proposed technique

$$a_{ij} = 0, i < j \quad (7)$$

$$a_{ij} \leq c_{ij}, 1 \leq i, j \leq n. \quad (8)$$

3) *ILP Analysis*: In order to show the scale of ILP model, we provide a complexity analysis of constraint size corresponding to the number of electrodes in Table II. All constraints are bounded by the number of electrodes and obstacles (number of electrodes equals to  $N$ , number of obstacles equals to  $O$ ). Note that the big O notation of total variables is bound to  $N^3$ .

TABLE II  
COMPLEXITY ANALYSIS IN ILP FORMULATION

ILP Constraints	# of constraints	BigO Notation
(2) $\sum_{j=1}^n a_{ij} = 1, 1 \leq i \leq n$	$N^2$	$O(N^2)$
(3) $\sum_{j=1}^n a_{ij} \leq M \cdot u_p, 1 \leq j \leq n$	$N^1$	$O(N^1)$
(4) $a_{ij} + a_{jk} = 1 + c_{ij}, 1 \leq i, j, k \leq n$	$N^3$	$O(N^3)$
(5) $a_{ij} + a_{jk} \leq 2 - NI(p_i, p_j),$ $\forall p_i, p_j \in P, 1 \leq i, j, k \leq n$	$N^3$	$O(N^3)$
(6) $a_{ij} + a_{jk} \leq 2 - OC(p_i, p_j, o),$ $\forall p_i, p_j \in P, \forall o_i \in O, 1 \leq i, j, k \leq n$	$O \cdot N^3$	$O(O \cdot N^3)$
(7) $a_{ij} = 0, i < j$	$\frac{N \cdot (N - 1)}{2}$	$O(N^2)$
(8) $a_{ij} \leq c_{ij}, 1 \leq i, j \leq n$	$N^2$	$O(N^2)$
Total # of constraints = $\frac{1}{2}N + \frac{2}{5}N^2 + (O + 2) \cdot N^3$		$O(N^3)$

### B. Wire Routing

After solving the ILP formulation in a routing iteration, we have an electrode grouping result with respect to the maximum allowable grouping distance  $R$ . We could also analyze the result and obtain the net information of these electrode groups, denoted as  $N$ . Each net in  $N$  is either a two-terminal net or a multiterminal net, and each terminal indicates an electrode group in previous routing iteration. Note that in the ILP result, the electrode groups assigned with the same group number are expected to be merged into one electrode group. The goal here is to establish wiring connections for nets in  $N$ . Our routing procedure is presented in Algorithm 2.

First, we calculate the passing count of each net by counting the number of wires that pass through the bounding box of a net. The wires to be counted include the routed wires and trial routing wires (discussed later) in previous routing iteration. Passing count can assess the wire congestion inside the bounding box of a net. Then, we iteratively pick up the smallest passing count net (i.e., the most sparse net) from  $N$ , denoted as  $n$ , and route it (lines 4–18). The wiring connection of each net can be electrode-to-wire, electrode-to-electrode, or wire-to-wire path, which connects electrode groups and merges them into one group. The paths can be quickly found by maze routing. In case of an infeasible route, we neglect the routing of failed net (lines 6–8). In addition to connecting electrodes, we have to ensure the connections to the chip boundary. Consequently, when  $n$  is routed, we adopt a trial routing to check if there exists a feasible escape route for each electrode group (line 10). There exists many trial routing algorithms in VLSI routing technology [4]. In this paper, we use the network-flow based algorithm, which is typically used for escape routing, as our trial routing method [20] (Section IV-C). A successful trial routing indicates the routing path of  $n$  is permissible, whereas an unsuccessful trial routing reveals that the routing path of  $n$  blocks that of other electrode groups. Therefore, for an unsuccessful trial routing, we neglect the routing of the net  $n$ . To avoid duplicate routing path in the afterward routing iteration, we use a history-based technique by assigning the grids along the failed routing path a penalty (line 13). Finally, we construct the escape routing for each

### Algorithm 2: Wire routing

---

```

Input : Electrode grouping information from ILP solution
/*  $N$ : net set obtained from ILP solution */
1 begin
2   while  $N \neq \phi$  do
3      $n \leftarrow$  pop the smallest passing count net from  $N$ ;
4     Route  $n$ ;
5      $path \leftarrow$  routing path of  $n$ ;
6     if Failure route then
7       drop  $path$ ;
8       continue;
9     end
10    conduct the trial routing;
11    if unsuccessful trial route then
12      drop  $path$ ;
13      assign the  $path$  as high penalty;
14      continue;
15    end
16    if resulting pin-count satisfies pin constraint then
17      break;
18    end
19  end
20  establish escape routing for each electrode group;
21  return routing result;
22 end

```

---

electrode group including those groups containing only one electrode (line 20).

The entire routing procedure runs once for each routing iteration (i.e., the iteration under the specified  $R$ ). Then, we check whether the resulting pin-count can satisfy the pin constraint. If the pin constraint is not met, we increment  $R$  by one and proceed to next routing iteration. On the other hand, the routing procedure terminates. Note that in the case of a failed route that cannot satisfy the pin constraint when entire algorithm ends, designers could resort to another controller with a higher pin-count specification.

### C. Trial Routing

Trial routing checks that if there exists a feasible escape route for each electrode group. In this paper, we use a maximum-flow based method to conduct trial routing [20]. If the resulting flow value is equal to the pin-count, a feasible escape route exists, and vice versa. However, as mentioned in Section II-B, multiterminal nets in this problem make it difficult to directly apply the typical escape router to our problem. Thus, modification is required to typical escape router. To deal with this issue and ensure the correctness of flow network, we construct a maximum-flow graph by the following steps.

- 1) For each routing grid and pin grid (Fig. 10) of the chip except for the grid with presence of obstacle, create a node with capacity one.
- 2) Create a dummy source node and a dummy sink node.
- 3) For each routing/pin grid node  $v_f$ , create directed edges with capacity 1 from  $v_f$  to its neighbor routing grid nodes  $v_t$  if one of the following holds:
  - a)  $v_f$  and  $v_t$  are both occupied by the same wire.
  - b)  $v_t$  is not routed by any wire (i.e., empty grid).
- 4) For each pin grid  $v_p$  that is responsible to flow (explain later), create a directed edge with capacity 1 from source node to  $v_p$ .

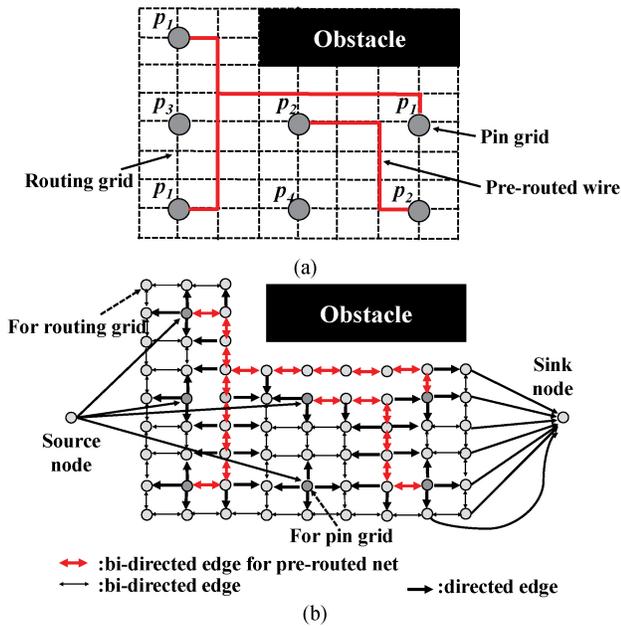


Fig. 10. Illustration of (a) wire routing and (b) corresponding network graph. Intersections of dashed lines in (a) indicate routing grids or pin grids.

- 5) For each routing grid node  $v_b$  on the boundary of chip, create a directed edge with capacity one from  $v_b$  to sink node.

Fig. 10 shows an example of maximum-flow graph construction. As shown in Fig. 10(a), there are two prerouted nets, one obstacle, and four electrode groups to be escaped (i.e.,  $p_1, p_2, p_3, p_4$ ). In Fig. 10(b), we construct four edges from source node to pin nodes. Note that for the electrode group with multiterminal pins, we only need one unit of flow to decide the escaping routing path. As a result, from source node, we only construct one edge to one of pin nodes, which belongs to this electrode group, and this pin node then is responsible to the flow of associated control pin. In order not to introduce wiring violations, flow cannot go through different wires. Consequently, there is no edge connection between different wires. The flows eventually arrive at the sink node through the boundary node  $v_b$  (we omit several edges for simpler illustration). Based on the above rules, we can obtain a feasible escape routing result if the maximum flow value is equal to the desired pin-count (i.e., four in this example). Otherwise, corresponding electrode grouping and wire routing is infeasible because there is no escape routing solution.

#### D. Exemplification

In this subsection, we use an example to illustrate our algorithm (Fig. 11). The index aside each pin in Fig. 11 represents the pin number assigned to corresponding electrode. Suppose  $P_{max}$  is 16 and there are 24 electrodes on the chip.

We begin our algorithm with  $R = 1$  and each electrode cannot be grouped with others under this condition. Thus, each electrode is addressed with a dedicated pin. Next, we conduct escape routing (i.e., trial routing) for each electrode group (i.e., each electrode forms its own group) to see whether the current result is available. The routing result obtained in first iteration

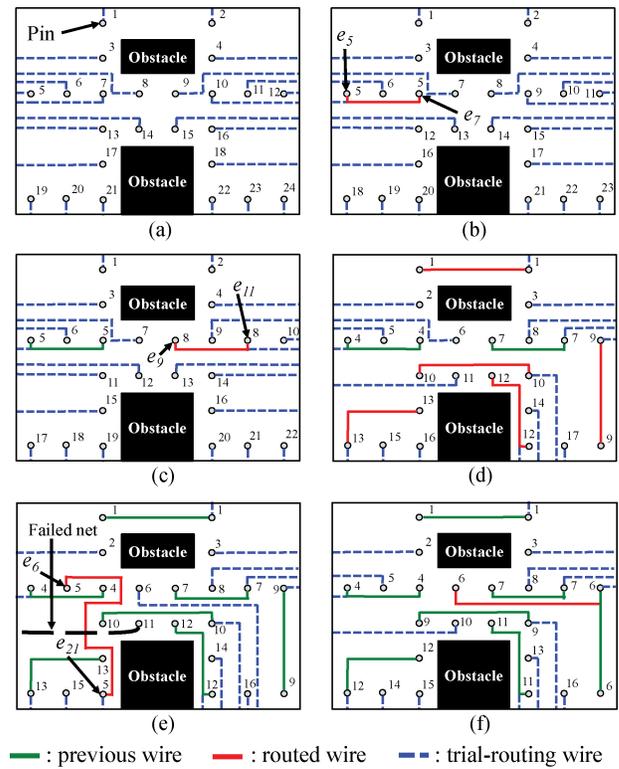


Fig. 11. Exemplification of our algorithm. Index aside each pin indicates pin number assigned to electrode. (a) Routing result after iteration  $R = 1$ . (b)–(c) Based on the electrode grouping result of ILP, we route nets  $(e_5, e_7)$ ,  $(e_9, e_{11})$  and obtain successful trial routing results sequentially. (d) Routing result after iteration  $R = 3$ . (e) Rip-up blocking net when trial routing fails. (f) Final result after iteration  $R = 4$ .

is illustrated in Fig. 11(a). We then increment  $R$  by one and the entire routing procedure proceeds to the second routing iteration. After solving the ILP, electrodes  $e_9, e_{11}$  as well as  $e_5, e_7$  are grouped into two electrode groups, respectively. We obtain a new grouping result with two nets  $\{(e_9, e_{11}), (e_5, e_7)\}$  and sort the nets by their passing counts described in previous section. We route the smaller passing count net  $(e_5, e_7)$  first by maze routing, followed by conducting trial routing to ensure all the paths of each electrode group can escape to the boundary of chip, as illustrated in Fig. 11(b). It can be observed that the escape routing in Fig. 11(b) is successful. Hence,  $e_5$  and  $e_7$  can be grouped together, reducing the pin-count by 1. We maintain the wire of  $(e_5, e_7)$  in present result and route the subsequent net of  $(e_9, e_{11})$  in the similar manner [Fig. 11(c)]. Our wire routing for this iteration is completed for that there is no untried net in  $N$ . Since required pin-count is now 22 and  $P_{max}$  is 16, we proceed to the next iteration with  $R = 3$ . By adopting the same routine [Fig. 11(d) and (f)] until pin constraint is met, a feasible solution with desirable pin-count is obtained in Fig. 11(f).

For the case of failure route (i.e., net-route or trial route) in Fig. 11(e), where the wire of routing  $(e_6, e_{21})$  obstructs the pathway of pin 11, we rip-up this failure net. We then neglect this net in this iteration. To avoid duplicate routing path in subsequent routing iterations, the original routing path of  $e_6$  and  $e_{21}$  is assigned penalty. Finally, we have the entire arrangement of wires and pins [Fig. 11(f)].

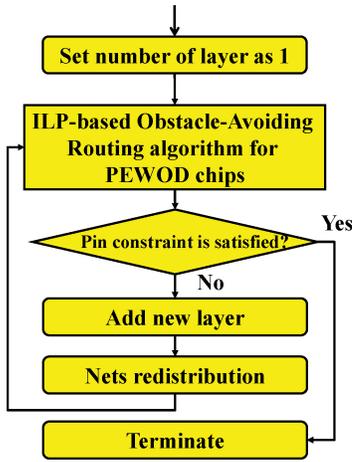


Fig. 12. Algorithm flow for multilayer EWOD chip designs. PEWOD denotes pin-constrained EWOD chips.

V. ALGORITHM FOR MULTILAYER DESIGNS

Using single routing layer is inadequate for high electrode density chips. Control pins may not be able to escape from the chip even with the simplest addressing method (i.e., direct addressing). In this condition, a multilayer architecture may be necessary to establish the electrical connection. In this section, we propose the routing algorithm for multilayer EWOD chip designs. To reduce the via overhead, which increases the fabrication cost, we only allow each net to be routed within one layer in this paper. We first give the problem formulation for multilayer EWOD designs and, then, the proposed algorithm will be described.

A. Problem Formulation

The input and constraints are similar to the problem formulation in Section III. With the multilayer architecture, we have additional constraints and a new objective.

Constraints:

- 1) Routing constraint #2: Each net (electrode group) should be routed within one layer to avoid the insertion of via.

**Objective:** Correctly deriving an electrode-addressing result whose pin-count cannot exceed the maximum allowable value  $P_{max}$  and establishing a feasible routing solution with minimized routing layer usage.

B. Algorithm Flow

Fig. 12 demonstrates the proposed algorithm flow for multilayer EWOD chip designs. In order to minimize the number of used layers, we start our algorithm by setting the number of layers as one. The ILP-OAR algorithm proposed in Section IV is then applied. We will check whether the pin constraint is satisfied when ILP-OAR algorithm ends. If it is not satisfied and the pin-count is still reducible (i.e., there exist two electrode groups that are compatible), an extra layer is added. With additional routing layer, we have more routing space to route the wires that failed previously because of congested wire density. Therefore, nets redistribution step, is conducted to adjust and move wires in original layers. Nets

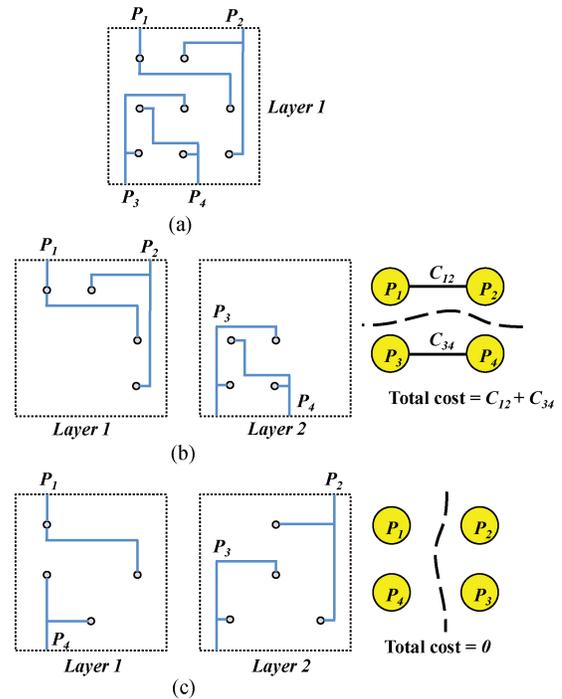


Fig. 13. Different net distribution results and their corresponding graphs: (a) The original routing result. Outcome of (c) has better congestion and wire length than that of (b).

redistribution includes two phases, electrode group partition and wire adjustment. In the first phase, a set of electrode groups are collected from current electrode grouping result and moved to the new layer. Hence, a FM-based partition algorithm is introduced to partition the electrode groups into two subsets: 1) a subset to stay in the original routing layer and 2) a subset moved to the new layer. After deciding which layer each electrode group should belong to, the second phase rips-ups and reroutes the wires to the new layer. Note that the wires in the original layer can be refined to shorten the wire length and release more routing resource because of the layer change of electrode groups. Following net redistribution, the next iteration can start by again applying ILP-OAR algorithm. We apply ILP-OAR algorithm to continue merge electrode groups to reduce pin-count. To conduct electrode grouping (similar to Section IV-A1), we solve the electrode grouping layer by layer because groups in different layers cannot be grouped together. Similarly, we iteratively conduct the algorithm flow and terminate the entire algorithm when the pin constraint is satisfied. Finally, the routing result with minimized layer usage can be obtained. In following subsection, we will detail the step of nets redistribution.

C. Nets Redistribution

After adding a new routing layer, the next major goal is to decide which electrode groups should be moved to the new layer. To utilize the routing resources evenly, we would like to evenly distribute the electrode groups among existing routing layers. Therefore, we partition the existing electrode groups into two subsets. The first one is the subset that electrode groups in this subset will stay in their original routing layer;

TABLE III  
OVERALL COMPARISON BETWEEN [12]-EXTENSION, BASELINE, AND OUR ALGORITHM

Chip	$ E_e $	Size	O(%)	$P_{max}$	[12]-extension				Baseline				Ours			
					#Pin	WL	#Fail	CPU	#Pin	WL	#Fail	CPU	#Pin	WL	#Fail	CPU
DNA-1	211	$16 \times 24$	13%	128	-	-	31	-	-	-	24	-	128	2819	0	2.7
DNA-2	77	$13 \times 21$	8.7%	32	-	-	6	-	-	-	2	-	32	1179	0	0.7
random-1	24	$6 \times 8$	12.5%	16	-	-	8	-	16	300	0	0.0	16	271	0	< 0.1
random-2	59	$15 \times 15$	16.8%	32	-	-	5	-	32	1256	0	0.4	32	1039	0	0.22
random-3	62	$15 \times 15$	13.7%	32	-	-	18	-	32	1571	0	0.8	32	1050	0	0.17
random-4	91	$15 \times 15$	12.4%	64	-	-	15	-	64	1650	0	1.0	64	1362	0	0.4
random-5	256	$20 \times 30$	15.0%	128	-	-	34	-	-	-	48	-	128	4196	0	4.16
random-6	400	$30 \times 40$	18.3%	256	-	-	71	-	-	-	22	-	256	7411	0	5.22
Total					188				96				0			

and electrode groups in another subset are expected to be moved to the new routing layer. However, due to the Routing Constraint #2, the electrode groups in different layers cannot be merged (or it will incur vias to make signal transmission between different layers). The compatibility of all groups could be degraded by (2). Also, net distribution is conducted after whole iteration of ILP part and there might be some electrode groups that were obstructed by other wires in the original layer and can be merged after moving them to a new layer. Thus, maximize the compatibility of the groups after partitioning is desirable. We hope that all the groups in the same layer are more likely to be compatible, which means the groups can be further merged and the total pin-count can be further reduced. As a result, we construct a graph  $G_p$ , where each node represents existing electrode group and each edge indicates the adjacent electrode groups are mutually compatible. The major goal of partition is to minimize the compatibility degradation of compatibility graph after partition. We use and modify the well-known partition algorithm proposed by Fiduccia and Mattheyses (FM) [9], to meet this demand. Specifically, we map the objective minimize the degradation of compatibility graph in electrode group partition to minimize the total cut in typical graph partition problem. The ratio between two subsets can be adjusted according to the number of layers.

In addition to the considering the compatibility, we also want to minimize the affects of net interference. Consider the example in Fig. 13(a), we want to distribute four nets  $P_1$ – $P_4$  to two routing layers. With additional routing layer, we can dispatch these groups to different layers without loss of compatibility. We list two different distribution result Figs. 13(b) and (c). A better distribution result is shown in Fig. 13(c), which has less net interference after partitioning. Regarding this, an intracost by calculating the overlapped area between the bounding box of two nets,  $C_{ij}$ , is assigned if these electrode groups are arranged in the same layer. For example in 13(b), it has intracost due to the overlapping of  $P_1$  and  $P_2$  in layer 1 and overlapping of  $P_3$  and  $P_4$ . Corresponding graphs are also shown in Fig. 13. After FM partition, we then iteratively swap pairs of nodes between two partitions as long as the total intracost can be reduced and the cut of  $G_p$  does not increase. By this manner, the routability after addition of layer is also concerned.

After electrode group partition, the second phase, wire adjustment, is then executed. According to the result of partition, we rip-up the nets, which are classified into the new added layer, and then route them in the new layer. For those nets that ought to stay in their original layer, we then modify the wires by smoothing the wires or rerouting because some wires are moved away.

The advantages of proposed technique are twofold. First, because we move some electrode groups and nets from original layers to new layers, the wire congestion in original layers are decreased. This implies some nets that are failed in previous iterations now have chances to be successfully routed in original layers after redistribution; and then the pin-count can be further reduced. Second, due to the movements of electrode groups and wires, some refinement can be conducted to wipe out detors and shorten wire length, improving subsequent routing.

## VI. EXPERIMENTAL RESULTS

We implement the proposed algorithm in C++ language on a 2.63-GHz 64-bit Linux machine with 32GB memory, and CPLEX [3] is used as our ILP solver. We evaluate our routing algorithm on two real-life EWOD chips for DNA sample preparation [1], [2]. In the first chip of DNA sample preparation (denoted as DNA-1), there are four on-chip obstacles of permanently embedded electrophoresis devices for particle separation. In the second chip of DNA sample preparation (denoted as DNA-2), there are two on-chip obstacles of permanently embedded magnet for washing protocols (i.e., eluting non-necessary particles for DNA purification). To demonstrate the robustness and scalability of our algorithm, we simulate the droplet behaviors and randomly generate six hard test chips with obstacles. To prevent trivially routable case, all random cases should follow these requirements:

- 1) number of blockages are between three to five to prevent concentrated or scattered obstacles;
- 2) the density of obstacle area is about 10% to 20% to simulate the environment of real case.

Table III shows the statistics of all test chips. “ $|E|$ ” denotes the number of electrodes, Size denotes the chip size, “O(%)” denotes the percentage of obstacle occupation, and “ $P_{max}$ ” denotes the maximum allowable number of control pins.

For comparison purpose, we implement two routing methods. The first method is the extension of [12], namely [12]-extension. We modify routing graph used in [12] by further considering the obstacles, [12]-extension can be implemented by the following steps. In first phase of [12], we construct global routing track as [12] do. Then, if any global routing track is obstructed by obstacles, we use breadth-first search (BFS) based approach to route the wires without crossing obstacles. In second phase of [12], we conduct progressive addressing and wire routing that considers obstacles in the cost function. Also, the wire cannot cross the obstacles during A\* search maze routing. The second method is to group electrodes and route wires separately, namely Baseline. In Baseline manner, we iteratively and pairwise group electrodes until pin constraint is satisfied. Then wire routing is conducted by maze routing to establish the correspondence between the control pins and signal ports.

Table III lists overall comparison results. #Pin denotes the used number of control pins, WL denotes the total wirelength computed by the number of routing grids, #Fail denotes the number of failed electrodes (unable to be routed), and CPU denotes the runtime measured by seconds. Our algorithm achieves better routability by completing all 8 test cases (100.0%), while the [12]-extension and the Baseline complete 0 (0.0%) and 4 (50.0%) test cases, respectively. The major reason our algorithm outperforms the [12]-extension is that the routability of [12] greatly relies on constructed routing tracks, which are oriented in the whole routing region without any space restriction. However, we find that several obstacles block these tracks, and thus obstruct lots of routing pathways, causing a significant decrease in routability. On the other hand, the major reason that our algorithm outperforms the Baseline is that our algorithm considers electrode grouping and wire routing simultaneously. The Baseline bears the burden of the gap between electrode grouping and wire routing although it does not rely on tracks of [12], thereby restraining the routability.

#### A. Multilayer EWOD Chips

The second experimental set evaluates our algorithm proposed in Section V for multilayer EWOD chip designs. To investigate the number of layers our algorithm uses given different pin constraints, we set the pin constraint  $P_{max}$  as 1/2, 1/3, 1/4 of  $|E_e|$ . This means we must reduce the pin-count by 50%, 67%, 75% of  $|E_e|$ , respectively.  $|E_e|$  represents original pin-count if the electrode-addressing method of direct addressing is applied. We evaluate our algorithm by giving various pin constraints on these test cases. Table IV lists the experimental results. #Layer in column 5 denotes the number of layers used in the proposed algorithm. The results indicate that our algorithm can complete most of test cases with few layer usage as well as no wiring violation. For example, in the cases of DNA-1 and random-6, proposed algorithm can reduce 75% of pin-count within three routing layers. This justifies the effectiveness of our algorithm. Table IV also shows that in some situations, our algorithm fails to meet the pin constraint due to very low pin-count demand. For these cases, we still report the minimum pin-count our method can achieve (remarked as \* in

TABLE IV  
EXPERIMENTAL RESULT OF MULTILAYER EWOD CHIP DESIGNS

Chip	$ E_e $	$P_{max}$	Ours		
			WL	#Layer	CPU
DNA-1	211	106	3215	2	3.27
		71	3606	2	4.05
		53	4900	3	11.07
DNA-2	77	39	1124	1	0.31
		26	1569	2	2.67
		20	-	-	-
		22*	1800	2	2.17
random-1	24	12*	353	2	< 0.1
		8	-	-	-
random-2	59	30	1048	1	0.25
		20	1497	2	1.24
		15	1535	2	1.26
random-3	62	31	1026	1	0.19
		21	1086	2	1.60
		16	1703	2	1.72
random-4	91	46	1493	1	0.75
		31	1650	2	1.48
		23	1910	3	3.67
random-5	256	128	4196	1	4.16
		86	4533	2	7.71
		64	4931	2	8.07
random-6	400	200	7480	1	6.55
		134	8407	2	10.24
		100	9188	2	14.59

column 4). Overall, our algorithm can complete these designs with low pin-count and minimal layer usage.

#### B. Runtime Analysis

In addition to the experiment for multilayer designs, we evaluate the effectiveness of ILP reduction in the third experiments. Table V compares the result with and without ILP reduction, which are denoted as Basic ILP and Reduced ILP, respectively. For the purpose to compare precisely, we report the ILP solving time and iterations for ILP solving (denoted as #I) in this experiment. The result shows that the Basic ILP needs more than one day to solve most of chips whereas Reduced ILP needs only few seconds. Also, the number of iterations for Basic ILP is ten times more than Reduced ILP. Moreover, the solution quality will not be restricted since we only prune the redundant solutions from the basic ILP formulation. This experiment shows the effectiveness of proposed technique.

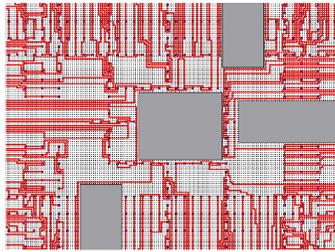
In summary, the experimental results demonstrate the effectiveness of the proposed routing algorithm in solving the obstacle-avoiding routing problem for pin-constrained EWOD chips. Fig. 14 demonstrates the routing result of random-6 with pin constraint  $P_{max} = 200$  and 100.

## VII. CONCLUSION

In this paper, we introduced a practical problem for EWOD-chip routing with on-chip obstacles. We presented the first routing algorithm with obstacle avoidance to deal with this

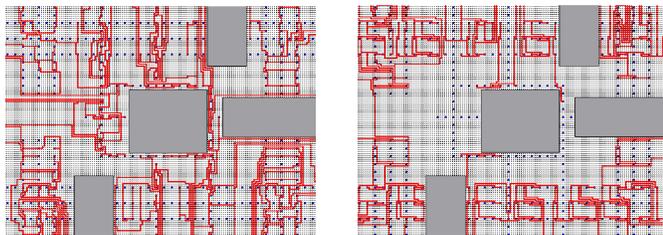
TABLE V  
COMPARISON OF RUNTIME WITH AND WITHOUT ILP REDUCTION

Chip	Basic ILP (s)		Reduced ILP (s)	
	CPU	#I	CPU	#I
DNA-1	> 1 day	> 1000000	2.0	12880
DNA-2	> 1 day	87272	0.5	489
random-1	6.59	274	< 0.1	10
random-2	> 1 day	71247	0.06	1718
random-3	> 1 day	20721	0.13	944
random-4	> 1 day	> 1000000	0.37	431
random-5	> 1 day	> 1000000	3.10	8708
random-6	> 1 day	> 1000000	4.24	1117



Layer 1

(a)



Layer 1

Layer 2

(b)

Fig. 14. Routing results of random-6 with  $P_{max}$  = (a) 200 and (b) 100, which use single layer and two layers, respectively.

design problem. Our algorithm, based on effective ILP formulation as well as efficient routing framework, can achieve high routability with low design complexity. The proposed algorithm can be utilized on both single layer and multilayer routing architecture. Two real-life EWOD chips used for DNA sample preparation and a set of self-generated test chips were used to evaluate the effectiveness of our routing algorithm for EWOD chips with the presence of obstacles.

## REFERENCES

- [1] Advanced Liquid Logic, Inc. [Online]. Available: <http://www.liquid-logic.com/>
- [2] Digital Microfluidics, Duke University [Online]. Available: <http://microfluidics.ee.duke.edu/>
- [3] CPLEX optimizer [Online]. Available: <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>
- [4] C. J. Alpert, D. P. Mehta, and S. S. Sapatnekar, *Handbook of Algorithms for Physical Design Automation*. Boca Raton, FL, USA: CRC Press, 2009.
- [5] K. Chakrabarty, "Toward fault-tolerant digital microfluidic lab-on-chip: Defects, fault modeling, testing, and reconfiguration," in *Proc. IEEE ICBCS*, Nov. 2008, pp. 329–332.
- [6] R. B. Fair, "Digital microfluidics: Is a true lab-on-a-chip possible," *Microfluidics Nanofluidics*, vol. 3, pp. 245–281, Mar. 2007.
- [7] R. B. Fair, A. Khlystov, T. D. Tailor, V. Ivanov, R. D. Evans, P. B. Griffin, S. Vijay, V. K. Pamula, M. G. Pollack, and J. Zhou, "Chemical and biological applications of digital-microfluidic devices," *IEEE Design Test*, vol. 24, no. 1, pp. 20–24, Jan.–Feb. 2007.
- [8] J.-W. Fang, I.-J. Lin, Y.-W. Chang, and J.-H. Wang, "A network-flow-based RDL routing algorithm for flip-chip design," *IEEE Trans. Computer-Aided Design*, vol. 26, no. 8, pp. 1417–1429, Aug. 2007.
- [9] C. M. Fiduccia and R.M. Mattheyses, "A linear-time heuristic for improving network partitions," in *Proc. ACM/IEEE DAC*, Jun. 1982, pp. 175–181.
- [10] J. Gong and C. J. Kim, "Direct-referencing two-dimensional-array digital microfluidics using multilayer printed circuit board," *IEEE J. Microelectromech. Syst.*, vol. 17, no. 2, pp. 257–264, Apr. 2008.
- [11] T.-Y. Ho, J. Zeng, and K. Chakrabarty, "Digital microfluidic biochips: A vision for functional diversity and more than Moore," in *Proc. IEEE/ACM ICCAD*, Jul. 2010, pp. 578–585.
- [12] T.-W. Huang, S.-Y. Yeh, and T.-Y. Ho, "A network-flow based pin-count aware routing algorithm for broadcast electrode-addressing EWOD chips," in *Proc. IEEE/ACM ICCAD*, Nov. 2010, pp. 425–431.
- [13] T.-W. Huang and T.-Y. Ho, "A two-stage ILP-based droplet routing algorithm for pin-constrained digital microfluidic biochips," in *Proc. ACM ISPD*, 2010, pp. 201–208.
- [14] C. C.-Y. Lin and Y.-W. Chang, "ILP-based pin-count aware design methodology for microfluidic biochips," in *Proc. ACM/IEEE DAC*, Jul. 2009, pp. 258–263.
- [15] Y.-Y. Lin, R. D. Evans, E. Welch, B.N. Hsu, A. C. Madison, and R. B. Fair, "Low voltage electrowetting-on-dielectric platform using multilayer insulators," *Sensors Actuators B, Chem.*, vol. 150, no. 1, pp. 465–470, 2010.
- [16] M. G. Pollack, A. D. Shenderov, and R. B. Fair, "Electrowetting-based actuation of droplets for integrated microfluidics," *Lab Chip*, vol. 2, no. 2, pp. 96–101, May 2002.
- [17] J. H. Song, R. Evans, Y. Y. Lin, B. N. Hsu, and R. B. Fair, "A scaling model for electrowetting-on-dielectric microfluidic actuators," *Microfluidics Nanofluidics*, vol. 7, no. 1, pp. 75–89, Jul. 2009.
- [18] F. Su, K. Chakrabarty, and R. B. Fair, "Microfluidics based biochips: Technology issues, implementation platforms, and design-automation challenges," *IEEE Trans. Computer-Aided Design*, vol. 25, no. 2, pp. 211–223, Feb. 2006.
- [19] T. Xu and K. Chakrabarty, "Broadcast electrode-addressing for pin-constrained multifunctional digital microfluidic biochips," in *Proc. ACM/IEEE DAC*, Jun. 2008, pp. 173–178.
- [20] T. Yan and M. D. F. Wong, "A correct network flow model for escape routing," in *Proc. ACM/IEEE DAC*, Jul. 2009, pp. 332–335.
- [21] T. Yan and M. D. F. Wong, "Optimal simultaneous pin assignment and escape routing for dense PCBs," in *Proc. ACM/IEEE ASP-DAC*, Jan. 2010, pp. 275–280.
- [22] Y. Zhao and K. Chakrabarty, "Co-optimization of droplet routing and pin assignment in disposable digital microfluidic biochips," in *Proc. ACM ISPD*, 2011, pp. 69–76.



**Jia-Weng Chang** received the B.S. and M.S. degrees in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2010 and 2012, respectively.

His current research interests include design automation on digital microfluidic biochips.



**Sheng-Han Yeh** received the B.S. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2011. He is currently a graduate student with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan.

His current research interests include physical design and design automation on digital microfluidic biochips.



**Tsung-Wei Huang** received the B.S. and M.S. degrees in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2010 and 2011, respectively. He is currently pursuing the Ph.D. degree at the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA, with Microelectronics and Computer Development Fellowships.

His current research interests include design automation for reconfigurable architecture such as digital microfluidic biochips and FPGAs.

Mr. Huang received the First Prize from the ACM SIGDA Student Research Competition (SRC), awarded at ACM/IEEE DAC in 2011. He was also the Second Prize Winner in ACM SRC Grand Final, awarded at ACM Annual Award Banquet, associated with the Turing Award Ceremony in 2011.



**Tsung-Yi Ho** (M'08–SM'12) received the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2005.

Since 2007, he has been with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, where he is currently an Associate Professor. He has published several papers in top journals and conferences, such as IEEE TCAD, ACM TODAES, ACM/IEEE DAC, IEEE/ACM ICCAD, and ACM ISPD. His current research interests include design

automation for microfluidic biochips and nanometer integrated circuits.

Dr. Ho was a recipient of many research awards, such as the Dr. Wu Ta-You Memorial Award of the National Science Council (NSC) of Taiwan (the most prestigious award from the NSC for junior researchers), the Distinguished Young Scholar Award of the Taiwan IC Design Society, the ACM Taipei Chapter Young Researcher Award, the IEEE Tainan Chapter Gold Member Award, the Invitational Fellowship of the Japan Society for the Promotion of Science, Japan, and the Humboldt Research Fellowship from the Alexander von Humboldt Foundation, Germany.