A Two-Stage Integer Linear Programming-Based Droplet Routing Algorithm for Pin-Constrained Digital Microfluidic Biochips

Tsung-Wei Huang and Tsung-Yi Ho, *Member, IEEE*

***Abstract*—With the increasing design complexities, the design of pin-constrained digital microfluidic biochips (PDMFBs) is of practical importance for the emerging marketplace. However, solutions of current pin-count reduction are inevitably limited by simply adopting it after the droplet routing stage. In this paper, we propose the *first* droplet routing algorithm for PDMFBs that can integrate pin-count reduction with droplet routing stage. Furthermore, our algorithm is capable of minimizing the number of control pins, the number of used cells, and the droplet routing time. We first present a basic integer linear programming (ILP) formulation to optimally solve the droplet routing problem for PDMFBs with simultaneous multiobjective optimization. Due to the complexity of this ILP formulation, we also propose a two-stage technique of global routing followed by *incremental* ILP-based routing to reduce the solution space. To further reduce the runtime, we present a *deterministic* ILP formulation that casts the original routing optimization problem into a decision problem, and solve it by a binary solution search method that searches in logarithmic time. Extensive experiments demonstrate that in terms of the number of the control pins, the number of the used cells, and the routing time, we obtain much better achievement than all the state-of-the-art algorithms in any aspect.**

***Index Terms*—Broadcast-addressing biochips, integer linear programming, routing.**

I. Introduction

As THE MICROFLUIDICS technology advances, digital microfluidic biochips (DMFBs) have attracted much attention recently. Compared with the conventional laboratory experiment procedures, which are usually cumbersome and expensive, these miniaturized and automated DMFBs show numerous advantages such as high portability, high throughput, high sensitivity, minimal human intervention, and low sample/reagent volume consumption. Due to these advantages, various laboratory procedures and practical applications such as infant health care, point-of-care disease diagnostics, environmental toxin monitoring, and drug discovery have been successfully demonstrated in DMFB platforms [15].

Manuscript received June 2, 2010; revised August 17, 2010; accepted October 5, 2010. Date of current version January 19, 2011. This work was supported in part by National Science Council of Taiwan, under Grants NSC 99-2220-E-006-013 and NSC 99-2221-E-006-220. This paper was recommended by Associate Editor P. Saxena.

The authors are with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: twhuang@eda.csie.ncku.edu.tw; tyho@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.2010.2097190

In performing various fluidic-handling functions, droplet-based operations are introduced in DMFB platforms [7]. A primary issue is the control scheme of droplet movements. In the most common droplet control scheme, each electrode is directly and independently addressed and controlled by a dedicated control pin, as illustrated in Fig. 1(a), which allows each electrode to be individually activated. In this paper, we refer to these types of DMFBs as *direct-addressing* DMFBs. By independently controlling the voltage of electrodes, droplets can be moved along this activation line of electrodes due to the principle of electrowetting-on-dielectric [15]. Therefore, many fluidic operations such as mixing and dilution can be performed anywhere on the DMFB within different time intervals. For example, a mixing reaction can be performed by moving two droplets toward the same cell, and then turning them around a pivot for a uniform mixing solution [7].

Previous droplet routing algorithms mainly focus on direct-addressing DMFBs [3], [8], [10], [16], [18], [22]. This scheme maximizes the freedom of the droplet manipulation, but it suffers from the major deficiency that the number of control pins rapidly increases as the system complexity increases. Moreover, a large number of control pins necessitates multiple conductive layers, which potentially raise the price of production cost. Specifically, the interconnect wiring problem for high pin-count demand has made this architecture only suitable for small-scale biochips.

Recently, *pin-constrained* digital microfluidic biochips (PDMFBs) have raised active discussions to overcome this problem. One of the major approaches, *broadcast-addressing* scheme, provides high throughput for bioassays and reduces the number of control pins by identifying and connecting them with "*compatible*" control signals. In other words, multiple electrodes are controlled by a single signal source and are thus activated simultaneously, as shown in Fig. 1(b). To realize the broadcast-addressing scheme for pin-count reduction, many state-of-the-art algorithms such as routing path partitioning, electrode grouping, and control signal merging have been proposed in the literature [17], [19], [20], [23]. However, these works approach the broadcast-addressing scheme only by simply post-processing the routing solutions with sharing compatible control signals. Therefore, the quality of such sharing methods is inevitably limited by given routing solutions and may result in suboptimal outcomes.

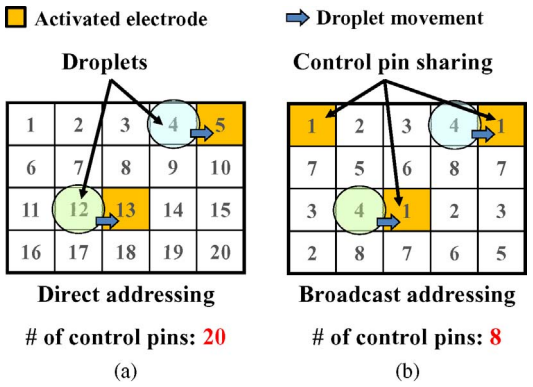

Fig. 1. Illustration of direct-addressing and broadcast-addressing schemes. (a) Direct-addressing scheme with high required number of control pins. (b) Broadcast-addressing scheme with low required number of control pins.

As more complex bioassays are concurrently executed on a single digital microfluidic platform, the complexity of the system and the number of the electrodes are bound to increase steadily [17]. Recently, a DMFB that embeds more than 600 000 20 μm by 20 μm electrodes has been demonstrated [2]. Thus, the design of droplet control scheme with pin-count reduction is of great practical importance for PDMFBs. Besides, minimizations of routing time and used cells during bioassay execution are also critical for real-time applications and high fault-tolerant designs [6]. Hence, it is desirable to develop an integrated routing algorithm that can concurrently handle these concerns.

Consequently, we propose in this paper the *first* ILP-based droplet routing algorithm that concurrently takes the droplet routing and the broadcast-addressing schemes into consideration for PDMFBs. The main challenge of this routing problem is to derive different constraints into ILP formulations, while ensuring correct droplet movements and minimizing the number of control pins. Different from the aforementioned works, our algorithm, by using well-formulated constraints in ILP formulations, is capable of minimizing the number of the control pins, the number of the used cells, and the routing time to achieve better design performance.

A. Previous Work

In DMFB marketplace, designing PDMFBs is a critical issue in biochip design automations due to their large impact on production cost and fabrication issues. There are many state-of-the-art algorithms in the literature for handling the pin-count reduction problem [12], [17], [19], [20], [23]. The work in [17] proposes an array-partition-based method to group the electrode set without introducing unexpected fluidic-level behaviors for scheduled fluidic operations. The work in [19] partitions the scheduled routing paths into different regions while minimizing the interference among these regions. Then, for each region, a graph-coloring-based formulation is derived to minimize the required number of control pins. However, the two works suffer from a high number of partitioned regions for multioverlapped routing paths, and thus the required pin count may potentially increase. The work in [20] presents a clique-partition-based algorithm to formulate the compatibility between control signals derived from scheduled routing results. By recognizing a minimum clique partition, the required number of control pins can be optimized. However, since the minimum clique partition is well-known as an NP-hard problem, a heuristic method of iterative clique recognitions is also proposed. The work in [23] adopts a two-phased algorithm of clique recognition followed by post-processing the pin-count reduction. They first use a heuristic to generate a set of control pins for performing scheduled fluidic operations. Then, based on the pin-assignment result, a stalling strategy is applied to synchronize the movements among different droplets in a parallel manner. However, stalling the droplet movements also increases the total routing time, thereby introducing another practical problem such as reducing the reliability of biochips [6]. Recently, a novel pin-count aware design methodology for pin-constrained digital microfluidic biochips (PDMFBs) is proposed in [12]. This paper further integrates various pin-count saving techniques into fluidic-level synthesis, and then systematically addresses electrodes according to pre-classified categories of pin demand. However, the minimization issues of pin count, number of used cells, and droplet routing time, are still separately considered from the droplet routing. Therefore, the solution quality may be restricted.

Although these state-of-the-art algorithms provide many pin-count reduction techniques to tackle the steadily increased number of electrodes, a common drawback is the separate considerations for droplet routing and pin-count reduction. However, the solution quality and performance of pin-count reduction actually depend on given routing solutions, which reveals a demand for design convergence. Therefore, it is necessary to develop an integrated method to assist in this concern.

B. Our Contribution

In this paper, we propose the *first* droplet routing algorithm for PDMFBs that minimizes the number of control pins, the number of used cells, and the routing time. We first present a basic integer linear programming (ILP) formulation to optimally solve the droplet routing problem for PDMFBs. Due to its complexity, we also propose a two-stage technique of global routing followed by incremental ILP-based routing to reduce the solution space effectively. Our algorithm divides the original routing problem to global routing paths spatially to reduce the solution space of ILP formulations. In this way, the original problem is reduced to a manageable size, then we can practically apply an incremental ILP-based method to finding a high-quality solution within reasonable CPU time. To achieve further efficiency, we propose a *deterministic* ILP formulation that casts the original optimization into a decision problem and solve it by a search technique with logarithmic time complexity. The major contributions of this paper include the following.

1) We propose the *first* droplet routing algorithm that considers the droplet routing and the broadcast-addressing scheme for PDMFBs. In contrast with the previous works that start with an initial direct-addressing-based routing result, our algorithm has higher flexibility to solve the droplet routing problem on PDMFBs globally.
2) Unlike the previous works that only minimize the number of control pins, our algorithm can minimize not only

the number of the control pins but also the number of used cells and the routing time, which is attributed to the well-founded formulations of the constraints into our ILP formulations.

3) To tackle the complexity of the basic ILP formulations, we propose a two-stage routing scheme of global routing followed by incremental ILP-based routing. For the basic ILP, the problem instance is whole 2-D plane and it handles all droplets simultaneously. For our two-stage ILP, the problem instance is reduced to global routing paths and the droplets are routed in incremental manner that reduce the solution space significantly. Therefore, our algorithm can obtain a high-quality solution within reasonable CPU time.
4) To further reduce the runtime, we present a *deterministic* ILP formulation that casts the original routing optimization problem into a decision problem, and then solves it by a binary solution search method that searches in logarithmic time.

Compared with the direct-addressing and the broadcast-addressing schemes, the extensive experiments demonstrate that in terms of the number of the control pins, the number of the used cells, and the routing time, we acquire much better achievement than all the current state-of-the-art algorithms in any aspect.

The remainder of this paper is organized as follows. Section II details the background of PDMFBs, control mechanism of broadcast-addressing scheme, and formulates the droplet routing problem. Section III presents the basic ILP formulations for droplet routing problem. Section IV introduces the two-stage ILP routing scheme to tackle the design complexity incurred by the basic ILP formulations. Section V analyzes the design complexity of the proposed ILP formulation. Section VI shows the experimental results and Section VII provides the concluding remarks.

II. Preliminaries

In this section, we first show the PDMFBs and the control mechanism of broadcast-addressing scheme. Then we present the problem formulation of the droplet routing problem for PDMFBs.

A. Pin-Constrained Digital Microfluidic Biochips

In addressing the need for low-cost and practical fabrication issues, pin-count reduction has served as a major solution in current DMFB design automations [5]. These kinds of biochips, also referred to as PDMFBs, reduce the required number of control pins either by reforming the electrode architecture or grouping the electrode set with mutually compatible control signals. Typically, there are two kinds of PDMFBs, *cross-referencing* biochips and *broadcast-addressing* biochips, respectively.

In cross-referencing biochips, the electrode architecture is formed in a row/column manner. That is, the electrode in one row (or column) is connected to a single control pin. Therefore, the number of control pins is greatly reduced as

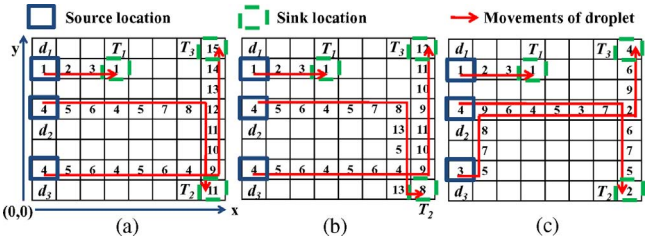

Fig. 2. Example of an 8 × 8 DMFB with three droplets. (a) Apply the broadcast-addressing scheme to a routing result. (b) Apply the droplet routing and broadcast-addressing scheme simultaneously. (c) Apply the droplet routing and broadcast-addressing scheme simultaneously with minimizing the number of control pins, the number of used cells, and the routing time.

proportional to the perimeter of the chip rather than the area of the chip. By activating the row and column electrodes with opposite signals, the electrode spots at their intersections become most hydrophilic and thus droplets move toward them [9]. However, the simultaneous driving of multiple droplets in this platform is limited due to the large electrode interference, and thus restricts the throughput of bioassays [21]. Furthermore, this kind of biochips also introduces complicated electrical connections and device packaging problems, which increases the fabrication cost.

In broadcast-addressing biochips, the number of control pins is reduced by assigning a single control pin to multiple electrodes with mutually compatible control signals [20]. In other words, multiple electrodes are controlled by a single control signal and are thus driven simultaneously. Compared with cross-referencing biochips, broadcast-addressing biochips offer two major advantages. First, the broadcast-addressing scheme also provides the maximum freedom for droplet movements as the direct-addressing scheme. The number of simultaneous driving of multiple droplets can be maximized without incurring significant electrode interference. Second, the electrodes are still patterned in a regular 2-D array without causing cumbersome wiring and packaging problems. In this paper, we focus on the second kind of PDMFBs, broadcast-addressing biochips.

B. Broadcast-Addressing Scheme

To execute a specific bioassay, the routing and the operation scheduling for droplets are programmed into a microcontroller to drive the electrodes. The information of routing and scheduling is stored in the form of electrode activation sequences. Each bit in the sequence represents the activation status of the electrode in a specific time step, and can be represented as activated ("1"), deactivated ("0"), or do not care ("X"). A do not care signal represents that the input signal of electrode can be either activated or deactivated, which does not change the routing scheme.

An example is shown in Fig. 2(a). When the droplet d_1 moves from $(0, 6)$ to $(1, 6)$, the electrode in cell $(1, 6)$ must be assigned "1" and the cell $(0, 6)$ and $(2, 6)$ must be assigned "0." In this time step, the cell $(3, 6)$ is treated as do not care that we can assign "1" or "0" to this cell which has no impact on d_1's movement. We use the three value "1," "0," and "X" to represent the electrode activation sequences for a bioassay. As shown in Fig. 2(a), when droplet d_1 moves from

cell $(0, 6)$ to cell $(3, 6)$ within time 0 to 3, the corresponding activation sequences of cell $(0, 6)$ to cell $(3, 6)$ (noted as c_1 to c_4) can be represented as "100X," "0100," "X010," and "XX01." By carefully replacing the do not care terms in c_4, we can identify c_4 with c_1 in this activation sequence "1001." We refer to the sequences of c_1 and c_4 as "compatible sequence." In broadcast-addressing scheme, the corresponding electrodes of c_1 and c_4 can be connected to a single control pin. Therefore, compared with direct-addressing scheme, the number of control pins can be significantly reduced. However, with increased design complexities, the solution is inevitably limited by using the direct-addressing-based routing result as the input to apply the broadcast-addressing scheme [20]. In Fig. 2(a), if we only adopt the broadcast-addressing scheme to a given routed result, we need 15 control pins to execute this bioassay. But if we simultaneously consider the routing and the broadcast-addressing scheme, as shown in Fig. 2(b), we only need 13 control pins for this bioassay. In addition to minimizing the number of control pins in PDMFBs, it is desirable to minimize the number of used cells and the routing time for fast bioassay execution and better reliability [6]. In Fig. 2(c), our droplet routing algorithm can concurrently address these optimization issues to minimize the number of control pins, the number of used cells, and the routing time, thereby achieving significantly better routing solution. Therefore, in addition to simultaneously considering the droplet routing and the broadcast-addressing scheme, it is desirable to minimize the number of control pins, the number of used cells, and the routing time for PDMFBs.

C. Problem Formulation

As aforementioned, in addition to minimizing the number of control pins in PDMFBs, it is desirable to minimize the number of unit cells that are used during routing for better fault tolerance. This issue is especially crucial for safety-critical applications, such as patient health monitoring or biosensors for detecting environmental toxins [14]. A DMFB contains primary cells for bioassay execution and spare cells for replacing faulty primary cells to ensure the correctness during bioassay execution. Therefore, to maximize the number of spare cells for better fault tolerance, it is necessary to minimize the number of used cells for droplet routing. Furthermore, droplet transportation time is also critical for applications requiring real-time response for early warnings, such as point-of-care disease diagnostics and monitoring environmental toxins. Moreover, shorter droplet routing time improves the reliability of DMFBs. Longer droplet routing time implies that high activation voltage must be maintained for a long period of time, thereby accelerating dielectric breakdown or defects in physical domain on some cells [4]. Therefore, it is desirable to minimize the droplet routing time (i.e., latest arrival time among routing all droplets) to achieve fast bioassay execution and better chip reliability.

Besides the objectives of droplet routing, there are two routing constraints in droplet routing: the *fluidic constraints* and the *timing constraint*. The fluidic constraints are used to avoid unexpected mixtures between two droplets of different nets during their transportation and it can be further divided into the

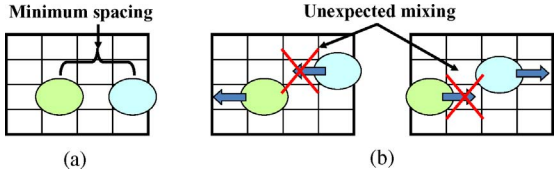

Fig. 3. Illustration of the fluidic constraints. (a) Static fluidic constraint. (b) Dynamic fluidic constraint.

static fluidic constraint and *dynamic* fluidic constraint [16]. Let d_i at cell (x_t^i, y_t^i) and d_j at cell (x_t^j, y_t^j) denote two independent droplets at time t. Then, the following constraints should be satisfied for any t during routing.

1) *Static constraint:* $|x_t^i - x_t^j| > 1$ or $|y_t^i - y_t^j| > 1$.
2) *Dynamic constraint:* $|x_{t+1}^i - x_t^j| > 1$ or $|y_{t+1}^i - y_t^j| > 1$ or $|x_t^i - x_{t+1}^j| > 1$ or $|y_t^i - y_{t+1}^j| > 1$.

The static fluidic constraint states that the minimum spacing between two droplets is one cell for any time t during routing [see Fig. 3(a)]. The dynamic fluidic constraint states that the activated cell for d_i cannot be adjacent to d_j between successive time cycles t and $t + 1$. The reason is that there can be more than one activated neighboring cell for d_j. Therefore, we may have an unexpected mixing between d_i and d_j [see Fig. 3(b)]. Beyond the fluidic constraints, there exists the timing constraint, which specifies the maximum arrival time among routing droplets from source cells to sink cells. Regarding these concerns, the droplet routing problem for the PDMFBs can be formulated as follows.

Input: *A netlist of n droplets* $D = \{d_1, d_2, \ldots, d_n\}$*, the locations of blockages, and the timing constraint* $T_{\max}$.

Constraint: *Both fluidic and timing constraints should be satisfied.*

Objective: *Route all droplets from their source cells to their sink cells while minimizing (1) the number of control pins, (2) the number of used cells, and (3) the droplet routing time.*

III. Basic ILP Formulation for Droplet Routing

In this section, we propose the basic ILP formulation that considers the droplet routing and the broadcast-addressing scheme simultaneously for PDMFBs. We show how the basic ILP formulation optimizes the droplet routing with the three objectives of minimizing the number of control pins, the number of used cells, and the routing time. In addition to sharing the routing paths in a time-multiplexed manner, our basic ILP formulation addresses the issue of scheduling droplets under practical constraints imposed by the fluidic and timing restrictions. Furthermore, our basic ILP formulation simultaneously considers the control signal sharing among different droplet routing paths. For the sake of brevity and generality, we focus on 2-pin net routing. The notations used in our ILP formulations are shown in Table I.

A. Formulation Rules

One of the most difficult challenges of this problem is to model the electrode activation constraint into an ILP formulation, considering the activated ("1"), deactivated ("0"), and do not care ("X") activation terms. To successfully obtain

TABLE I
NOTATIONS USED IN OUR BASIC ILP FORMULATION

D	Set of droplets
C	Set of available cells
B	Set of cells inside one enlarged bounding box of a DMFB (Note that $C \subset B$)
T_{max}	Constraint for maximum droplet routing time
P_{max}	Constraint for maximum available control pins
$E_5^C(x, y)$	Set of cell (x, y) and its four adjacent cells in C
$E_8^C(x, y)$	Set of cell (x, y)'s eight neighboring cells in C
$E_9^C(x, y)$	Set of cell (x, y) and its eight neighboring cells in C
$E_8^B(x, y)$	Set of cell (x, y)'s eight neighboring cells in B
(s_x^i, s_y^i)	Location of the source cell of droplet d_i
(sk_x^i, sk_y^i)	Location of the sink cell of net n_i
$c(i, x, y, t)$	A 0–1 variable represents that droplet d_i locates at cell (x, y) at time t
T_l	Latest arrival time among all droplets (i.e., droplet routing time)
$uc(x, y)$	A 0–1 variable represents that cell (x, y) is used
$st(i, t)$	A 0–1 variable represents that d_i stalls from time $t-1$ to t
$a_0(i, x, y, t)$	A 0–1 variable represents that cell (x, y) must be deactivated in controlling d_i's movement at time t
$a_1(i, x, y, t)$	A 0–1 variable represents that cell (x, y) must be activated in controlling d_i's movement at time t
$a_X(i, x, y, t)$	A 0–1 variable represents that cell (x, y) is do not care in controlling d_i's movement at time t
$A_0(x, y, t)$	A 0–1 variable represents that cell (x, y) must be deactivated in total movements control at time t
$A_1(x, y, t)$	A 0–1 variable represents that cell (x, y) must be activated in total movements control at time t
$A_X(x, y, t)$	A 0–1 variable represents that cell (x, y) is do not care in total movements control at time t
$as(x, y, t)$	Activation sequence of cell (x, y) at time t
$cmp(x_1, y_1, x_2, y_2)$	A 0–1 variable represents the activation sequences of cell (x_1, y_1) and (x_2, y_2) are compatible
$cp(x, y, p)$	A 0–1 variable represents that the cell (x, y) is controlled by pin p
$up(p)$	A 0–1 variable represents that pin p is used

each cell's activation sequence, we add "must" restriction to this constraint in the formulation. As shown in Fig. 4(a), when droplet d_1 moves from cell (1,4) to cell (2,4) at time t, all neighboring cells of cell (1,4) and cell (2,4) must be deactivated, except for the cell (2,4), which must be activated. When droplet d_2 stalls at its original cell at time t, all the neighboring cells of cell (4,1) must be deactivated, while the cell (4,1) must be activated to hold this droplet [17]. Those cells that have no impact on droplet transportation are regarded

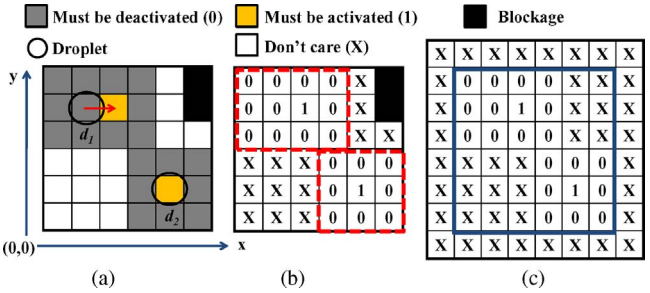

Fig. 4. Modeling of electrode activation constraint. (a) Droplet d_1 moves while the other droplet d_2 stalls. (b) Corresponding activations of electrodes. (c) One bounding box enlarged array.

as do not care terms which can be assigned "1" or "0" [20]. Fig. 4(b) describes the corresponding electrode activation. To model this constraint, we use the notation (x_t^i, y_t^i) to represent the location of droplet d_i at time t. Therefore, the electrode activation constraint can be formulated in the following rules.

1) *EC-Rule I:* if a droplet d_i moves from cell (x_{t-1}^i, y_{t-1}^i) to cell $(x_t^i, y_t^i) \in E_5^C(x_{t-1}^i, y_{t-1}^i)$, all the cells $(x', y') \in \{E_9^C(x_{t-1}^i, y_{t-1}^i) \cup E_9^C(x_t^i, y_t^i)\}$ must be deactivated at time t, except for the cell (x_t^i, y_t^i), which must be activated at time t.
2) *EC-Rule II:* if a droplet d_i stalls at time t, the exact number of must-be-deactivated cells is 8; otherwise, if d_i moves to the four adjacent cells at time t, the exact number of must-be-deactivated cells is 11.
3) *EC-Rule III:* the cells that have no impact on droplets transportation are do not care terms.

Because of the blockages and the boundary restriction in the microfluidic array, it is hard to directly apply the three rules to the ILP formulations in the cell set C. For example, if a droplet d_i stalls at the location (0, 0) within time $t-1$ to t in Fig. 4(a), due to the boundary restriction of microfluidic array, the exact number of must-be-deactivated cells is 3 instead of 8. In other words, we may need extra constraints and variables to determine the exact number in EC-Rule II, which significantly increases the complexities of the ILP formulations. Therefore, we apply the three rules on the microfluidic array which is enlarged one bounding box of the original microfluidic array to solve this problem. As shown in Fig. 4(c), the cells inside the 8×8 array belong to the cell set B (note that the cell set C is a subset of B). As the example mentioned earlier, to hold the droplet d_i at cell (0, 0) at time t, the exact number of must-be-deactivated cells in B is 8. In this way, we can achieve the three rules without increasing the size of electrode activation constraint.

Another major challenge in the routing problem is to model the broadcast constraint into an ILP formulation. Each activation sequence may contain several do not care terms, which can be replaced by "1" or "0." This feature increases the solution space of our ILP. In other words, a naïve formulation may increase the size of constraints and the complexity of ILP. Therefore, we propose the three major rules to tackle the broadcast constraint as follows.

1) *BC-Rule I:* two activation sequences are compatible if and only if the corresponding binary values are the same.

2) *BC-Rule II:* if the activation sequences of two cells are incompatible, we cannot broadcast the two cells with the same control pin.
3) *BC-Rule III:* if the activation sequences of two cells are compatible, we can broadcast the two cells with the same control pin *or not*.

BC-Rule I states the essence of broadcast-addressing scheme. Both BC-Rules II and III describe the broadcast rules for two cells (i.e., electrodes). For example, given three electrodes with activation sequences as e_1 = 0100X, e_2 = 01001, and e_3 = X0100. Electrodes e_1 and e_3 cannot be assigned by a single control pin since there is no common compatible sequences between e_1 and e_3. In contrast, the do not care term in the activation sequence of e_1 can be replaced by "1" such that e_1 and e_2 can be grouped together and assigned by the same control pin. Similarly, e_1 and e_2 can be separately assigned by different control pins without broadcast addressing.

In the following subsections, we introduce the objective function and constraints of our basic ILP formulations.

B. Objective Function

Our goal is to minimize the number of control pins, the number of used cells, and the droplet routing time. Therefore, the objective function is defined as follows:

$$Minimize: \alpha \cdot \sum_{p=1}^{P_{\max}} up(p) + \beta \cdot \sum_{(x,y)\in C} uc(x,y) + \gamma \cdot T_l \quad (1)$$

where α, β, and γ are set to one as the default value.

C. Constraints

There are total ten constraints in our basic ILP formulations.

1) *Source requirement:* all droplets are at their source location at time zero. Therefore, the source requirement can be represented as follows:

$$c(i, s_x^i, s_y^i, 0) = 1, \forall d_i \in D. \quad (2)$$

2) *Sink requirement:* all droplets must reach their sinks within timing constraint. Once a droplet reaches its sink, it remains there. Therefore, the sink requirements can be represented as follows:

$$\sum_{t=0}^{T_{\max}} c(i, sk_x^i, sk_y^i, t) \geq 1, \forall d_i \in D \quad (3)$$

$$c(i, sk_x^i, sk_y^i, t) - c(i, sk_x^i, sk_y^i, t+1) \leq 0 \quad \forall d_i \in D, 0 \leq t < T_{\max}. \quad (4)$$

3) *Exclusivity constraint:* each droplet has only one location at each time step. Therefore, the exclusivity constraint can be represented as follows:

$$\sum_{(x,y)\in C} c(i,x,y,t) = 1, \forall d_i \in D, 0 \leq t \leq T_{\max}. \quad (5)$$

4) *Computation of the latest arrival time (required droplet routing time):* if a droplet reaches its sink at time t, then the time it reaches its sink can be computed as t times the difference of $c(i, sk_x^i, sk_y^i, t)$ and $c(i, sk_x^i, sk_y^i, t-1)$. Therefore, the computation of latest arrival time can be represented as follows:

$$t \cdot (c(i, sk_x^i, sk_y^i, t) - c(i, sk_x^i, sk_y^i, t-1)) \leq T_l, \quad \forall d_i \in D, 0 < t \leq T_{\max}. \quad (6)$$

5) *Computation of total used cells:* a cell (x, y) is used if a droplet ever located at this cell before. Otherwise, if there is no droplet locating at the cell (x, y) during the whole bioassay execution, the cell is un-used. Therefore, the above two constraints can be represented as follows:

$$uc(x,y) \geq c(i,x,y,t), \quad \forall d_i \in D, (x,y) \in C, 0 \leq t \leq T_{\max} \quad (7)$$

$$uc(x,y) \leq \sum_{d_i \in D} \sum_{t=0}^{T_{\max}} c(i,x,y,t)(x,y) \in C. \quad (8)$$

6) *Droplet movement constraint:* a droplet can have only five possible movements; stall or move to four adjacent cells from t to $t+1$. Therefore, the movement constraint can be represented as follows:

$$c(i,x,y,t) \leq \sum_{(x',y')\in E_5^C(x,y)} c(i,x',y',t+1) \quad \forall d_i \in D, (x,y) \in C, 0 \leq t < T_{\max}. \quad (9)$$

7) *Fluidic constraints:* as described in Section II, there are two fluidic constraints: static and dynamic fluidic constraints. Static fluidic constraint states the minimum spacing between two droplets must be one cell. In other words, there are no other droplets in the 3×3 region centered by a droplet. Therefore, the static fluidic constraint can be represented as follows:

$$c(i,x,y,t) + \sum_{(x',y')\in E_9^C(x,y)} c(j,x',y',t) \leq 1 \quad \forall d_i, d_j \in D, d_i \neq d_j, (x,y) \in C, 0 \leq t \leq T_{\max}. \quad (10)$$

To prevent unexpected mixing during droplet movement, dynamic fluidic constraint requires that at time $t+1$, d_i cannot move to the cell (x, y), which is the neighboring cells of d_j's location at time t. Therefore, the dynamic fluidic constraint can be represented as follows:

$$c(i,x,y,t+1) + \sum_{(x',y')\in E_9^C(x,y)} c(j,x',y',t) \leq 1 \quad \forall d_i, d_j \in D, d_i \neq d_j, (x,y) \in C, 0 \leq t < T_{\max}. \quad (11)$$

8) *Electrode constraints:* EC-Rule I states that if droplet locates at cell (x, y) at time t, this cell (x, y) must be activated. Therefore, this activated rule can be represented as follows:

$$a_1(i,x,y,t) = c(i,x,y,t), \quad \forall d_i \in D, (x,y) \in C, 0 \leq t \leq T_{\max} \quad (12)$$

$$\sum_{(x,y)\in B} a_1(i,x,y,t) = 1, \forall d_i \in D, 0 \leq t \leq T_{\max}. \quad (13)$$

Note that the (13) states the exclusivity constraint in cell set B. The two constraints state the deactivated condition in EC-Rule I as follows:

$$\sum_{(x',y')\in E_8^B(x,y)} a_0(i,x',y',t) \geq 8 \cdot c(i,x,y,t), \quad \forall d_i \in D, (x,y) \in C, 0 \leq t \leq T_{\max} \tag{14}$$

$$\sum_{(x',y')\in E_8^B(x,y)} a_0(i,x',y',t) \geq 7 \cdot c(i,x,y,t-1), \quad \forall d_i \in D, (x,y) \in C, 0 < t \leq T_{\max}. \tag{15}$$

Note that as shown in Fig. 4(b), constraints (14) and (15) only determine the number of cells which must be deactivated in the dash-line area (e.g., the lower bound number of cells that must be deactivated). For example, for droplet 1, we can assign "0" to the cell (0, 0), which still satisfies (14) and (15), but violates the "must be deactivated" condition. To tackle this problem, we use $st(i,t)$ to represent that droplet d_i stalls within time $t-1$ to t, and use EC-rule II to determine the exact number of cells which must be deactivated

$$st(i,t) \geq c(i,x,y,t) + c(i,x,y,t-1) - 1 \quad \forall d_i \in D, (x,y) \in C, 0 < t \leq T_{\max} \tag{16}$$

$$st(i,0) = 1, \forall d_i \in D \tag{17}$$

$$\sum_{(x,y)\in B} a_0(i,x,y,t) = 8 + 3 \cdot (1 - st(i,t)) \quad \forall d_i \in D, 0 \leq t \leq T_{\max}. \tag{18}$$

Note that even if (16) makes $st(i,t)$ to be 0 or 1 when droplet d_i does not stall within time $t-1$ to t. Due to (14) and (15), the lower bound of the number of cells that must be deactivated in B is 11 when droplet d_i moves. Therefore, to satisfy (14), (15), and (18), the value of $st(i,t)$ is restricted to be 0 only.

9) *Activation sequence constraints:* Due to the electrode constraint, we obtain the electrode activation for each droplet at any time t. We use the following constraints to derive the global activation sequences for total movements control. Note that the "must" condition still holds. For the "must" be activated cells

$$A_1(x,y,t) = \sum_{d_i \in D} a_1(i,x,y,t) \quad (x,y) \in B, 0 \leq t \leq T_{\max}. \tag{19}$$

For the "must" be deactivated cells

$$A_0(x,y,t) \geq a_0(i,x,y,t) \quad \forall d_i \in D, (x,y) \in B, 0 \leq t \leq T_{\max} \tag{20}$$

$$A_0(x,y,t) \leq \sum_{d_i \in D} a_0(i,x,y,t) \quad (x,y) \in B, 0 \leq t \leq T_{\max}. \tag{21}$$

For the do not care cells (EC-Rule III)

$$A_0(x,y,t) + A_1(x,y,t) + A_X(x,y,t) = 1 \quad (x,y) \in B, 0 \leq t \leq T_{\max}. \tag{22}$$

Since the do not care term can be replaced by "1" or "0," we use the constraints to obtain all possible activation sequences of each cell as follows:

$$0 \cdot A_0(x,y,t) + 1 \cdot A_1(x,y,t) + 0 \cdot A_X(x,y,t) \leq as(x,y,t) \quad (x,y) \in B, 0 \leq t \leq T_{\max} \tag{23}$$

$$0 \cdot A_0(x,y,t) + 1 \cdot A_1(x,y,t) + 1 \cdot A_X(x,y,t) \geq as(x,y,t) \quad (x,y) \in B, 0 \leq t \leq T_{\max}. \tag{24}$$

10) *Broadcast constraints:* the three broadcast rules mentioned earlier can be represented as follows:

$$1 - cmp(x_1,y_1,x_2,y_2) \geq as(x_1,y_1,t) - as(x_2,y_2,t) \quad (x_1,y_1),(x_2,y_2) \in C, 0 \leq t \leq T_{\max} \tag{25}$$

$$1 - cmp(x_1,y_1,x_2,y_2) \geq as(x_2,y_2,t) - as(x_1,y_1,t) \quad (x_1,y_1),(x_2,y_2) \in C, 0 \leq t \leq T_{\max} \tag{26}$$

$$cp(x_1,y_1,p) + cp(x_2,y_2,p) \leq cmp(x_1,y_1,x_2,y_2) + 1 \quad (x_1,y_1),(x_2,y_2) \in C, 1 \leq p \leq P_{\max} \tag{27}$$

where (25) and (26) represent the BC-Rule I, and (27) represents the BC-Rules II and III. Note that if two cells (x_1, y_1) and (x_2, y_2) are compatible, the value of $cmp(x_1, y_1, x_2, y_2)$ can be 0 or 1, which still holds due to the BC-Rule III. The three constraints state the computation of minimized the number of control pins as follows:

$$\sum_{p=1}^{P_{\max}} cp(x,y,p) = uc(x,y), (x,y) \in C \tag{28}$$

$$cp(x,y,p) \leq up(p), (x,y) \in C, 1 \leq p \leq P_{\max} \tag{29}$$

$$up(p) \leq \sum_{(x,y)\in C} cp(x,y,p), 1 \leq p \leq P_{\max}. \tag{30}$$

Constraint (28) states that we should assign a pin to the cell which is used [20]. Constraints (29) and (30) state that if a cell (x, y) is controlled by a pin p, then p is used; otherwise p is un-used.

IV. Two-Stage ILP-Based Algorithm

Although the basic ILP formulations can optimally solve the droplet routing problem for PDMFBs, it is still limited in handling the dramatically growing complexity in practical bioassays. In this section, we propose a two-stage ILP-based droplet routing algorithm of global routing followed by incremental ILP-based routing for PDMFBs. We first overview our two-stage routing algorithm, and then detail each phase of our algorithm in the following subsections. Finally, we use an example to clarify the proposed algorithm.

A. Routing Algorithm Overview

Fig. 5 shows the overview of our two-stage ILP-based droplet routing algorithm. The essential intuition behind our algorithm is to reduce the complexity of the solution space in

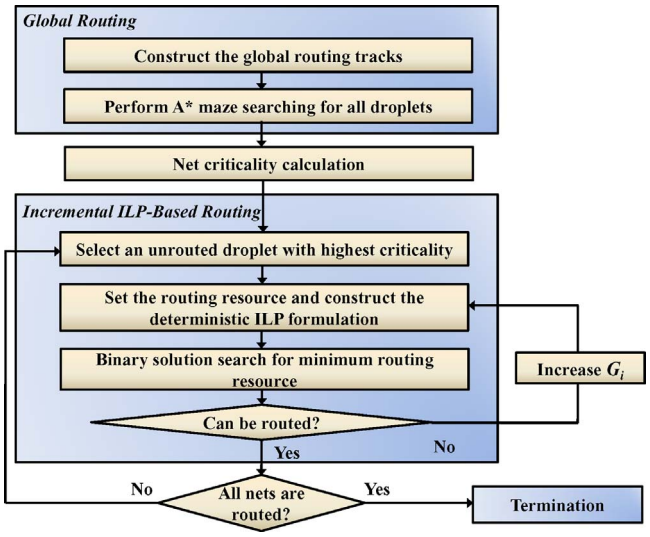

Fig. 5. Overview of our two-stage ILP-based droplet routing algorithm.

the basic ILP formulations by using a two-stage technique of global routing followed by incremental ILP-based routing.

The global routing stage first constructs the global routing tracks by analyzing the preferred moving direction of each droplet to guide the A* maze searching. Since droplets are recommended to route along the global routing tracks orderly, it can reduce the number of used cells and routing complexity. By performing A* maze routing for all droplets in global routing, the solution space for each droplet is reduced significantly from whole 2-D plane to a global routing path.

In net criticality calculation, we determine the criticality of each droplet. A droplet is said to be critical if it is difficult to route it, due to the severe interference with other droplets. This criticality information will be used in the incremental ILP-based routing stage.

Instead of considering all droplets at the same time, we propose an incremental ILP (IILP) approach to solve the routing problem in several manageable iterations to reduce the number of variables and constraints of the ILP formulations significantly. In each iteration, we select an un-routed droplet with the highest criticality, then route it with the previous routed droplets by solving the ILP formulations incrementally. Since searching a feasible solution is much faster than searching the optimal solution in a given ILP formulation, to further reduce the runtime, we propose a deterministic integer linear programming (DILP) formulation that casts the original routing optimization problem into a decision problem. The DILP will determine whether a feasible solution exists within given routing resources. To search a feasible solution in a decision problem efficiently, we perform a binary solution search method that searches it in logarithmic time. If this droplet cannot be routed, we will increase routing resources to improve the routability. Finally, iterations terminate until all droplets are routed.

B. Global Routing

The goal of global routing is to schedule the initial droplet routing paths to reduce the complexity of the solution space in the ILP formulations from whole 2-D plane to global routing paths. With the increased design complexities, any naïve routing path may violate the timing and fluidic constraints easily. Furthermore, if droplets route disorderly, a large number of cells and independent control pins will be used. Hence, the reliability and fault tolerance for bioassays will be significantly degraded. To overcome these drawbacks, we use the same concept of global routing tracks in [11]. We construct the global routing tracks with the preferred moving direction to derive an initial routing path on these tracks for each droplet. Due to the fluidic constraints, it is desirable to maintain a minimum space when droplets move on the microfluidic array. Therefore, the initial global routing tracks are constructed on nonadjacent rows and columns. Then we determine the preferred moving direction of these tracks by analyzing the preferred moving direction of each net. We define $pmdl_i(x, y)$, $pmdr_i(x, y)$, $pmdu_i(x, y)$, and $pmdd_i(x, y)$ to represent the cell (x, y) with the left, right, up, and down preferred moving directions, respectively, within the bounding box of net n_i. Note that we use the real bounding box computed by the maze routing algorithm. For each cell (x, y) in the bounding box of net n_i, there are two preferred moving directions which are determined by the coordinates of source and sink. Therefore, the preferred moving direction of global routing tracks can be defined as follows.

1) For tracks on rows (tr_j).
   If
   $$\sum_{(x,y)\in tr_j}\sum_{n_i\in N} pmdr_i(x, y) \geq \sum_{(x,y)\in tr_j}\sum_{n_i\in N} pmdl_i(x, y)$$
   the preferred moving direction is right; otherwise it is left.
2) For tracks on columns (tc_j).
   If
   $$\sum_{(x,y)\in tc_j}\sum_{n_i\in N} pmdu_i(x, y) \geq \sum_{(x,y)\in tc_j}\sum_{n_i\in N} pmdd_i(x, y)$$
   the preferred moving direction is up; otherwise it is down.

After that, we model the routing path of droplet d_i as $P_{d_i} = \{v_1, v_2, \ldots, v_n\}$ where each node v_i represents the cell used in microfluidic array, then apply A* maze searching to find a min-cost routing path for each droplet. Note that v_1 is the location of source and v_n is the location of sink. If droplet moves along the preferred moving direction from v_i to v_{i+1}, we assign the routing cost c_1; otherwise, we assign a higher routing cost c_2 for penalty. In this paper, we set c_1 and c_2 to be 1 and 3, respectively.

C. Net Criticality Calculation

A key issue in the droplet routing problem is the determination of the droplet routing order. Motivated by [11], we use the concept of net criticality to determine the routing order. A droplet d_i is said to be critical if d_i has fewer possible solutions (routing paths and schedules) due to the severe interference with other droplets or blockage cells. We use $crit(d_i)$ to denote the criticality of droplet d_i and $crit(d_i)$ is defined as follows:

$$crit(d_i) = \frac{(|E_b^i| + |E_s^i|) - |E_t^i|}{|BB_i|} \tag{31}$$

TABLE II
NOTATIONS USED IN OUR TWO-STAGE ILP-BASED DROPLET ROUTING ALGORITHM

G_i	Set of used cells in global routing path for droplet d_i
G'_i	Set of used cells by the previous routed droplet d_i
N	Netlist among all subproblems
$T^i_{\max}$	Maximum available routing time that can be used for routing droplet d_i
$P^i_{\max}$	Maximum available number of control pins that can be used for routing droplet d_i
T^i_l	Lower bound of $T^i_{\max}$
T^i_u	Upper bound of $T^i_{\max}$
P^i_l	Lower bound of $P^i_{\max}$
P^i_u	Upper bound of $T^i_{\max}$
M_i	Routing resources for droplet d_i to route with the previous routed droplets
IS	Increasing scalar of routing resources
BB_i	Set of available cells in bounding box of droplet d_i
E_b	Set of blockage cells
E_{s_i}	Set of available cells in the 3×3 area center by source s_i
E_{t_i}	Set of available cells in the 5×5 area center by target t_i

where

$$E^i_b = \{c | c \in E_b \cap BB_i\}$$
$$E^i_s = \{c | c \in E_{s_j} \cap BB_i, \forall d_j \in D/d_i\}$$
$$E^i_t = \{c | c \in E_{t_j} \cap BB_i, \forall d_j \in D/d_i\}.$$

The intuition behind the net criticality can be described as follows. Due to the blockage constraints and fluidic property, the blockage cells and source cells inside BB_i will have detrimental effects on the routability of droplet d_i. Similarly, when a droplet arrives at its target cell, this cell becomes a 3×3 blockage to avoid the unexpected mixing with other droplets. Let us consider the droplet d_i with many target cells inside its bounding box, it is hard to route d_i when other droplets are routed on these target cells. Droplet d_i thus has more routing solutions before these target cells are routed, implying less interference between d_i and others (note that all droplet are treated as un-routed in determining the criticality). As shown in (31), the larger the $crit(d_i)$ is, the more critical the d_i is. There are two major reasons: 1) as the numerator increases, the droplet d_i suffers from more interference with other nets and blockages, and 2) since the cells in BB_i are possible be used frequently for routing, as the denominator decreases, there are fewer routing solutions for d_i.

D. Incremental ILP-Based Routing

After the global routing stage, the solution space is reduced significantly from the whole 2-D plane to global routing paths. To further reduce the solution space that directly considers all droplets at the same time, the incremental ILP-based routing

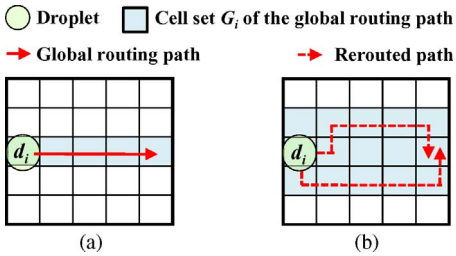

Fig. 6. (a) Initial cell set of the global routing path. (b) Enlarged cell set of the global routing path by one bounding box.

routes an un-routed droplet with the previous routed droplets incrementally. Thus, for an un-routed droplet d_i and a previous routed droplet d_j, we reformulate the ILP constraints by replacing the whole 2-D available cell set C with the cell set G_i which is used in its global routing path and the cell set G'_j which is used by the previous routed paths, respectively.

To further reduce the runtime, we cast the original optimization problem into a decision problem by solving the DILP formulation. In each iteration, we select an un-routed droplet with the highest criticality, then route it with the previous routed droplets by solving the DILP formulation incrementally. To search a feasible solution within minimal routing resources efficiently, we perform a binary solution search method that searches the feasibility in logarithmic time.

Although the above proposed method can solve the droplet routing problem in a reasonable runtime by global routing followed by incremental routing. However, as the increased design complexity of DMFBs, if the routing paths are restricted to the global routing paths, the freedom of droplets is also restricted, which may cause routability problem. Therefore, if we cannot route an un-routed droplet d_i with the previous routed droplets in the cell set G_i of global routing path, we increase the cell set G_i by one bounding box and reroute it. Fig. 6 illustrates this routing concern. Finally, the iteration terminates until all droplets are routed.

1) *DILP Formulation:* By global routing, the solution space is reduced from the whole 2-D plane to global routing paths. Specifically, the number of used cells in the original three optimizations (i.e., used cells, control pins, and routing time) has been determined. The objective now is scheduling the control pins and routing time with minimum requirements with respect to these global routing paths. However, directly minimizing the two objectives may still incur a high design complexity with runtime overhead. To further reduce the runtime, we propose a DILP formulation that casts the original routing optimization problem into a decision problem. We directly bound the maximum allowable routing time and control pins in the ILP formulation. In this regard, many redundant solutions can be avoided and thus the searching time for ILP can be reduced. When incrementally routing an un-routed droplet with previous routed droplets, we determine the minimum bounds of routing time and control pins such that all the ILP constraints can be satisfied. That is, the goal is minimizing the bounds of maximum allowable routing time and control pins so that this un-routed droplet can be successfully routed with previous routed droplets. Based on

these issues, the objective function can be redefined as follows:

$$Minimize : 1. \tag{32}$$

Instead of directly optimizing the original objective function within fixed-size $T_{\max}$ and $P_{\max}$, we try to determine the minimum bounds for incrementally routing an un-routed droplet with previous routed droplets. For an un-routed droplet d_i with the previous routed droplets, we define the routing resources for routing d_i as $T^i_{\max}$ and $P^i_{\max}$, representing the bounds of maximum allowable routing time and control pins, respectively. In formulating constraints, we replace the maximum allowable routing time $T_{\max}$ with $T^i_{\max}$ and replace the maximum allowable control pins $P_{\max}$ with $P^i_{\max}$, respectively. Since it is desirable to minimize the routing time and control pins for routing d_i with previous routed droplets, scheduling the two routing resources with minimum bounds for a feasible solution is the most important issue.

2) *Solution Search of DILP:* The key issue in the DILP formulation is to minimize the routing resources such that we can successfully route an un-routed droplet d_i with the previous routed droplets. Intuitively, a naïve approach is to exhaustively search all the permutations of routing resources among the range of [0, $T_{\max}$] and [0, $P_{\max}$]. This method is time-consuming due to the time complexity is $O(T_{\max} \times P_{\max})$. For example, given a droplet d_1 with $T_{\max} = 9$ and $P_{\max} = 9$. The exhaustively search requires $10 \times 10 = 100$ iterations (permutations of $T^1_{\max}$ and $P^1_{\max}$) in determining the feasible solution for routing d_1 with previous routed droplets. Furthermore, this kind of method also introduces many redundant searching iterations. For example, if we have known that the $T^1_{\max}$ must be larger than 4, all the permutations with respect to $T^1_{\max} = 0 \sim 4$ are not necessary. That is, a total of $5 \times 10 = 50$ searching iterations are redundant. To remedy these deficiencies, we use a linear combination of the two routing resources to be one single objective function M_i and define the increasing scalar to characterize the growth rate of the two routing resources. The routing resources M_i for routing an un-routed droplet d_i is defined as follows:

$$M_i = (T^i_l + \sigma_1 \cdot IS) + (P^i_l + \sigma_2 \cdot IS) \tag{33}$$

where IS is the *common* growth rate of the two routing resources $T^i_{\max}$ and $P^i_{\max}$ and both σ_1 and σ_2 are user specified constants. Note that IS is an integer. As the experimental setting, we set σ_1 and σ_2 to be 0.8 and 1, respectively. The objective goal is simply searching a minimum increasing scalar for a feasible solution instead of exhaustive permutations. Considering the previous example, all the required permutations of ($T^1_{\max}$, $P^1_{\max}$) are ranged from (0, 0), (0.8, 1), (1.6, 2), ..., (7.2, 9), (8, 9), (8.8, 9), (9, 9). Note that when one of the routing resources exceeds its maximum allowable value, it stops increasing. Compared with the exhaustive method with 100 searching iterations, only 13 iterations are required by finding the increasing scalar (from 0 to 12), which greatly reduces the searching time. Based on the definition, we have the following lemma.

Lemma 1: Given two increasing scalars IS_1 and IS_2 where $IS_1 < IS_2$. If droplet d_i can be routed with IS_1, then droplet d_i can be also routed with IS_2.

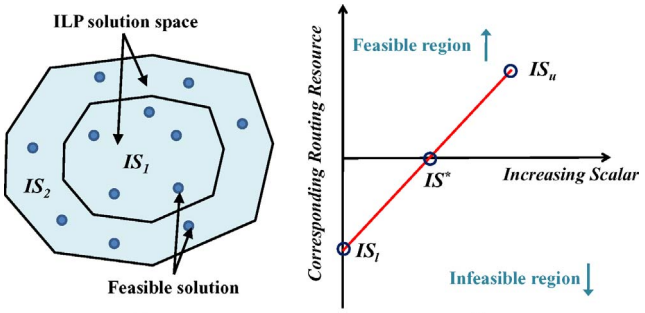

Fig. 7. (a) Solution space with larger IS includes the one with smaller IS. (b) Relationship between increasing scalar and feasibility of the DILP formulation.

Algorithm 1: Binary Search for IS^*

```
1  begin
2      //Set the lower bound and upper bound of routing resources
       ;
3      T_l^i ← max{G_i, T_max^j} ;
4      T_u^i ← T_max ;
5      P_l^i ← max{0, P_max^j} ;
6      P_u^i ← P_max ;
7      //Set the lower bound and upper bound of IS ;
8      IS_l ← 0 ;
9      IS_u ← max{(T_max − T_l^i)/σ1, (P_max − P_l^i)/σ2} ;
10     while IS_l < IS_u do
11         IS_m ← (IS_l + IS_u)/2 ;
12         Set the corresponding routing resources M_i with IS_m ;
13         if d_i can be routed then
14             IS_u = IS_m;
15         else
16             IS_l = IS_m + 1;
17         end
18     endw
19     return IS_u ;
20 end
```

Proof: Lemma 1 follows true since when the increasing scalar IS increases, the corresponding routing resources will increase monotonically. If we have found feasible routing resources that can route an un-routed droplet, increasing the routing resources only frees up more solution space of the ILP. Specifically, as shown in Fig. 7(a), if there exists a feasible solution with respect to IS_1, for any $IS_2 > IS_1$, the corresponding solution space must include such a feasible solution. ■

Fig. 7(b) demonstrates the essence of lemma 1: the *continuous* relationship of increasing scalar and feasibility of the DILP formulation. This feature of increasing scalar shows the capability of solving in logarithmic time by performing a binary search method. Therefore, to further reduce the searching iterations, we propose a binary solution search method to optimally search the minimum increasing scalar, denoted by IS^*, for routing resources to route the un-routed droplet d_i successfully.

Algorithm 1 shows our binary solution search method for IS^*. When routing an un-routed droplet d_i, we first set the lower and upper bound of routing resources as shown in line 2–6. Since we route each un-routed droplet with the previous routed droplet incrementally, and the cell set in DILP formulation is based on the global routing paths, we define the

lower bound T_l^i of $T_{\max}^i$ to be the maximum value between the cells in global routing path G_i and $T_{\max}^j$ found by the previous routed droplet d_j. We define the lower bound of $P_{\max}^i$ to be the maximum value between zero and $P_{\max}^j$ since there may exist a subproblem without any droplet to be routed in PDMFBs. The upper bound of $T_{\max}^i$ and $P_{\max}^i$ is equal to the original constraints of $T_{\max}^i$ and $P_{\max}^i$. The goal of our searching algorithm is to search the IS^* of M_i. By determining the lower and upper bound of routing resources, we can find the lower and upper bound of IS for M_i as shown in line 8–9. Then we perform the binary solution search method to find IS^* in line 10–18. For each searching iteration, the mean value $IS_m = (IS_l + IS_u)/2$ is used to formulate the DILP. Finally, the searching iterations terminate until we find IS^*.

By the binary solution searching algorithm, the complexity of iterations is reduced to $O(log(IS_u - IS_l))$. Compared with the exhaustively searching, the proposed algorithm reduces the runtime significantly.

E. Handling Three-Pin Nets

In handling a three-pin net, we need to merge the two droplets during their transportation. We decompose a three-pin net into two two-pin nets d_1 and d_2. In performing the merging operation, we sequentially route the two droplets by first routing d_1 to its sink location followed by routing d_2 to the sink location. Note that the fluidic constraints and electrode constraints are not considered between d_1 and d_2.

F. Exemplification

For purposes of clarity, we use an example to exemplify the proposed two-stage ILP-based routing, as illustrated in Fig. 8. As discussed before, the essence of our two-stage ILP-based routing can be summarized as two major parts: 1) in order to reduce the runtime complexity, we find a specific routing path for each droplet such that the solution space of the ILP formulation can be reduced from the entire 2-D plane to the path, and 2) since searching a feasible solution is much faster than searching the optimal solution in a given ILP formulation, we propose a deterministic routing strategy to achieve further runtime reduction.

Given a 2-D digital microfluidic array with three nets as shown in Fig. 8(a). In the first routing stage, we construct the global routing tracks on nonadjacent rows and columns by analyzing the preferred moving direction of each droplet. For example, the row 1 in Fig. 8(a) will be horizontally affected by droplets d_2 and d_3. Since the preferred moving direction of d_2 is from the upper-left to the lower right and the preferred moving direction of d_3 is from the lower left to the upper right, there are a total of seven cells with preferred moving direction by right (i.e., $pmdr_2(1,1)$, $pmdr_2(1,2)$, $pmdr_2(1,3)$, $pmdr_2(1,4)$, $pmdr_2(1,5)$, $pmdr_3(1,6)$, $pmdr_3(1,7)$). Therefore, the global routing tracks on row 1 is assigned by right direction. By adopting this calculation on other rows and columns, the entire global routing tracks and corresponding directions can be constructed, as shown in Fig. 8(b).

Since it is desirable to reduce the number of used cells and routing complexity, an A* searching method is adopted to guide these droplets along the global routing tracks orderly, as the arrow lines illustrated in Fig. 8(c). After that, we determine the routing priority by calculating the net criticality [through (31)] of each of the three nets. For example, for droplet d_1, the values of BB_1, E_b^1, E_s^1, and E_t^1 are 16, 2, 3, 0, respectively. The criticality $crit(d_1)$ of droplet d_1 can be calculated as $((2+3)-0)/16$. Similarly, $crit(d_2)$ and $crit(d_3)$ can be obtained by $((0+0)-3)/15$ and $((2+3)-10)/18$. Since in this paper a more critical net has higher routing priority, the routing order is determined by d_1, d_2, and d_3.

In the second routing stage, instead of concurrently solving all droplet routing, we use an incremental ILP approach to iteratively perform the routing such that the design complexity can be reduced. In the first iteration, the droplet d_1 with the highest net criticality will be selected first for routing. Since the routing path of d_1 has been found in the first routing stage (global routing), the values of T_l^1 and P_l^1 will be set to be 9 (equal to the $|G_1|$) and 0, respectively; the values of T_u^1 and P_u^1 depend on the bioassay (given as input values). In this case, T_u^1 is set to be 20 and P_u^1 is set to be 10. Then, we construct the deterministic ILP formulation to cast the original routing optimization problem into a decision problem. That is, the objective is now searching the minimum bounds of routing time ($9 \sim 20$) and required control pins ($0 \sim 10$) such that the droplet d_1 can be routed. As discussed before, an exhaustive search requires $(20-9+1)\times(10-0+1) = 132$ iterations, which incurs a runtime overhead. Therefore, we define the routing resources M_1 of droplet d_1 as $(9+0.8\times IS) + (0+1\times IS)$ and search the minimum IS to resolve this problem. By performing the binary search and formulating the routing into DILP, a feasible solution is obtained with IS^* being equal to 4. That is, the bounds of routing time and control pins are 12 and 4. We trace the actual routing time and control pins from the variables in the DILP solution and obtain the two values by 9 and 4, as shown in Fig. 8(d). In the same way, after droplet d_1 is routed, we select d_2 and route it with the previous routed droplet d_1 by adopting the same procedure, as shown in Fig. 8(e). Finally, iterations terminate until all droplets are routed [see Fig. 8(f)].

V. Complexity Analysis

We assume the width and height of the given chip are W and H. As the basic ILP formulation discussed in Section III, the number of variables are bounded by the following three major types of variables. First, for each droplet, four variables are used to represent its location at each time cycle. Second, four variables are used to represent the compatibilities between activation sequences. Third, for each cell, three variables are used to represent the pin assigned on it. Therefore, the number of variables is $O(|D|WHT_{\max} + W^2H^2 + WHP_{\max})$. Similarly, the number of constraints are bounded by the fluidic constraints and broadcast constraints. Therefore, the number of constraints is $O(|D|^2WHT_{\max} + W^2H^2T_{\max} + W^2H^2P_{\max})$. In the DILP formulation, the solution space is reduced from the whole 2-D plane to global routing paths. Since the DILP formulation is based on an incremental routing strategy, only one selected droplet (un-routed droplet) is needed to be considered. Therefore, the number of variables and constraints are $O((W+H)T_{\max} + (W+H)^2 + (W+H)P_{\max})$ and $O(|D|(W+H)T_{\max}+(W+H)^2T_{\max}+(W+H)^2P_{\max})$, respectively.

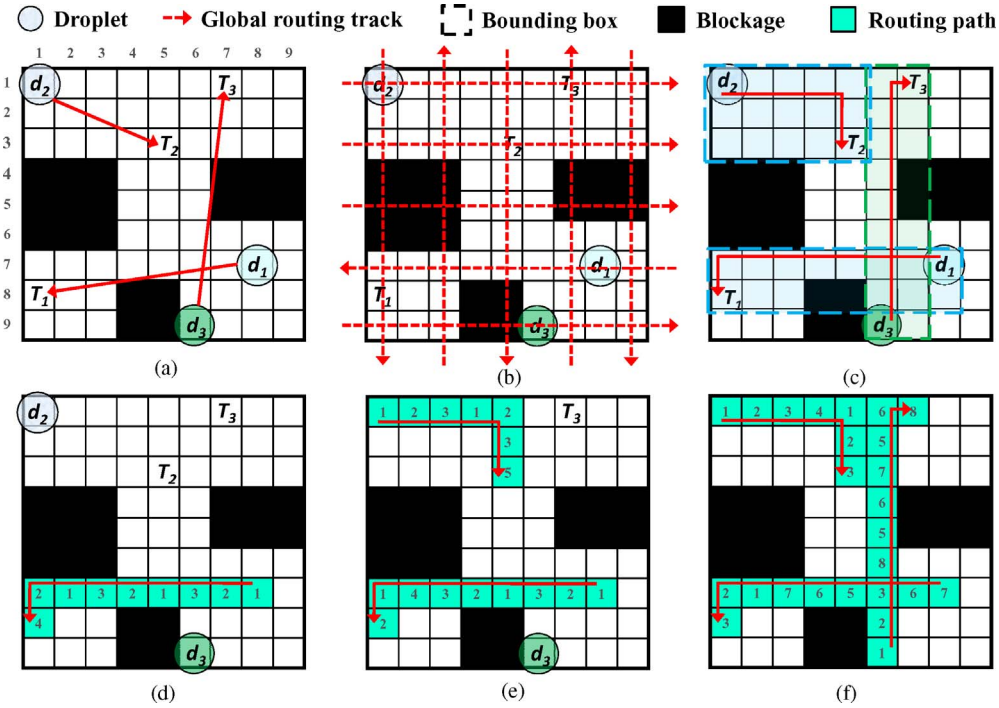

Fig. 8. This example describes the proposed two-stage ILP-based routing algorithm. (a) DMFB with three nets (droplets d_1, d_2, and d_3). (b) Global routing tracks construction. (c) Adopt the A* maze searching to route droplets along the preferred routing tracks. Then, we determine the routing priority by calculating the criticality of each net. (d)–(f) Incremental ILP-based routing. (d) Successfully route droplet d_1 with four control pins and nine time cycles. (e) Incrementally route droplet d_2 with the previous routed droplet d_1 with five control pins and 10 time cycles. (f) Incrementally route droplet d_3 with previous routed droplets d_1 and d_2 with eight control pins and 13 time cycles.

TABLE III
STATISTICS OF THE ROUTING BENCHMARKS

Benchmark	Size	#Sub	$T_{\max}$	#Net	$\#D_{\max}$
vitro-1	16 × 16	11	20	28	5
vitro-2	14 × 14	15	20	35	6
protein-1	21 × 21	64	20	181	6
protein-2	13 × 13	78	20	178	6

Size: size of the microfluidic array.#Sub: number of subproblems.
$T_{\max}$: timing constraint.#Net: total number of nets.
$\#D_{\max}$: maximum number of droplets among subproblems.

VI. EXPERIMENTAL RESULTS

Our two-stage ILP-based droplet routing algorithm was implemented in the C++ language and ran on a 2 GHz 64-bit Linux machine with 16 GB memory, and GLPK [1] was used as our ILP solver. We evaluated all routing algorithms on the two practical bioassays used in the previous work [22]: the *in-vitro* diagnostics and the colorimetric protein assay. Table III shows the statistics of each benchmark. To show the effectiveness and the robustness of our algorithm, we conducted three experiments for the number of control pins, the number of used cells, and the droplet routing time among the direct-addressing ([22], [8]), the broadcast-addressing scheme([22] + [20] and [8] + [20]), and ours ([22] + IILP, [8] + IILP, and two-stage ILP) in Tables IV–VI. In [22] + IILP and [8] + IILP, we replace our global routing results with the routing paths found in [22] and [8]. Note that the four benchmarks are derived from the placement of previous DMFB design flow. A series of 2-D placement configurations of fluidic modules in different time intervals are obtained in the placement stage. Therefore, the droplet routing problem is decomposed into a series of subproblems corresponding to different reactions. More details can be referred to [14], [15]. For fair comparison with [22] and [8], we compare the maximum and average values among all subproblems. We also compare the runtime between [22] + IILP, [8] + IILP, and our two-stage ILP to show the effectiveness and efficiency of the proposed method in Table VII.

For the first experiment, we compared the number of control pins among different schemes as listed in Table IV. Compared with the direct-addressing scheme ([22], [8]), the respective maximum and average number of control pins among all subproblems are (4.53×, 3.82×) and (4.44×, 4.03×) of our algorithm. Compared with the broadcast-addressing scheme ([22] + [20], [8] + [20]), the respective maximum and average number of control pins among all subproblems are (1.74×, 1.90×) and (1.78×, 2.06×) of our algorithm. To demonstrate the effectiveness of our IILP formulations, compared with the integrated scheme ([22] + IILP, [8] + IILP), the respective maximum and average number of control pins among all subproblems are (1.32×, 1.73×) and (1.54×, 1.83×) of our algorithm.

In the second experiment, we compared the number of used cells among different schemes as listed in Table V. For the direct-addressing scheme ([22], [8]), the number of used cells among all subproblems are (1.02×, 1.07×) of our algorithm. Since the broadcast-addressing scheme is directly applied to the direct-addressing-based routing result, the number of used cells are the same with the direct-addressing scheme.

TABLE IV
COMPARISONS FOR THE NUMBER OF PINS

Benchmark	Direct Addressing				Broadcast Addressing				Two-Stage ILP					
	[22]		[8]		[22] + [20]		[8] + [20]		[22] + IILP		[8] + IILP		Ours	
	$P_{\max}$	P_{avg}	$P_{\max}$	P_{avg}	$P_{\max}$	P_{avg}	$P_{\max}$	P_{avg}	$P_{\max}$	P_{avg}	$P_{\max}$	P_{avg}	$P_{\max}$	P_{avg}
vitro-1	45	21.55	50	23.45	21	9.48	22	10.11	15	9.11	18	9.49	13	4.51
vitro-2	50	15.73	42	16.40	21	8.95	24	10.64	17	8.03	17	9.21	12	5.01
protein-1	67	25.83	75	26.38	18	9.52	18	10.55	14	8.54	15	9.25	12	5.43
protein-2	54	12.03	46	12.35	23	8.73	21	8.55	17	7.72	23	7.38	11	4.43
Avg.	4.53	3.82	4.44	4.03	1.74	1.90	1.78	2.06	1.32	1.73	1.54	1.83	1	1

$P_{\max}$: maximum number of control pins among all subproblems. P_{avg}: average number of control pins among all subproblems.
Avg.: average comparison of number of control pins.

TABLE V
COMPARISONS FOR THE NUMBER OF USED CELLS

Benchmark	Direct Addressing		Broadcast Addressing		Two-Stage ILP		
	[22]	[8]	[22] + [20]	[8] + [20]	[22] + IILP	[8] + IILP	Ours
	U.C.	U.C.	U.C.	U.C.	U.C.	U.C.	U.C.
vitro-1	237	258	237	258	231	243	231
vitro-2	236	246	236	246	231	229	229
protein-1	1618	1688	1618	1688	1597	1627	1582
protein-2	939	963	939	963	927	943	930
Avg.	1.02	1.07	1.02	1.07	1.00	1.02	1

U.C.: total number of used cells. Avg.: average comparison of number of used cells.

TABLE VI
COMPARISONS FOR THE DROPLET ROUTING TIME

Benchmark	Direct Addressing				Broadcast Addressing				Two-Stage ILP					
	[22]		[8]		[22] + [20]		[8] + [20]		[22] + IILP		[8] + IILP		Ours	
	$T_{\max}$	T_{avg}	$T_{\max}$	T_{avg}	$T_{\max}$	T_{avg}	$T_{\max}$	T_{avg}	$T_{\max}$	T_{avg}	$T_{\max}$	T_{avg}	$T_{\max}$	T_{avg}
vitro-1	20	13.00	19	14.30	20	13.00	19	14.30	19	12.47	19	13.55	18	12.41
vitro-2	17	11.33	20	12.00	17	11.33	20	12.00	17	11.01	17	11.48	18	10.46
protein-1	20	16.31	20	16.55	20	16.31	20	16.55	20	16.08	20	15.44	20	15.42
protein-2	20	10.51	20	12.19	20	10.51	20	12.19	20	10.33	20	11.52	20	10.22
Avg.	1.01	1.05	1.04	1.14	1.01	1.05	1.04	1.14	1.00	1.03	1.00	1.08	1	1

$T_{\max}$: maximum routing time among all subproblems. T_{avg}: average droplet routing time among all subproblems.
Avg.: average comparison of droplet routing time among all subproblems.

Compared with the integrated scheme ([22] + IILP, [8] + IILP), the number of used cells among all subproblems are (1.00×, 1.02×) of our algorithm.

In the third experiment, we compared the droplet routing time among different schemes as listed in Table VI. Compared with the direct-addressing scheme ([22], [8]), the respective maximum and average droplet routing time among all subproblems are (1.01×, 1.05×) and (1.04×, 1.14×) of our algorithm. Since the broadcast-addressing scheme is directly applied to the direct-addressing-based routing result, the statistics of the droplet routing time are the same with the direct-addressing scheme. To demonstrate the effectiveness of our IILP formulations, compared with the integrated scheme ([22] + IILP, [8] + IILP), the average droplet routing time among all subproblems are (1.03×, 1.08×) of our algorithm.

Table VII shows the runtime comparison among the basic ILP, direct-addressing + IILP, and our two-stage ILP algorithm. For the basic ILP, the problem instance is whole 2-D plane and it solves all the droplets simultaneously. For our two-stage ILP, the problem instance is reduced to global routing paths and the droplets are routed in incremental manner that reduce the solution space significantly. The results show that the basic ILP needs at least five days to solve all 2-D planes

TABLE VII
COMPARISONS FOR THE RUNTIME

Benchmark	Basic ILP	[22] + IILP	[8] + IILP	Ours
	CPU (min)	CPU (s)	CPU (s)	CPU (s)
vitro-1	> 7200	14.33	15.31	10.11
vitro-2	> 7200	16.49	18.38	8.32
protein-1	> 7200	28.43	34.51	30.13
protein-2	> 7200	22.16	28.38	21.38
Avg.	N.C.	1.34	1.55	1

N.C.: noncomparable. Avg.: average comparison of runtime.

of one benchmark, which is not feasible for this problem; in contrast, our two-stage ILP algorithm needs at most 30.13 s due to the significantly smaller solution space. To demonstrate the effectiveness of our IILP formulations, compared with the integrated scheme ([22] + IILP, [8] + IILP), our algorithm reduced the runtime by about (34%, 55%).

Based on the evaluation of four experiments, our two-stage ILP-based droplet routing algorithm achieves the best result of the number of control pins (an improvement by at most 4.53×), the number of used cells (an improvement by at most 1.07×), and the droplet routing time (an improvement by at most 1.14×) over the existing algorithms within reasonable CPU times. The experimental results demonstrate that our algorithm is very effective for droplet routing on PDMFBs.

VII. Conclusion

In this paper, we proposed the *first* droplet routing algorithm that considers the droplet routing and the broadcast-addressing scheme simultaneously for PDMFBs. We first presented a basic ILP formulation to optimally solve the droplet routing problem with simultaneously minimizing the number of control pins, the number of used cells, and the droplet routing time. Due to its complexity, we also proposed a two-stage technique of global routing followed by incremental ILP-based routing. To further reduce the runtime, we presented a *deterministic* ILP formulation that casts the original routing optimization problem into a decision problem, and then solves it by a binary solution search method that searches in logarithmic time. Extensive experiments demonstrate that in terms of the number of the control pins, the number of the used cells and the droplet routing time, we acquire much better achievement than all the state-of-the-art algorithms in any aspect.

References

[1] GNU Operating System. (2008, Oct. 16). *GLPK (GNU Linear Programming Kit)* [Online]. Available: http://www.gnu.org/software/glpk

[2] Silicon Biosystems [Online]. Available: http://www.siliconbiosystems.com

[3] K. F. Böhringer, "Modeling and controlling parallel tasks in droplet based microfluidic systems," *IEEE Trans. Comput.-Aided Des.*, vol. 25, no. 2, pp. 334–344, Feb. 2006.

[4] K. Chakrabarty, "Toward fault-tolerant digital microfluidic lab-on-chip: Defects, fault modeling, testing, and reconfiguration," in *Proc. IEEE Int. Conf. BioCAS*, Nov. 2008, pp. 329–332.

[5] K. Chakrabarty, "Design automation and test solutions for digital microfluidic biochips," *IEEE Trans. Circuits Syst.*, vol. 57, no. 1, pp. 4–17, Jan. 2010.

[6] K. Chakrabarty and T. Xu, *Digital Microfluidic Biochips: Design and Optimization*. Boca Raton, FL: CRC Press, 2010.

[7] S. K. Cho, H. Moon, and C. J. Kim, "Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits," *J. Microelectromechanic. Syst.*, vol. 27, no. 10, pp. 1714–1724, 2008.

[8] M. Cho and D. Z. Pan, "A high-performance droplet routing algorithm for digital microfluidic biochips," *IEEE Trans. Comput.-Aided Des.*, vol. 27, no. 10, pp. 1714–1724, Oct. 2008.

[9] S. K. Fan, C. Hashi, and C. J. Kim, "Manipulation of multiple droplets on $N \times M$ grid by cross-reference EWOD driving scheme and pressure contact packaging," in *Proc. MEMS Conf.*, 2003, pp. 694–697.

[10] E. J. Griffith, S. Akella, and M. K. Goldberg, "Performance characterization of a reconfigurable planar-array digital microfluidic system," *IEEE Trans. Comput.-Aided Des.*, vol. 25, no. 2, pp. 345–357, Feb. 2006.

[11] T.-W. Huang, C.-H. Lin, and T.-Y. Ho, "A contamination aware droplet routing algorithm for digital microfluidic biochips," in *Proc. IEEE/ACM ICCAD*, Nov. 2009, pp. 151–156.

[12] C. C.-Y. Lin and Y.-W. Chang, "ILP-based pin-count aware design methodology for microfluidic biochips," *IEEE Trans. Comput.-Aided Des.*, vol. 29, no. 9, pp. 1315–1327, Sep. 2010.

[13] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," *Int. J. Rob. Res.*, vol. 24, no. 4, pp. 295–310, 2005.

[14] F. Su and K. Chakrabarty, "Design of fault-tolerant and dynamically-reconfigurable microfluidic biochips," in *Proc. IEEE/ACM DATE*, Mar. 2005, pp. 1202–1207.

[15] F. Su, K. Chakrabarty, and R. B. Fair, "Microfluidics based biochips: Technology issues, implementation platforms, and design-automation challenges," *IEEE Trans. Comput.-Aided Des.*, vol. 25, no. 2, pp. 211–223, Feb. 2006.

[16] F. Su, W. Hwang, and K. Chakrabarty, "Droplet routing in the synthesis of digital microfluidic biochips," in *Proc. IEEE/ACM DATE*, Mar. 2006, pp. 1–6.

[17] T. Xu and K. Chakrabarty, "Droplet-trace-based array partitioning and a pin assignment algorithm for the automated design of digital microfluidic biochips," in *Proc. IEEE/ACM CODES+ISSS*, Oct. 2006, pp. 112–117.

[18] T. Xu and K. Chakrabarty, "Integrated droplet routing in the synthesis of microfluidic biochips," in *Proc. IEEE/ACM DAC*, Jun. 2007 pp. 948–953.

[19] T. Xu, W. Hwang, F. Su, and K. Chakrabarty, "Automated design of pin-constrained digital microfluidic biochips under droplet-interference constraints," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 3, no. 3, pp. 14.1–14.23, Nov. 2007.

[20] T. Xu and K. Chakrabarty, "Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips," in *Proc. IEEE/ACM DAC*, Jun. 2008, pp. 173–178.

[21] T. Xu and K. Chakrabarty, "A droplet-manipulation method for achieving high-throughput in cross-referencing-based digital microfluidic biochips," *IEEE Technol. Comput.-Aided Des.*, vol. 27, no. 11, pp. 1905–1917, Nov. 2008.

[22] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang, "BioRoute: A network-flow based routing algorithm for digital microfluidic biochips," in *Proc. IEEE/ACM ICCAD*, Nov. 2007, pp. 752–757.

[23] Y. Zhao, R. Sturmer, K. Chakrabarty, and V. K. Pamula, "Synchronization of concurrently-implemented fluidic operations in pin-constrained digital microfluidic biochips," in *Proc. IEEE Int. Conf. VLSI Des.*, Jan. 2010, pp. 69–74.

Tsung-Wei Huang is 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 automation for biochips.

Mr. Huang received first place in the ACM SIGDA Student Research Competition in 2010.

Tsung-Yi Ho (M'08) received the M.E. degree in computer science from National Chiao-Tung University, Hsinchu, Taiwan, in 2001, and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2005.

From 2003 to 2004, he was a Visiting Scholar with the University of California, Santa Barbara. In 2005, he was with Waseda University, Tokyo, Japan, and in 2008, he was with Synopsys, Mountain View, CA, respectively. 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 Assistant Professor. His current research interests include physical design automation for nanometer integrated circuits and biochips.