An ILP-based Obstacle-Avoiding Routing Algorithm for Pin-Constrained EWOD Chips

Jia-Wen Chang, Tsung-Wei Huang, Tsung-Yi Ho,
Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan

***Abstract*—Electrowetting-on-dielectric (EWOD) chips have become the most popular actuator particularly for droplet-based digital microfluidic (DMF) 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 EWOD-chip routing problem needs to address the issue of signal sharing for pin-count reduction under a practical constraint posed by 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 works consider the EWOD-chip routing with obstacles. To remedy this insufficiency, we propose in this paper the first obstacle-avoiding routing algorithm for pin-constrained EWOD chips. Our 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 our obstacle-avoiding routing algorithm for pin-constrained EWOD chips.**

I. Introduction

Electrowetting-on-dielectric (EWOD) chips have emerged as the most widely used actuators particularly for *droplet*-based digital microfluidic (DMF) platforms [14]. EWOD chips enable electrical manipulations of droplets on a two-dimensional (2D) microfluidic array with various advantages such as flexibility, accuracy, parallel processing, and automated controls [9]. These advantages are increasing the practicality of applications on miniaturized DMF platforms, including immunoassays, DNA sequencing, and point-of-care diagnosis [15].

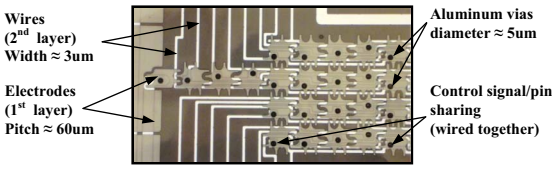

Figure 1. Schematic view of an EWOD chip.

The general diagram of EWOD chips is schematically presented in Figure 1 [12], [14]. It comprises two major layers of 2D electrodes patterned in the first layer and conduction wires routed in the second layer, as well as an inter-insulator of silicon dioxide for via holes 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], [13].

Typically, the regular design flow of EWOD chips consists of three major stages, electrode addressing, routing (i.e., wire routing), and fabrication [12]. In this paper we shall focus on automated designs of electrode addressing and routing, which are two key dominating stages and dominate the manufacturing complexity and fabrication cost [9]. Electrode addressing is a method through which electrodes are addressed with control pins to identify input signals. Since the control pins are actuated by an external controller which supplies a limited number of signal ports, it is infeasible to actuate a large number of control pins especially for high-density electrode array. 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 pins by assigning a single control pin to multiple electrodes with mutually compatible control signals [16].

After electrodes are appropriately addressed with control pins, conduction wires must be routed to establish connections between pins and signal ports. This routing problem becomes more critical than ever for modern EWOD-chip designs which need to consider several routing obstacles incurred 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 the temperature variation for 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 the EWOD-chip routing, conduction wires should avoid routing through these obstacles, thereby increasing the problem complexity. Therefore, currently manual design methods might suffer from either poor solution quality or time-consuming human effort without the assistance of CAD tools.

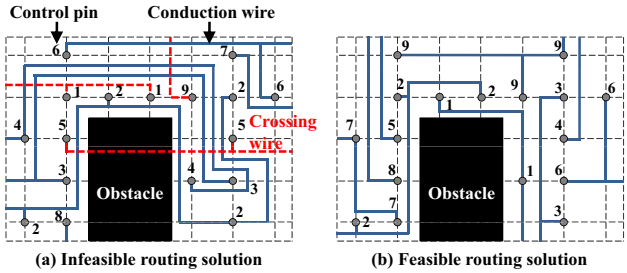

Figure 2. Comparison of two different design methods for performing the same droplet controls. (a) Separate consideration of electrode addressing and routing. (b) Simultaneous consideration of electrode addressing and routing.

Despite the interdependence between electrode addressing and routing, most EWOD-chip design methods treat the two problems as being independent. The potential design gap becomes even critical when the concern of routing obstacles is involved. Specifically, if the electrode addressing and obstacle-avoiding routing are not simultaneously considered, the feasibility and quality of resulted routing solution may be inevitably limited. For example, Figure 2 illustrates two routing solutions under two different design methods that perform the same droplet controls. In the case of (a), which is an infeasible routing solution due to separate consideration of electrode addressing and routing, additional post processes such as electrode readdressing and rerouting or even a multi-layer routing structure should be considered. As a result, the entire design effectiveness will be greatly restricted. In contrast, in (b), simultaneous consideration

978-1-4673-0772-7/12/$31.00 ©2012 IEEE

avoids wiring across the obstacle, thereby providing a higher solution feasibility and quality in terms of routability and wirelength.

Regarding all the above discussions, it is necessary to develop an integrated design automation for pin-constrained EWOD-chip designs. Consequently, we propose in this paper an obstacle-avoiding routing algorithm 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 [10], [11], [16], [18], and there is only one work proposed in [10] considering the automated routing for pin-constrained EWOD chips. The state-of-the-art work in [10] adopts a two-stage technique of global routing followed by a progressive routing scheme. Although this method provides a solution to automated routing for pin-constrained EWOD chip, it is based on the assumption of non-existent obstacles. The entire routing procedure might confront severe routability problems if we attempt to apply this method to solve the obstacle-avoiding routing. And such an impediment justifies the necessity of a dedicated obstacle-avoiding 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 in the literature that provides integrated electrode addressing and routing considering on-chip obstacles. Our contributions can be summarized as follows:

- We consider the EWOD chips with the presence of obstacles and introduce a practical problem formulation of obstacle-avoiding routing for EWOD-chip designs.
- We propose the first routing algorithm to solve this practical problem from EWOD-chip designs, which can relieve the current design burden of time-consuming manual optimizations.
- Our algorithm, based on effective integer-linear-programming (ILP) formulation as well as efficient routing framework, solves the design problem with high routability while keeping the induced design complexity minimized.

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

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 Section IV details the proposed algorithm to solve this practical routing problem. Section V and Section VI shows the experimental results and concludes this paper, respectively.

II. Pin-Constrained EWOD-Chip Designs

Typically, EWOD chips are 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 which specifies the maximum allowable pin count. This criterion brings about the *pin-constrained EWOD-chip designs* in the genre. 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, broadcast addressing, has been presented in [16]. 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"(de-actuated), or "X"(don't-care)) of the electrode at a specific time step [16]. Two electrode actuation sequences are identified as being *compatible*, if either the values of two bits at every time step are the same, or the value of one bit is "X". 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 operations of bioassays. Researchers also model the broadcast addressing into a compatibility graph [10], [16], where the vertex set represents the electrode set and an edge between two electrodes indicates their corresponding actuation sequences are compatible. Therefore, the derivation of a broadcast-addressing result can be mapped to a graph problem of clique partition.

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 generally locate outside the component (i.e., defined as the 2D electrode array) boundary, the routing problem that connects these inside terminal pins to outsides signal ports is similar to the typical escape routing problem appearing in many VLSI designs [4]. However, in pin-constrained EWOD-chip designs, multiple electrodes may share the same control pin and therefore a single control signal may actuate multi-terminal pins. To realize the electrical connections, multi-terminal 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 of two-terminal pins, unsuitable for the EWOD-chip routing problem. In addition to establishment of escape routing connections, modern EWOD-chip routing should address the issue of routing obstacles. For avoiding signal transmission error, any likelihood of routing wires across these obstacles is prohibited. Regarding the 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.

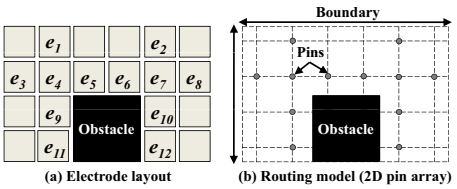

Figure 3. Routing model.

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 [8]. We allow wires to

be routed in horizontal and vertical fashions with 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 2D pin array (see Figure 3). The routing model used in this paper is the same as that in [10], which is based on uniform grid structure with a maximum number of three wires passing through adjacent pins. Besides, a multi-layer arrangement of wiring connections necessitates a mechanism for passing signals between layers (e.g., vias and contacts), raising a cost and reliability issue of PCB-based fabrication. Accordingly, we focus on single-layer routing in this paper.

III. Problem Formulation

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

Input: *A set of electrodes 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:

- 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.*
- Routing constraint: *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. Algorithm

The overview of our ILP-based obstacle-avoiding routing algorithm for EWOD chips are presented in Algorithm 1. 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 based on the electrode grouping result (line 6). The entire routing iterations terminate when R covers the whole chip. 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 (line 7-9).

In following sections, we detail the proposed ILP-based routability-driven electrode grouping and then the 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 will incur wiring detours, which may cause an infeasible routing solution. For example in Figure 2, the electrode grouping result in (a) causes an infeasible routing solution with many detours, while the electrode grouping result in (b) leads to a feasible one. Thus, it is necessary to incorporate the routability issue into the electrode grouping. To this end, we propose a routability-driven

Algorithm 1: ILP-Based Obstacle-Avoiding Routing Algorithm

Input : A 2D 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 ← 0;
3      while R ≤ W + H do
4          R ← 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

electrode grouping method based on ILP formulation. The major goal is to minimize the pin count (i.e., maximize the likelihood of pin count reduction) so as to meet the pin constraint while taking routability issues into account. Idea here is try to avoid unnecessary detours by utilizing the constraints of ILP formulation. We identify there are two major factors having the potential to introduce detours, called *net interference* and *obstacle crossing*.

- *Net interference*: The net interference describes a situation of how the routing path of a net interfere others, particularly for the cause of detours. Nets on chip may interfere each other when conducting wire routing. A net with longer wiring distance reflects an aspect of more likelihood to block the routing paths of other wires than that of a shorter one. 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 the example in Figure 4(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 Figure 4(b). Therefore, during the electrode grouping, we manage to keep the Manhattan distance among electrodes of each net as minimum as possible.

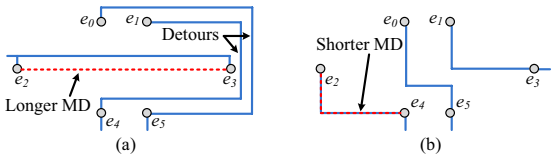

Figure 4. A grouping result with longer Manhattan distance (MD) in (a) incurs more detours than that of shorter MD in (b).

- *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 Figure 5(a), the bounding box of net (e_0, e_1) overlaps with the obstacle horizontally, which is referred to as *horizontal crossing* in this paper. To avoid crossing with obstacle, routing e_0 and e_1 must incur detour. Similarly, Figure 5(b) shows the type of *vertical crossing*. Therefore, during the electrode grouping, we intend to prevent the bounding box of a net from horizontal/vertical

crossing with obstacles.

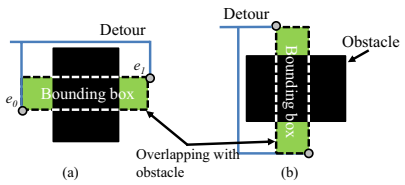

Figure 5. Overlapping between the obstacle and the net incurs detours. (a) Horizontal crossing. (b) Vertical crossing.

To consider the routability issue in electrode grouping, we incorporate the above two factors into the ILP formulation. There are two key strategies in our ILP formulation. The first strategy is associating the factor *net interference* with a parameter R to limit the Manhattan distance among 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 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 the other electrodes which are nearby within R. By this strategy, the nets with long wirelength could be avoided . The second strategy is considering *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 the obstacles. Hereafter, we introduce the objective function and constraints of our ILP formulation.

TABLE I
NOTATIONS USED IN OUR 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
$OC(p_i, p_j, o_l)$	a 0-1 constant represents the bounding box of p_j has horizontal/vertical crossing with obstacle o_l after p_i and p_j are merged into one group

1) ILP formulatioin: Given the set of n electrode groups $P = \{p_1, p_2, \ldots, p_n\}$ where each $p_i \in P$ indicates an electrode group, the parameter R, the set of m obstacles $O = \{o_1, o_2, \ldots, o_m\}$, and the compatibility graph G of electrodes, the ILP formulation is presented as follows:

Objective function:

$$Minimize : \Sigma_{i=1}^{n} u_i \tag{1}$$

Subject to:

$$\Sigma_{j=1}^{n} a_{ij} = 1, 1 \le i \le n \tag{2}$$

$$\Sigma_{i=1}^{n} a_{ij} \le M \cdot u_j, 1 \le j \le n \tag{3}$$

$$a_{ij} \le c_{ij}, 1 \le i, j \le n \tag{4}$$

$$a_{ik} + a_{jk} \le 1 + c_{ij}, 1 \le i, j, k \le n \tag{5}$$

$$a_{ik} + a_{jk} \le 2 - NI(p_i, p_j), \forall p_i, p_j \in P, 1 \le i, j, k \le n \tag{6}$$

$$a_{ik} + a_{jk} \le 2 - OC(p_i, p_j, o_l), \forall p_i, p_j \in P, \forall o_l \in O, 1 \le i, j, k \le n \tag{7}$$

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

- *Grouping constraints*: We merge the electrode groups by assigning each $p_i \in P$ to a group number (1 to n). Electrode groups are merged into one group if these electrode groups are assigned to the same number. Constraint (2) guarantees that each electrode group can only be assigned to a group number. And constraint (3) is used to maintain the 0-1 variable u_i. u_i is 1 if there is at least one electrode group assigned to group number i. Otherwise, it is 0 (i.e., the group number i is not used). However, this assignment formulation will incur redundant solution space. Without affecting the quality of ILP result, we use constraint (4) to restrict the group number that each electrode group can be assigned to. This constraint states that electrode group p_i can be assigned to group number j if the electrodes in electrode groups p_i and p_j are mutually compatible.
- *Broadcast constraints*: Constraint (5) states that for each pair of electrode group (p_i, p_j), if p_i and p_j are assigned to the same group number k, electrodes in p_i and p_j should be mutually compatible.
- *Net interference*: As afore mentioned, 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 into constraint (6).
- *Obstacle crossing*: The detours caused by horizontal crossing or vertical crossing should also be minimized. The constraint (7) realizes this demand. For each pair of electrode group (p_i, p_j), p_i and p_j cannot be merged together if the resulting group incurs horizontal/vertical crossing.

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 analysis the result and obtain the net information of these electrode groups, denoted as N. Each net in N is a 2-terminal net, where each terminal indicates an electrode group in previous routing iteration. Note that in ILP result, the electrode groups assigned to the same group number are going to be merged into one electrode group. The goal now 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. Then, we iteratively pick up k smallest passing count nets from N, denoted as N', and route each one (line 4-11), where k is initialized as 1 in each wire routing iteration (line 2). The wiring connection of each net can be electrode-to-wire, electrode-to-electrode, or wire-to-wire path, which connects two groups and merges them into one group. These paths can be quickly found by maze routing. In case of an infeasible route,

Algorithm 2: Wire routing

```
   Input : Electrode grouping information from ILP solution
   /* N: net set obtained from ILP solution                    */
1  begin
2      k ← 1;
3      while N ≠ φ do
4          N' ← pop k smallest passing count nets from N;
5          for each n_i ∈ N' do
6              Route n_i;
7              path_i ← routing path of n_i;
8              if Failure route then
9                  drop path_i;
10             end
11         end
12         conduct the trial routing;
13         if unsuccessful trial route then
14             drop path_i, n_i ∈ N';
15             assign the path_i as high penalty, n_i ∈ N';
16             if continuous unsuccessful trial routes for Z times then
17                 k ← k + 1;
18             end
19         end
20         if result pin count satisfies pin constraint then
21             break;
22         end
23     end
24     establish escape routing for each electrode group;
25     return routing result;
26 end
```

we neglect the routing of failed net (i.e., $n_i \in N'$). When nets in N' are routed (neglect the failed nets), we adopt a trial routing to check if there exists a feasible escape route for each of electrode group (line 12). There exists many trial routing algorithms in VLSI routing technology [4], and in this paper we use the network-flow based algorithm, which is typically used for escape routing, as our trial routing method [17]. A successful trial route represents the routing paths of N' are permissible while an unsuccessful trial route reveals that the routing paths of N' blocks that of other electrode groups. Therefore, for an unsuccessful trial route, we neglect the routing of the nets in N'. To avoid duplicate routing path in the afterwards routing iteration, we use a history-based technique by assigning the failed routing path a penalty (line 15). In addition, continuous trial routes indicates that there may be too many escaping electrode groups. Specifically, it is too congested to escape such many electrode groups from the chip. Note that each net can merge two electrode groups into one group, which reduces pin count by one. Therefore, to further reduce pin count, we increase k by 1 when Z continuous unsuccessful trial routes occurs, where Z is an user defined parameter (line 16-18). Finally, we construct the escape route for each electrode group including those groups containing only one electrode (line 24).

The entire routing procedure runs once for one routing iteration (i.e., under the specified R). Then, we check whether the resulted pin count can satisfy the pin constraint or not. If there is a violation of pin constraint, we increment R by one and proceed to next routing iteration. On the other hand, once the pin constraint is satisfied, the routing procedure terminates. Note that in case of a failed route that cannot satisfies the pin constraint, designers should resort to another controller with a higher pin-count specification.

C. Exemplification

In this subsection we use an example to exemplify our algorithm, as illustrated in Figure 6. The index aside each pin in Figure 6 represents the pin number assigned to corresponding electrode. Suppose P_{max} is 16 and there are 24 electrodes on the chip.

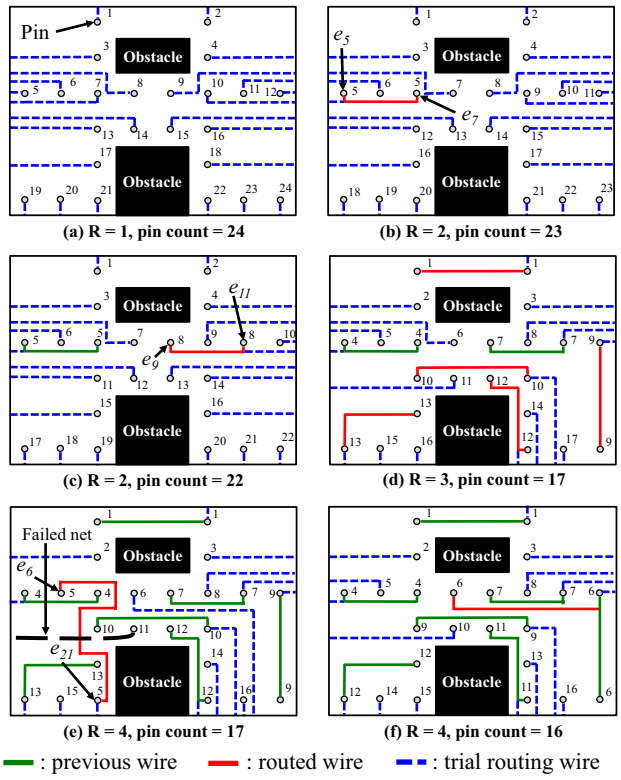

Figure 6. Exemplification of our algorithm. The index aside each pin indicates the pin number assigned to the electrode. (a) Routing result after the iteration $R = 1$. (b)-(c) Based on the electrode grouping result of ILP, we route the nets (e_5, e_7), (e_9, e_{11}) and obtain successful trial routing results sequentially. (d) Routing result after the iteration $R = 3$. (e) Rip-up the blocking net when trial routing fails. (f) Final result after the iteration $R = 4$.

We begin our algorithm with $R = 1$ and obviously each electrode cannot be grouped with others under this condition. Thus each electrode is addressed with dedicated pin. Next, we conduct escape routing (i.e., trial routing) for each electrode group (i.e., each electrode has a group itself) to see whether the current result is available. The routing result obtained in first iteration is illustrated in Figure 6(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. Since $k = 1$ now, we pick up one net at a time. We route the smallest passing count net (e_5, e_7) first by maze routing, followed by conducting trial routing to ensure all the routing of each electrode group can escape to the boundary of chip, as illustrated in Figure 6(b). It can be observed that the escape routing in Figure 6(b) is successful. Hence e_5 and e_7 can be grouped together, thereby 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 (see Figure 6(c)). Our wire routing for this iteration is completed for that there is no untried net in N'. Since required pin count now is 22 while P_{max} is 16, we proceed to the next iteration with $R = 3$. By adopting the same routine (see (d) and (f)) until pin constraint is meet, a feasible solution with desirable pin count is obtained in (f).

For the case of failure route (i.e., net route or trial route) in Figure 6(e), where the wire of routing (e_6, e_{21}) obstructs the pathway of pin

TABLE II
OVERALL COMPARISON BETWEEN THE [10]-EXTENSION, THE HEURISTIC, AND OUR ALGORITHM

Chip	$\|E_e\|$	Size	O(%)	P_{max}	[10]-extension				Baseline				Ours			
					#Pin	WL	#Fail	CPU	#Pin	WL	#Fail	CPU	#Pin	WL	#Fail	CPU
DNA-1	211	16 × 24	13%	128	-	-	31	-	-	-	24	-	128	3003	0	3.7
DNA-2	77	13 × 21	8.7%	32	-	-	6	-	-	-	2	-	32	1113	0	0.7
random-1	24	6 × 8	12.5%	16	-	-	8	-	16	300	0	0.0	16	271	0	0.0
random-2	59	15 × 15	16.8%	32	-	-	5	-	32	1256	0	0.4	32	991	0	0.4
random-3	62	15 × 15	13.7%	32	-	-	18	-	32	1571	0	0.8	32	1153	0	0.5
random-4	91	15 × 15	12.4%	64	-	-	15	-	64	1650	0	1.0	64	1417	0	0.4
random-5	256	20 × 30	15.0%	128	-	-	34	-	-	-	48	-	128	4742	0	11.4
random-6	400	30 × 40	18.3%	256	-	-	71	-	-	-	22	-	256	9099	0	26.1
Total							188				96				0	

11, we rip-up this failure net. We then neglect this net in this iteration. To avoid duplicate routing path in afterwards routing iterations, the original routing path of e_6 and e_{21} is assigned with penalty. Finally, we have the entire arrangement of wire and pins (Figure 6(f)).

V. 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. The parameter, Z, in wire routing stage is set as 5. 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 (i.e., 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 (i.e., 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 6 hard test chips with obstacles. Table II shows the statistics of the these chips. "$|E_e|$" denotes the number of electrodes, "Size" denotes the chip size, "O(%)" denotes the percentage of obstacle occupation, "P_{max}" denotes the maximum allowable number of control pins, "#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.

For comparison purpose, we implement two routing methods. The first method is the extension of [10], namely "[10]-extension". We modify routing graph used in [10] by further considering the obstacles, and use breadth-first search (BFS) to route the wire without crossing with obstacles. The second method is to group electrodes and route wires separately, namely "Baseline". In "Baseline" manner, we iteratively and pairwisely 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 II lists the overall comparison results. Our algorithm achieves better routability by completing all 8 test cases (100.0%), while the "[10]-extension" and the "Baseline" complete 0 (0.0%) and 4 (50.0%) test cases, respectively. The major reason that our algorithm outperforms the "[10]-extension" is that the routability of [10] 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 routability degradation. 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 [10], and thereby restraining the routability. To summarize, the experimental results demonstrate the effectiveness of the proposed routing algorithm on solving the obstacle-avoiding routing problem for pin-constrained EWOD chips.

VI. CONCLUSIONS

In this paper, we have introduced a practical problem for EWOD-chip routing with on-chip obstacles. We have presented the first obstacle-avoiding routing algorithm to deal with this design problem. Our algorithm, based on effective integer-linear-programming (ILP) formulation as well as efficient routing framework, can complete the routing with high routability while inducing low design complexity. Two real-life EWOD chips used for DNA sample preparation and a set of self-generated test chips have been used to evaluated the effectiveness of our routing algorithm for EWOD chips with the presence of obstacles.

REFERENCES

[1] http://www.liquid-logic.com/
[2] http://microfluidics.ee.duke.edu/
[3] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/
[4] C. J. Alpert, D. P. Mehta, S. S. Sapatnekar, "Handbook of Algorithms for Physical Design Automation," CRC Press, 2009.
[5] K. Chakrabarty, "Towards fault-tolerant digital microfluidic lab-on-chip: defects, fault modeling, testing, and reconfiguration," *Proc. IEEE ICBCS*, pp. 329–332, 2008.
[6] R. B. Fair, "Digital microfluidics: Is a true lab-on-a-chip possible?" *Microfluidics and Nanofluidics*, vol. 3, pp. 245–281, 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 and Test*, vol. 24, pp. 20–24, 2007
[8] J. Gong and C. J. Kim, "Direct-referencing two-dimensional-array digital microfluidics using multilayer printed circuit board," *IEEE J. MEMS*, no. 2, pp. 257–264, 2008.
[9] T.-Y. Ho, J. Zeng, and K. Chakrabarty, "Digital microfluidic biochips: A vision for functional diversity and more than Moore," *Proc. IEEE/ACM ICCAD*, pp. 578–585, 2010.
[10] 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," *Proc. IEEE/ACM ICCAD*, pp. 425–431, 2010
[11] C. C.-Y. Lin and Y.-W. Chang, "ILP-based pin-count aware design methodology for microfluidic biochips," *Proc. ACM/IEEE DAC*, pp. 258–263, 2009.
[12] 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 Multi-Layer Insulators.," *Sensors and Actuators B: Chemical*, pp. 465–470, 2010.
[13] M. G. Pollack, A. D. Shenderov, and R. B. Fair, "Electrowetting-based actuation of droplets for integrated microfluidics," *LOC*, pp. 96–101, 2002.
[14] 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 and Nanofluidics*, pp. 75–89, 2009.
[15] F. Su, K. Chakrabarty, and R. B. Fair, "Microfluidics based biochips: Technology issues, implementation platforms, and design-automation challenges," *IEEE Trans. on CAD*, pp. 211–223, 2006.
[16] T. Xu and K. Chakrabarty, "Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips," *Proc. ACM/IEEE DAC*, pp. 173–178, 2008.
[17] T. Yan and M. D. F. Wong, "A correct network flow model for escape routing," *Proc. ACM/IEEE DAC*, pp. 332–335, 2009.
[18] Y. Zhao and K. Chakrabarty, "Co-optimization of droplet routing and pin assignment in disposable digital microfluidic biochips," *Proc. ACM ISPD*, pp. 69–76, 2011.