# A SAT-Based Routing Algorithm for Cross-Referencing Biochips

Ping-Hung Yuh*, Cliff Chiung-Yu Lin†, Tsung-Wei Huang‡, Tsung-Yi Ho‡, Chia-Lin Yang§, Yao-Wen Chang††
*Taiwan Semiconductor Manufacturing Company, Taiwan
†Department of Electrical Engineering, Stanford University, Stanford, CA
‡Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan
§Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
††Department of Electrical Engineering and the Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan

***Abstract*—CAD problems for microfluidic biochips have recently gained much attention. One critical issue is the droplet routing problem. On *cross-referencing* biochips, the routing problem requires an efficient way to tackle the complexity of simultaneous droplet routing, scheduling and voltage assignment. In this paper, we present the *first* SAT based routing algorithm for droplet routing on cross-referencing biochips. The SAT-based technique solves a large problem size much more efficiently than a generic ILP formulation. We adopt a two-stage technique of global routing followed by detailed routing. In global routing, we iteratively route a set of nets that heavily interfere with each other. In detailed routing, we adopt a negotiation based routing algorithm and the droplet routing information obtained in the global routing stage is utilized for routing decision.. The experimental results demonstrate the efficiency and effectiveness of the proposed SAT-based routing algorithm on a set of practical bioassays.**

## I. Introduction

Due to the advances of microfabrication, there are many developments in the microfluidic technology [11]. Various conventional laboratory procedures can now be performed on microfluidic biochips with advantages such as the reduced sample/reagent volume, the shorter assay completion time, and the improved portability. As a result, microfluidic biochips are getting their popularity in molecular biology.

Lately, the *second-generation* (digital) microfluidic biochips have been proposed [9]. On digital microfluidic biochips, *droplets* are controlled by the electrohydrodynamic forces generated by a 2D array of electrodes to achieve different droplet manipulations. Under the control of the electrodes, the droplets can move anywhere in the 2D array to perform the desired reactions. On a same biochip, the control sequences for the electrodes can be reconfigured for different bioassays. This is impossible on the *first-generation* (analog) microfluidic biochips, which manipulate continuous liquid using fixed micropumps.

A typical digital microfluidic biochip contains a 2D array of cells for droplet movement. The simplest control scheme is to connect each electrode with a dedicated control pin, and thus each electrode can be individually controlled. In this paper, we refer to this type of biochips as *direct-addressing biochips*. Such a control scheme provides the highest flexibility for droplet movement. The only limit on the droplet movement is the *fluidic constraint* that prevents unexpected merging between distinct droplets. However, the number of control wires increases dramatically with the system complexity, and raises the production cost and control wiring complexity. Therefore, this architecture is only suitable for small-scale biochips [12].

To overcome this limitation, a new digital microfluidic biochip architecture has been proposed [5] recently. The architecture, referred to as *cross-referencing biochips*, uses a row/column addressing scheme for droplet movement, where a row/column of electrodes are connected to a control pin. In this way, the number of control pins can be greatly reduced. However, the major limitation of this architecture is the electrode interference problem, where a row/column of electrodes can potentially affect the movement of all droplets in the same row/column. This problem incurs a higher complexity for droplet movement than that of direct-addressing biochips, and leads to performance degradation. Note that the electrode interference problem is a global effect that affects all droplets in the same row/column. Therefore, we cannot naively extend the previous routing algorithms on the direct-addressing biochips [3], [4], [6], [10], [15] to solve the droplet routing problem on cross-referencing biochips.

In this paper, we deal with the droplet routing problem on cross-referencing biochips. The main challenge is to ensure the correct droplet movement under the constraint imposed by the fluidic property of droplets, which prevents unexpected mixing among droplets, and the electrode activation problem, which prevents multiple droplets to move simultaneously. The goal of droplet routing is to minimize the maximum droplet transportation time for fast bioassay execution and better integrity.

### A. Previous Work

There are several works that handle the droplet routing problem in the literature [3], [4], [6], [10], [15], and all of them focus on direct-addressing biochips. However, due to the electrode interference problem, their method cannot be applied directly. Recently, several droplet manipulation methods have been proposed for cross-referencing biochips [6], [13], [14]. Griffith *et al.* [6] and Xu *et al.* [13] both proposed a graph-based method for droplet manipulation that based on given routing solutions for direct-addressing biochips. However, the initial routing solutions are not directly optimized for the cross-referencing architecture. Yuh *et al.* [14] proposed the first routing algorithm that directly targets at cross-referencing biochips. They proposed a progressive integer linear programming (ILP) based formulation that determines the droplet position progressively, one time step at a time. Therefore, their algorithm is not able to consider the routing information of other droplets when routing a droplet. This may lead to unnecessary droplet movement or detours, which increase the maximum droplet transportation time.

### B. Our Contribution

In this paper, we propose the *first* Boolean Satisfiability (SAT) based routing algorithm for the droplet routing on cross-referencing biochips. We adopt the SAT-based approach in our droplet routing algorithm regarding the nature of the problem and the recent advances of SAT solvers. The droplet routing problem, which will be formulated in Section III, involves only boolean variables in its constraints and is naturally more suitable to be solved as a boolean satisfiability problem. On the other hand, current state-of-art SAT solvers can solve problem instances with tens of thousands of variables and millions of clauses [2] and are much more efficient in handling large problem size when compared to ILP solvers.

Furthermore, in order to utilize the routing information of other droplets when routing a droplet, we take a two-stage approach of global routing followed by detailed routing. In global routing, we iteratively identify a set of nets that heavily interfere with each other and find the optimal routing path and schedule the selected nets based on a SAT formulation. In detailed routing, we adopt a negotiation based method to simultaneously perform droplet routing, scheduling and voltage assignment with the aid of a 3D routing graph, which is used to offer higher flexibility. In this stage, the droplet routing information obtained in the global routing stage is utilized for routing decision. Our method

is also capable of handling multi-pin nets for efficient mix operations. Followings are the major contributions of this paper:

- To reflect the nature of the problem, and to tackle the high routing complexity of cross-reference biochips, we propose the *first* SAT-based routing algorithm for cross-referencing biochips. SAT-based techniques are more efficient than generic ILP formulations, therefore, they are more suitable of handling the complexity of the routing problem of cross-reference biochips, which requires simultaneous droplet routing, scheduling and voltage assignment.
- Unlike [14] which does not have the routing information of other droplets when routing a droplet, our two-stage routing method allows the routing information obtained from the first-stage global routing to be utilized in detailed routing for routing decision. This avoids unnecessary droplet movement or detours thereby achieving better solution quality.
- In detailed routing, we use a 3D routing graph, which allows the droplets to move back and forth in different time spans, and thus offers a higher flexibility for droplet movement than that in the 2D one, which implicitly implies that one droplet can visit one basic cell at most once. This flexibility is important for droplet routing on cross-referencing biochips due to the high complexity of droplet movement and the need for concessions between droplets.

Experimental results demonstrate the effectiveness of the proposed routing scheme compared with previous indirect and direct approaches. For example, the proposed routing algorithm obtains smaller maximum droplet transportation time (19 cycles vs. 24 cycles) within reasonable CPU time for the *in-vitro* diagnostics, compared with the progressive-ILP based routing scheme [14]. Our algorithm also obtains much better solution compared with indirect algorithms which require a direct-addressing routing solution.

The remainder of this paper is organized as follows. Section II describes the routing concerns on cross-referencing biochips and the problem formulation. Section III presents the proposed routing algorithm. Section IV shows our experimental results. Finally, concluding remarks are provided in Section V.

## II. Routing on Cross-Referencing Biochips

In this section, we first introduce the architecture of cross-referencing biochips. Then we describe the routing constraints for droplet routing on cross-referencing biochips. Finally, we present the problem formulation for the droplet routing problem.

### A. Cross-Referencing Biochips

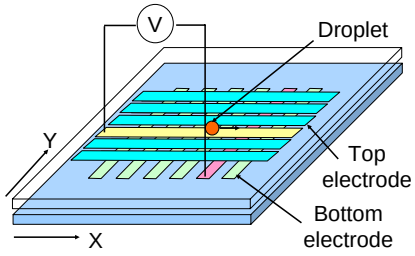


Figure 1. Top view of a cross-referencing biochip.

Figure 1 shows the top view of a cross-referencing biochip. A droplet is sandwiched by two plates. A set of orthogonal electrodes span a full row in the $X$-direction (the top plate) and a full column in the $Y$-direction (the bottom plate). Each set of the electrodes are assigned either a driving or reference voltage for droplet movement. A droplet can move to a basic cell (*i.e.*, unit cells for droplet movement) only when the basic cell is "activated"; i.e., there is a voltage difference between the upper and lower electrodes.

The advantage of this architecture is twofold [5]. First, we do not need a multi-layer architecture for electrodes. Second, the number of control wires for activating electrodes can be greatly reduced - it needs $W + H$ instead of $WH$ control wires, where $W$ ($H$) is the width (height) of a biochip in terms of the number of electrodes in one dimension.

### B. Routing Constraints

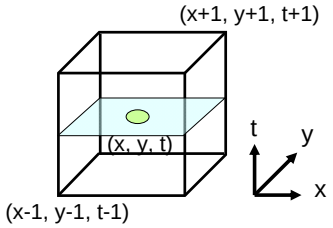


Figure 2. Illustration of the fluidic constraints.

There are two routing constraints in droplet routing: the *fluidic constraint* [10], [15] and the *electrode constraint* [14]. Both constraints are used to guarantee the correct droplet movement on cross-referencing biochips. The fluidic constraint says that if a droplet $d$ uses a basic cell $(x, y)$ at time $t$ for routing, then no other droplets can use the eight neighboring cells of $(x, y)$ from times $t - 1$ to time $t + 1$ for routing. Therefore, the fluidic constraints can be modeled as a 3D cube in the 3D space shown in Figure 2 [15], where the $X$ and $Y$ dimensions represent the biochip width and height, while the $T$ dimension represents the droplet transportation time. Under this 3D models, to satisfy the fluidic constraints, no other droplet is allowed in the 3D cube defined by $d$.

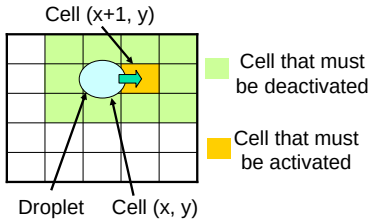


Figure 3. The modeling of the electrode constraint.

When moving multiple droplets in a cross-referencing biochip, the assignments of voltage in rows/columns may cause unwanted or incorrect droplet movement, since a single row/column can be assigned only one voltage (high, low, or floating) at a time [13], [14]. This scenario is called *electrode interference*, and it must be avoided during droplet transportation. We refer to the restriction used to avoid the electrode interference as the *electrode constraint*. Figure 3 shows how we model the electrode constraint [14]. If a droplet $d$ moves to cell $(x+1, y)$ from cell $(x, y)$ at time $t$, then the eight neighboring cells of $(x + 1, y)$ and $(x, y)$; i.e., the green cells in Figure 3, must be deactivated, with only the cell $(x + 1, y)$ activated for droplet movement. If $d$ stays at its original location instead, then the eight neighboring cells must be deactivated, but the underlying cell may or may not need to be activated. To make sure that these specific cells are deactivated, restrictions must be applied on the electrodes on the same row/column.

### C. Problem Formulation

In this paper, we deal with the droplet routing problem on cross-referencing biochips. We use the same partitioning method in [10], [15], in which the bioassay reactions are divided into a set of 2D planes based on the time they are scheduled. Since the reactions on different planes happens in different time slots, the routing problem can be solved independently. Thus, in this subsection, we focus our problem formulation on the subproblem on one 2D plane at a time.

Several issues should be considered about the droplet transportation. Besides the fluidic and electrode constraints, the problem formulation takes obstacles and 3-pin nets into account. When we are routing the droplets on a 2D plane, the modules being used for reactions and the

TABLE I
THE NOTATIONS USED IN OUR ALGORITHM.

| | |
|---|---|
| $N$ ($N'$) | set of all (3-pin) nets |
| $D$ | set of droplets |
| $d_j^i$ | $j$-th droplet of net $n_i$; $j = \{1, 2\}$ |
| $C$ ($\hat{C}$) | set of basic (global) cells |
| $(x, y, t)$ | a basic cell $(x, y)$ at time $t$ |
| $U_f(x, y, t)$ | s$\{(x', y', t')\|\|x - x'\| \leq 1, \|y - y'\| \leq 1, \|t - t'\| \leq 1\}$ |
| $U_e(x'y', x, y, t)$ | set of eight neighboring basic cells of $(x', y')$ and $(x, y)$, except $(x, y)$, at time $t$ |
| $(x_g, y_g, t_g)$ | a global cell $(x_g, y_g)$ at time $t_g$ |
| $(s_{g,x}^{i,j}, s_{g,y}^{i,j})$ | global cell location of the source of $d_j^i$ |
| $(\hat{t}_{g,x}^i, \hat{t}_{g,y}^i)$ | global cell location of the sink of net $n_i$ |
| $E_x^g/E_y^g$ | set of global cells with the same $x$- or $y$-coordinate as $(x_g, y_g)$ except $(x_g, y_g)$ |
| $E_x^b/E_y^b$ | set of basic cells with the same $x$- or $y$-coordinate as $(x, y)$ except $(x, y)$ |
| $\tilde{A}(x_g, y_g)$ | four adjacent global cells of $(x_g, y_g)$ and $(x_g, y_g)$ itself |
| $a(t_g)$ | a 0-1 variable to represent that every droplet reaches its sink at time $t_g$ |
| $\tilde{c}(x_g, y_g, t_g, k)$ | a 0-1 variable to represent the congestion of $(x_g, y_g)$ at time $t_g$ is greater than or equal to $k$ |
| $\tilde{c}_k^M$ | a 0-1 variable to represent that at least one variable $\tilde{c}(x_g, y_g, t_g, k)$ is one |
| $p_j^i(x_g, y_g, t_g)$ | a 0-1 variable to represent that droplet $d_j^i$ locates at the global cell $(x_g, y_g)$ at time $t_g$ |
| $m^i(x_g, y_g, t_g)$ | a 0-1 variable to represent that the two droplets of net $n_i$ are merged at $(x_g, y_g)$ at time $t_g$. |
| $\tilde{t}(i, i', t_g)$ | a 0-1 variable to represent that the droplet of two nets $n_i$ and $n_{i'}$ are in their common sink at time $t_g$ |
| $\hat{B}(x, y, t)$ | base cost for the node $v_{x,y,t}$ in detailed routing |
| $\hat{F}(x, y, t)$ | fluidic penalty for node $v_{x,y,t}$ in detailed routing |
| $\hat{E}(x', y', x, y, t)$ | electrode penalty for node $v_{x,y,t}$ in detailed routing |
| $\hat{A}(x, y, t)$ | activation penalty for node $v_{x,y,t}$ in detailed routing |
| $\hat{D}(x, y, t)$ | deactivation penalty for node $v_{x,y,t}$ in detailed routing |

surrounding segregation cells are considered as obstacles. And 3-pin nets are necessary for practical bioassays since two droplets need to be merged during their transportation for efficient mix operations [10]. Therefore, we formulate the droplet routing problem as follows:

**Input:** *A list of $m$ nets $N = \{n_1, n_2, \ldots, n_m\}$, where each net $n_i$ can be a 2-pin net (one droplet) or a 3-pin net (two droplets), and the locations of pins and obstacles.*

**Objective:** *Route all droplets from their source pins to their target pins while minimizing the maximum time to route all droplets.*

**Constraint:** *Both fluidic and electrode constraints must be satisfied, and the droplet routing should not step on the obstacles.*

## III. BIOCHIP ROUTING ALGORITHM

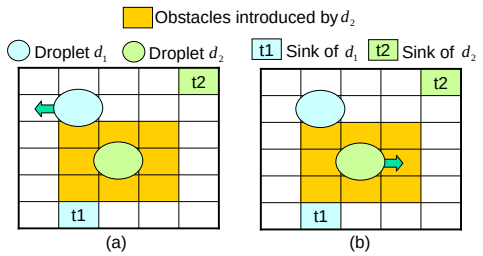


Figure 4. The motivation. Droplet $d_1$ moves one cell left (a) or stay at its original location (b).

The proposed biochip routing algorithm is a two-stage technique of global routing followed by detailed routing. There are two main advantages of the two-stage approach. First, a subset of the nets which heavily interfere with each other are first routed in the global routing phase. With the reduced problem size, the optimal routing solution could be found through a SAT formulation. Second, the global routing information could be utilized in detailed routing for better routing decision.

Unlike the previous works that route the droplets independently and incrementally, we maintain the routing information of all droplets during the whole process. Figure 4 illustrates the importance of considering the routing information of other droplets when routing a droplet. Note that droplet $d_2$ is considered as a $3 \times 3$ obstacle when determining the routing cost of $d_1$ due to the fluidic constraints. Previous works determine droplets' movement cost based only on the current positions of all droplets. Therefore, $d_1$ may move one cell left due to shorter distance to its sink and less routing congestion, as shown in Figure 4 (a). However, if the routing path of $d_2$ is known, then $d_1$ may stall, resulting with lesser cycles to move to its sink, as shown in Figure 4 (b). In other words, the absence of routing path information of other droplets may introduce unnecessary detours, which increases the droplet transportation time. Therefore, it is important to know the routing path of all other droplets while routing a droplet.

In this section, we first introduce the overview of the proposed algorithm. We then detail the global and detailed routing algorithms. The notations used in our algorithm are given in Table I.

### A. Routing Algorithm Overview

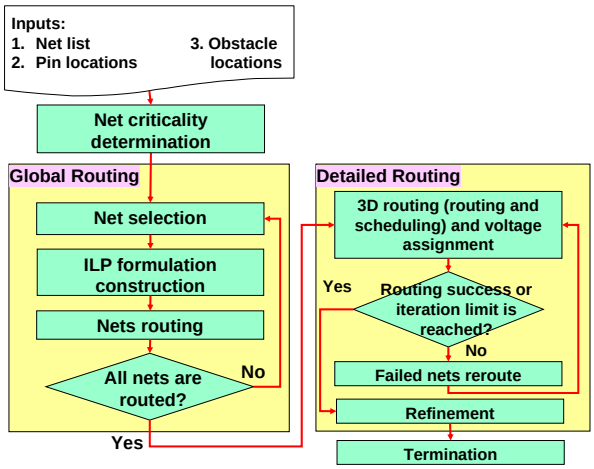


Figure 5. The routing algorithm overview.

Figure 5 shows the overview of the proposed routing algorithm. There are three phases in our algorithm: (1) net criticality calculation, (2) SAT-based global routing formulation, and (3) detailed routing based on a negotiation based routing scheme.

In net criticality calculation, we estimate the level of interference between net pairs and the criticality of each net. This information will be used in both global routing and detailed routing. In global routing, the goal is to determine a rough routing path and schedule of each droplet and to minimize the interference among droplets. We divide a biochip into a set of global cells. We first select a set of nets that heavily interfere with each other. Based on these global cells, we construct the SAT formulation and then transform it into an equivalent conjunctive normal form (CNF) for routing, where CNF is the conjunction of disjunctions of boolean variables.

In detailed routing, the goal is to simultaneously perform routing and scheduling and voltage assignment based on the result of global routing. A 3D routing graph is used, where each node represents a basic cell at a time step. Performing routing and scheduling is equivalent to finding a routing tree on the 3D routing graph. A post-step refinement is performed after finding a feasible solution. The whole algorithm terminates when there is no further improvement or no feasible solution can be found within a specified maximum number of iterations.

### B. Net Criticality Calculation

We say a net $n_i$ is critical if many other nets interfere with $n_i$. Let $\tilde{I}(i, i')$ denote the level of interference between two nets $n_i$ and $n_{i'}$.

$\tilde{I}(i,i')$ is defined in the following equation:

$$\tilde{I}(i,i') = \sum_{(x,y,t)\in\hat{U}_i,(x',y',t')\in U_f(x,y,t)} \frac{\hat{u}_{i'}(x',y',t')}{|\hat{U}_i||\hat{U}_{i'}|} + \sum_{(x,y,t)\in\hat{U}_i,(x',y')\in E_x^b\cup E_y^b\cup\{(x,y)\}} \frac{\hat{u}_{i'}(x',y',t)}{|\hat{U}_i||\hat{U}_{i'}|}, \quad (1)$$

where $\hat{U}_i$ is the set of $(x,y,t)$ that can be used by $d_j^i, j=1,2$, for routing and $\hat{u}_{i'}(x',y',t')$ is one if $d_j^{i'}, j=1,2$, can use $(x',y',t')$ for routing; otherwise, $\hat{u}_{i'}(x',y',t')$ is zero. The first term represents the possibility of violating the fluidic constraints while the second term represents the possibility of violating the electrode constraint. If $\tilde{I}(i,i')$ is large, then the routing solution of $n_i$ is very likely to be affected by the routing solution of $n_{i'}$. Finally, we define the criticality of $n_i$, denoted as $crit(i)$, as

$$crit(i) = \sum_{\forall i'\neq i, n_{i'}\in N} I(i,i'). \quad (2)$$

### C. Global Routing

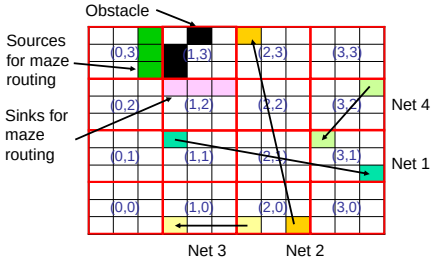


Figure 6. The illustration of a biochip being partitioned into a set of global cells and the determination of a prohibited turn.

The purpose of global routing is to determine a rough routing path and schedule for droplets and to minimize the interference among droplets. We decompose the global routing problem into a set of subproblems by selecting a set of nets that heavily interfere with each other. In this subsection, we first explain how the net selection is done. Then we present the SAT formulation for the global routing problem on the selected nets.

*1) Net Selection:* We use a heuristic for net selection. We first select $n_a$ and $n_b$ with the maximum interference value, denoted as $I^M$, into a set $N_g$. Then we iteratively add a net $n_c$ into $N_g$ if it satisfies the following equation:

$$0.5\times\sum_{n_a\in N_g} I(a,c) + 0.5\times\max_{n_a\in N_g} I(a,c) \geq \alpha\times I^M, \quad (3)$$

where $\alpha$ is a user specified constant and is set to 0.5 in this paper. The rational is that if the sum of the average and maximum interference values of $n_c$ and all nets in $N_g$ exceed a threshold, then the routing solution of $n_c$ will be heavily affected by the nets in $N_g$. Therefore, we should route $n_c$ with the nets in $N_g$. The above process repeats until each net is processed. Finally, $N_g$ is used as the input to our global routing algorithm.

*2) SAT Formulation:* In this subsection, we present the SAT formulation for global routing. We partition a biochip into a set of global cells $\hat{C}$, each containing $3\times 3$ basic cells. Figure 6 shows a biochip divided into 16 global cells. We assume that a droplet needs three cycles to pass though a global cell. We label the bottom-left global cell as $(0,0)$ and the top-right one as $(3,3)$. The rational behind this partition is that if we allow at most one droplet in a global cell at any time, then the fluidic constrains will be satisfied when routed appropriately. However, there are two cases that multiple droplets that should be allowed in the same global cell. First, two droplets of a net can move to a same global cell for merging. Second, the sources/sinks of multiple nets can dwell in the same global cell. For example, the sources of nets $n_2$ and $n_3$ in Figure 6 are in a same global cell while the sinks of nets $n_1$ and $n_4$ are in a same global cell. Thus, the two cases must be allowed in the global routing formulation.

Due to obstacles, droplets may not move from a global cell to another global cell. This restriction must be modeled in our SAT formulation. We say that three adjacent global cells $\{(x_{g1},y_{g1}),(x_{g2},y_{g2}),(x_{g3},y_{g3})\}$ as a *turn*. Obstacles may prevent a droplet from moving from a global cell $(x_{g1},y_{g1})$ to another global cell $(x_{g2},y_{g2})$ via cell $(x_{g3},y_{g3})$. For example, as shown in Figure 6, droplets are not allowed to move from $(0,3)$ to $(1,2)$ via $(1,3)$. If no droplets can move via a turn, we say that this turn is *prohibited*. The constraint that uses no prohibited turns is called *turn prohibition constraint* and must be satisfied in global routing.

A simple maze routing algorithm can be used to determine all prohibited turns. The sources (sinks) of maze routing are the basic cells along the boundary of $(x_{g1},yg1)$ $((x_{g3},y_{g3}))$. Only basic cells that are in the global cell $(x_{g2},y_{g2})$ (and sources/sinks) can be used for routing. If there is a path from a source to a sink, then this turn is not a prohibited turn. For example, the turn $\{(0,3),(1,3),(1,2)\}$ shown in Figure 6 is a prohibited turn.

In the following subsections, we first introduce how to construct the SAT formulation.

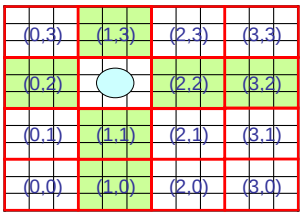


Figure 7. The illustration of the congestion in global routing.

*Objective Function:* The goal of the global routing is to minimize the maximum droplet transportation time and the interference among droplets by minimizing the maximum congestion of all global cells. The *congestion* of a global cell is defined as the number of droplets that may affect the movement of droplets in it. Note that a droplet introduces one unit of congestion to all cells $(x'_g,y'_g)\in E_x^g\cup E_y^g$ at time $t$ if $(x_g,y_g,t_g)$ is used for routing at time $t$, as shown in Figure 7. Note that if there are no droplets in a global cell, we do count for the congestion of it. Therefore, the objective function is defined by the following equation:

$$\beta\sum_{1\leq t_g<T_g} t_g(a(t_g)-a(t_g-1)) + \gamma\sum_{k=3}^{|D|}\tilde{c}_k^M, \quad (4)$$

where $\beta$ and $\gamma$ are user-specified constants and are both set to be 1 in this paper. Note that the electrode constraint will be violated only if we move more than two droplets simultaneously. Therefore, we do not add $\tilde{C}_k^M$ in the objective function if $k$ is less than 3.

*Constraints:* There are total six constraints in our SAT formulation.

1) *Objective function computation:* $a(t_g)$ is one if and only if every droplet reaches its sink at time $t_g$; that is, $p_j^i(\hat{t}_{g,x}^i,\hat{t}_{g,y}^i,t_g)=1$ for all droplets. Therefore, $a(t_g)$ can be computed by the following equation:

$$\prod_{\forall d_j^i\in D}(p_j^i(\hat{t}_{g,x}^i,\hat{t}_{g,y}^i,t_g)\leftrightarrow a(t_g)). \quad (5)$$

Now we show how to compute the congestion value $\tilde{c}(x_g,y_g,t_g,k)$. If a droplet $d_j^i$ locates at a global cell $(x_g,y_g)$ at time $t_g$, then $d_j^i$ introduces one unit of congestion for all global cells in $E_x^g\cup E_y^g$ and $(x_g,y_g)$ due to the electrode constraint. $\tilde{c}(x_g,y_g,t_g,k)$ is one if and only if the congestion of a global cell $(x_g,y_g)$ at time $t_g$ is greater than or equal to $k$. Therefore,

$\tilde{c}(x_g, y_g, t_g, k)$ can be modeled by the following equation:

$$\left(\sum_{(x'_g,y'_g)\in E^g_x\cup E^g_y,\forall d^i_j\in D} p^i_j(x'_g, y'_g, t_g) \geq k\right) \left(\bigvee_{\forall d^i_j\in D} p^i_j(x_g, y_g, t_g)\right) \leftrightarrow \tilde{c}(x_g, y_g, t_g, k). \quad (6)$$

Finally, $\tilde{c}^M_k$ can be modeled by the following equation:

$$\prod_{1\leq t_g<T_g, k\in[i,|D|]} (\tilde{c}^M_k \leftrightarrow \bigvee_{1\leq t_g<T_g,(x_g,y_g)\in\hat{C}} \tilde{c}(x_g, y_g, t_g, k)). \quad (7)$$

2) *Source and sink requirements:* We assume that at time zero, all droplets are located at their sources. All droplets must reach their sink except the second droplet of a 3-pin net, since it will definitely reach its sink once it is merged with the first droplet of the same net. Once a droplet reaches its sink, it stays at there. Therefore, the above requirements can be expressed by the following equations:

$$\prod_{\forall d^i_j\in D} p^i_j(s^{i,j}_{g,x}, s^{i,j}_{g,y}, 0) \quad (8)$$

$$\prod_{\forall d^i_j\in D} \bigvee_{t_g\in[0,T_g)} p^i_j(\hat{t}^i_{g,x}, \hat{t}^i_{g,y}, t_g) \quad (9)$$

$$\prod_{\forall d^i_j\in D', 0\leq t_g<T_g-1} \left(p^i_j(\hat{t}^i_{g,x}, \hat{t}^i_{g,y}, t_g) \rightarrow p^i_j(\hat{t}^i_{g,x}, \hat{t}^i_{g,y}, t_g+1)\right), \quad (10)$$

where $D'$ is the set of droplets $d^i_1$.

3) *Exclusivity Constraint:* The exclusivity constraint states that each droplet has only one location at a time. This constraint also states that at most one droplet can be in a global cell at any time, except the two cases mentioned before. Let $f(x_g, y_g, t_g) = \sum_{\forall d^i_j\in D} p^i_j(x_g, y_g, t_g) - \sum_{n_i,n_{i'}\in\tilde{N}} \tilde{t}(i, i', t_g)$ if more than two nets' sinks are in cell $(x_g, y_g)$; Otherwise, we could obtain that $f(x_g, y_g, t_g) = \sum_{\forall d^i_j\in D} p^i_j(x_g, y_g, t_g) - \sum_{n_{i'}\in N'} m^{i'}(x_g, y_g, t_g)$. $f(x_g, y_g, t_g) = 1$ means that there is at least one droplet in $(x_g, y_g)$ at time $t_g$. Therefore, this constraint can be modeled by the following equations:

$$\prod_{\forall d^i_j\in D, 0\leq t_g<T_g} \left(\sum_{(x_g,y_g)\in\hat{C}} p^i_j(x_g, y_g, t_g) = 1\right) \quad (11)$$

$$\prod_{(x_g,y_g)\in\hat{C}, 0\leq t_g<T_g} \left(f(x_g, y_g, t_g) \leq 1\right) \quad (12)$$

$$\prod_{\forall(\hat{t}^i_{g,x},\hat{t}^i_{g,y})=(\hat{t}^{i'}_{g,x},\hat{t}^{i'}_{g,y})} \left(\overline{p^i_1(x_g, y_g, t_g) \oplus p^i_2(x_g, y_g, t_g)} \leftrightarrow \tilde{t}(i, i', t_g)\right), \quad (13)$$

where $\oplus$ represents XOR.

4) *Droplet movement constraint:* A droplet can only move to one of its four neighboring global cells or stay at its original location at next time step. Therefore, this constraint can be modeled by the following equation:

$$\prod_{\forall d^i_j\in D,(x_g,y_g)\in\hat{C}, 0\leq t_g<T_g-1} \left(p^i_j(x_g, y_g, t_g) \rightarrow \bigvee_{(x'_g,y'_g)\in\bar{A}(x_g,y_g)} p^i_j(x'_g, y'_g, t_g+1)\right) \quad (14)$$

5) *Turn prohibition constraint:* This constraint states that no prohibited turns can be used and can be expressed by the following equation:

$$\prod_{\substack{\forall d^i_j\in D, 0\leq t_g<t'_g<T_g-1,\\ ((x_{g1},y_{g1}),(x_{g2},y_{g2}),(x_{g3},y_{g3})),\in U_t}} \left(\overline{p^i_j(x_{g1}, y_{g1}, t_g)} \vee \overline{p^i_j(x_{g3}, y_{g3}, t'_g+1)} \vee \bigvee_{t''_g\in(t_g,t'_g]} \overline{p^i_j(x_{g2}, y_{g2}, t''_g)}\right) \quad (15)$$

where $U_t$ is the set of prohibited turns. Note that the above constraint includes the case that $d^i_j$ stays at $(x_{g2}, y_{g2})$ for a period of time.

6) *3-pin nets:* We use the following four equations to handle 3-pin nets:

$$\prod_{\forall n_i\in N',(x_g,y_g)\in\hat{C},0\leq t_g<T_g} \overline{p^i_1(x_g, y_g, t_g) \oplus p^i_2(x_g, y_g, t_g)} \leftrightarrow m^i(x_g, y_g, t_g) \quad (16)$$

$$\prod_{\forall n_i\in N'} \bigvee_{(x_g,y_g)\in\hat{C},\ 0\leq t_g<T_g} m^i(x_g, y_g, t_g) \quad (17)$$

$$\prod_{\forall n_i\in N',(x_g,y_g)\in\hat{C},0\leq t_g<T_g-1} \left(m^i(x_g, y_g, t_g) \rightarrow \sum_{(x'_g,y'_g)\in\bar{A}(x_g,y_g)} m^i(x'_g, y'_g, t_g+1)\right) \quad (18)$$

$$\prod_{\forall n_i\in N'} \Big(\sum_{t_g=0}^{T_g-1} tm^i(x_g, y_g, t_g) - \sum_{t_g=0}^{T_g-2} (t_g+1) \sum_{(x_g,y_g)\in\hat{C}} (p^i_1(x_g, y_g, t_g+1) - p^i_1(x_g, y_g, t_g)) - p^i_1(\hat{t}^i_{g,x}, \hat{t}^i_{g,y}, 0) \leq -1\Big). \quad (19)$$

Constraint (16) is used to determine whether two droplets are merged; i.e., in the same global cell at the same time. Constraint (17) is used to guarantee that two droplets are merged during their transportation by restricting that their physical location must be the same for at leat one time step. Constraint (18) ensures that once two droplets are merged, they cannot be split. Constraint (19) states that these two droplets must be merged before reaching their sink. This constraint also includes one special case if the two droplets and their sink are all in the same global cell. In this case, they will be merged and reach their sink at time zero.

Finally, we AND all equations ((5) to (19)) to form the complete SAT formulation.

*3) Transformation into CNF:* Almost every constraint listed in Section III-C2 is not expressed in CNF. To transform these constraints into CNF, we use the following techniques:

1) Suppose that a constraint can be expressed as:

$$a_1\dot{x}_1 + a_2\dot{x}_2 + \ldots a_n\dot{x}_n \leq b, \quad (20)$$

where $a_k, b \in Z^{+1}$. This constraint is called as a *threshold* boolean function [2]. We can use the method proposed in [2] to transform a threshold boolean function to its equivalent CNF formula.

2) Every boolean function, implication, and biconditional ($\leftrightarrow$) can be transformed into its equivalent CNF formula by using methods shown in [8].

[1]Equality constraint can be handled as less-than-or-equal and greater-than-or-equal constraints, while the latter can be transformed in to the less-than-or-equal constraint by multiplying -1.

### D. Detailed Routing

In this subsection, we present the proposed detailed routing scheme, which is a negotiation based algorithm inspired by [7], [15]. The proposed routing algorithm iteratively routes and schedules all droplets by routing all droplets in a 3D routing graph, where each node represents a basic cell at a time step, in the decreasing order of their criticality. We also perform rip-up and reroute on failed nets. A failed nets is a net that cannot find a routing solution that satisfies the fluidic constraints and the electrode constraints. The proposed routing algorithm terminates when a feasible routing solution is found, or when a user-specified number of iterations is reached. Then we perform a post-step refinement to further improve the solution. Finally, we present how to handle 2-pin and then 3-pin nets. Table I lists the notations used in detailed routing.

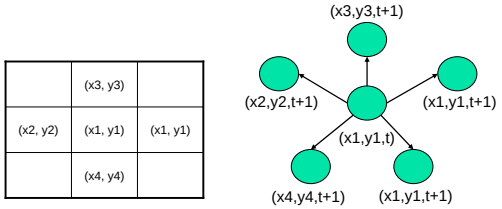


Figure 8. The illustration of the routing graph in detailed routing.

*1) Routing Graph Construction:* We construct a 3D directed routing graph $G_d = (V_d, E_d)$, where $V_d$ is the set of routing nodes and $E_d$ represents the set of edges. A node $v_{x,y,t}$ represents a basic cell $(x, y)$ at time $t$. Figure 8 shows how to construct such a routing graph. An edge connected from $v_{x,y,t}$ to $v_{x',y',t+1}$ if a droplet can move from $(x, y)$ to $(x', y')$ from time $t$ to time $t + 1$, such as the edges from $v_{x1,y1,t}$ to $v_{x2,y2,t+1}$. The 3D routing graph is more flexible for droplet movement compared with the 2D one used in [15].

Each routing node $v_{x,y,t}$ is associated with its cost $\phi(d^i, x', y', x, y, t)$ to represent the cost if $d^i$ uses both $v_{x',y',t-1}$ and $v_{x,y,t}$ for routing, where $d^i$ is the droplet of a 2-pin net $n_i$. We say that $v_{x,y,t}$ is *activated* if electrode of cell $(x, y)$ is activated at time $t$. To satisfy the fluidic constraints, no other droplets can use $v_{x'y't'} \in U_f(x, y, t)$ for routing if a droplet uses $v_{x,y,t}$ for routing. For the electrode constraint, if $d^i$ moves from $(x', y')$ to $(x, y)$ at time $t$, then no nodes $v_{x'',y'',t} \in U_e(x', y', x, y, t)$ can be activated.

The goal of detailed routing is to find the minimum cost routing tree $\hat{R}^i$ for each droplet embedded in $G_d$ and to perform voltage assignment for correct droplet movement, provided that the fluidic and electrode constraints are all satisfied. A routing tree's cost is the sum of the cost of all tree nodes.

*2) Cost of Routing Nodes:* The cost $\phi(d^i, x', y', x, y, t)$ is defined by the following equation:

$$\phi(d^i, x', y', x, y, t) = \hat{B}(x, y, t) + \hat{F}(x, y, t) + \hat{E}(x', y', x, y, t) + \hat{A}(x, y, t) + \hat{D}(x, y, t). \quad (21)$$

To minimize the maximum droplet transportation time, we want to minimize the time that a droplet reaches its sink. Therefore, $\hat{B}(x, y, t)$ is zero if $(x', y')$ is the sink of $d^i$; otherwise, $\hat{B}(x, y, t)$ is $\delta$, which is a user-specified constant and is set to be 4 in this paper. The fluidic penalty $\hat{F}(x, y, t)$ is used to guide the detailed router to satisfy the fluidic constraints and is defined by the following equation:

$$\hat{F}(x, y, t) = \hat{f}(x, y, t)/N_f \times H_f(x, y, t), \quad (22)$$

where $\hat{f}(x, y, t)$ is the number of 3D cubes $U_f(d^{i'}, x', y', t')$ that contains $v_{x,y,t}$, and $H_f(x, y, t)$ is the historical fluidic penalty. $N_f$ is the normalization factor and is set to be the maximum of all $\hat{f}(x, y, t)$.

The electrode penalty $\hat{E}(x', y', x, y, t)$, similar to the fluidic penalty $\hat{F}(x, y, t)$, is used to guide the detailed router to satisfy the electrode constraint and is defined by the following equations:

$$\hat{E}(x', y', x, y, t) = \sum \hat{e}_a(x', y', x, y, t)/N_e \times H_e(x, y, t), \quad (23)$$

where $\hat{e}_a(x', y', x, y, t)$ is the number of activated nodes in $U_e(x', y', x, y, t)$ and $H_e(x, y, t)$ is the historical electrode penalty. $N_e$ is the normalization factor and is set to be the maximum of all $\hat{e}_a(x', y', x, y, t)$.

Recall that when we activate $v_{x,y,t}$ for droplet movement, and the electrode constraint may be violated. Therefore, we add the activation penalty $\hat{A}(x, y, t)$ in the cost function and it is defined by the following equation:

$$\hat{A}(x, y, t) = \sum_{(x',y') \in E^b_x \cup E^b_y \cup (x,y)} \hat{a}(x', y', t)/N_a \times H_a(x, y, t), \quad (24)$$

where $\hat{a}(x, y, t)$ is the number of $U_e(x'', y'', x', y', t)$ that contains $v_{x,y,t}$ if $v_{x,y,t}$ is activated and $H_a(x, y, t)$ is the historical activation penalty. Note that if $\hat{a}(x'', y'', t) \in U_e(x', y', x, y, t)$, then we use $\hat{a}(x'', y'', t)+1$ when evaluating $\hat{A}(x, y, t)$. $N_a$ is the normalization factor and is set to be the maximum of all $\hat{a}(x', y', t)$.

Finally, besides activating other nodes, the activation of a node can deactivate some other nodes used by other droplets. Therefore, we penalize to activate $v_{x,y,t}$ if it results in the deactivation of other routing nodes that are used by other droplets, which is defined by the following equation:

$$\hat{D}(x, y, t) = \sum_{(x',y') \in E^b_x \cup E^b_y} \hat{d}(x', y', t) \times e^{r+1} \times H_d(x, y, t), \quad (25)$$

where $\hat{d}(x, y, t)$ is one if $v_{x,y,t}$ is deactivated and a droplet uses it for routing, $r$ is the current iteration number, and $H_d(x, y, t)$ is the historical deactivation cost. With this cost, the detailed router tends not to affect other droplets when the rip-up and reroute have been performed many times.

The historical cost $H_f(x, y, t)$ ($H_e(x, y, t)$, $H_a(x, y, t)$, and $H_d(x, y, t)$) is initialized to one and is increased whenever $\hat{f}(x, y, t)$ ($\hat{e}_a(x', y', x', y', t)$, $\hat{a}(x, y, t)$, or $\hat{d}(x, y, t)$) is larger than zero. If the constraints are violated in many previous iterations, the historical costs will be large, and thus the detailed router will be less likely to use $v_{x,y,t}$ for routing.

*3) Routing Algorithm:* We route all successfully routed nets in global routing in the decreasing order of its criticality. Initially, the voltage values of all rows and columns voltages are set as floating. When routing net $n_i$, we first remove the routing tree $\hat{R}^i$ obtained in the previous iteration and add the source of $d^i$ into the priority queue $Q$ with the row/column voltages being floating and cost being zero. Every time we remove $v_{x'y,'t-1}$ with the lowest cost from $Q$ and evaluate the cost of $v_{x,y,t}$ if $v_{x,y,t}$ is the fanout of $v_{x',y',t-1}$ and is used to route $d^i$ in global routing. We also need to assign the voltage values of row $y$ and column $x$ for droplet movement. If $d^i$ stays at its original location; i.e., $x' = x$ and $y' = y$, then the voltages of row $y$ and column $x$ remained unchanged. Otherwise, either row $y$ or $x$ should be assigned low voltage, but not both. Then we insert $v_{x,y,t}$ into $Q$ with cost $P_{x',y',t-1} + \phi(d^i, x', y', x, y, t)$ and assigned voltages, where $P_{x',y',t-1}$ is the path cost from the source to $v_{x',y',t-1}$.

The above process repeats until $d^i$ reaches its sink, and then it stays there. Then we perform back-trace to update the voltage of rows/columns and to construct the routing tree $\hat{R}^i$. We also update the fluidic, electrode and activation penalties. If $d^i$ uses $v_{x',y',t-1}$ and $v_{x,y,t}$ for routing, $\hat{a}(x'', y'', t)$ is increased by one, $(x'', y'') \in U_e(x', y', x, y, t)$. If $v_{x,y,t}$ is activated, we increase $\hat{e}_a(x'', y'', x', y', t)$ by one for all $v_{x,y,t} \in U_e(x'', y'', x', y', t)$. Finally, if $d^i$ uses $v_{x,y,t}$ for routing, then $\hat{f}_{x',y',t'}$ is increased by one, $v_{x,y,t} \in U_f(x', y', t')$.

If a net is failed for routing, we rip up and reroute this net in the next iteration without honoring the global routing result. Moreover, if $n_i$ fails in global routing, we treat $n_i$ as a failed net and route $n_i$ after the first iteration.

After finding a feasible detailed routing, we perform a post-routing refinement for further optimization. We iteratively rip up and reroute the net $n_i$ with the highest droplet transportation time without violating the fluidic and electrode constraints. The above process repeats until it is impossible to route all nets or there is no further improvement.

TABLE II
ROUTING RESULT OF THE TWO BIOASSAYS.

| Circuit | [15] + [13] | | [15] + [6] | | The progressive-ILP in [14] | | The proposed algorithm | |
|---|---|---|---|---|---|---|---|---|
| | Max/avg $T_m$ (cycle) | CPU time | Max/avg $T_m$ (cycle) | CPU time (sec) | Max/avg $T_m$ (cycle) | CPU time (sec) | Max/avg $T_m$ (cycle) | CPU time (sec) |
| Diagnostics_1 | 40/16.72 | $< 0.01$ | 47/20.18 | 0.01 | 24/13.09 | 0.54 | 19/12.36 | 2.66 |
| Diagnostics_2 | 35/13.46 | $< 0.01$ | 52/16.80 | 0.01 | 21/10.93 | 0.57 | 20/10.20 | 2.64 |
| Protein_1 | 48/10.32 | $< 0.01$ | 55/24.40 | 0.05 | 26/16.15 | 3.40 | 23/15.78 | 9.84 |
| Protein_2 | 36/11.00 | $< 0.01$ | 53/14.33 | 0.03 | 29/10.47 | 1.43 | 21/9.25 | 6.97 |

*4) Handling 3-pin nets:* We use the similar technique as that in [15] to route a 3-pin net. We first route the droplet $d_1^i$ with a longer Manhattan distance and then route $d_2^i$ to merge these two droplets. If merging is not possible, net $n_i$ is failed for routing and we reroute $d_1^i$ with a higher cost to routing nodes that cannot be used by $d_2^i$ for routing.

## IV. EXPERIMENTAL RESULT

Our algorithm was implemented in the C++ language and minisat+ [1] was used as our solver. We compared our results with the progressive-ILP formulation proposed in [14] and two indirect algorithms [6], [13] that require initial direct-addressing routing solutions. For these two algorithms, we used BioRoute [15] to generate a direct-addressing routing solution and then applied these algorithms to find the final solution. The maximum number of iterations to find a feasible solution in detailed routing is set to be 50. All algorithms were executed on a Linux machine with two 2.6 GHz AMD-64 CPUs and 6 GB memory. We evaluated our routing algorithm on two practical bioassays used in previous works [10], [14]: the *in-vitro* diagnostics and the colorimetric protein assay.

Table II shows our results. We report the maximum and average maximum droplet transportation time ($T_m$) for all 2D planes and CPU time required to route all 2D planes. As shown in this table, our routing algorithm can obtain smaller maximum and average droplet transportation time (in cycles) than previous approaches, under reasonable CPU times. For example, for the protein_2 benchmark, our algorithm can obtain smaller droplet transportation time (21 cycles vs. 29 cycles) with longer CPU time (6.97 sec vs. 0.57 sec) compared with the progressive-ILP formulation. This result demonstrates that our algorithm is very effective for droplet routing on cross-referencing biochips. Figure 9 shows the routing result of one 2D plane of the diagnostics_1 benchmark at cycle 3.

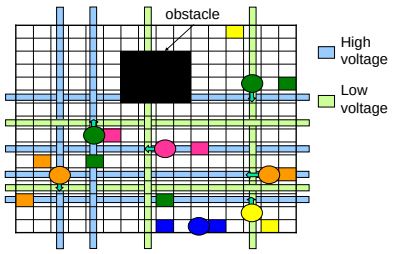


Figure 9. The result of the diagnostics_1 benchmark.

Here we discuss why our algorithm can obtain better solution quality than previous approaches. Previous indirect algorithms [6], [13] require direct-addressing routing solutions. Therefore, their solution quality is limited by that of the direct-addressing routing solutions used. In contrast, our algorithm directly targets at the droplet routing on cross-referencing biochips. Therefore, our algorithm has a high flexibility for droplet routing and scheduling. Although the progressive-ILP based routing scheme also directly targets at cross-referencing biochips, it determines the droplets' position based only on the current droplets' locations. Therefore, unnecessary droplet movement, such as the case shown in Figure 4 (a), may occur, and increase the maximum droplet transportation time. In our algorithm, we utilize the routing path information while maximizing the number of droplet movement in global routing. Hence, our algorithm can avoid the case shown in Figure 4 (a) and obtain a solution with faster droplet transportation time.

## V. CONCLUSION

In this paper, we have proposed the *first* SAT-based routing algorithm for cross-referencing biochips. We adopted a two-stage technique of global routing followed by detailed routing. In global routing, we use a SAT-based formulation to iteratively route a set of nets that heavily interfere with each other. In detailed routing, a negotiation-based algorithm is proposed to simultaneously perform droplet routing and scheduling and voltage assignment. Experimental results demonstrated the effectiveness of our routing algorithm.

Future work lies on power-aware droplet routing algorithm for cross-referencing biochips. Cross-referencing biochips incur much higher power consumption for droplet movement since a row and a column of electrodes must be simultaneously activated to move a droplet. For an $N \times N$ biochip, the activation power of a single control pin is $N$ times higher than that of direct-addressing biochips which only need to activate one electrode. This concern becomes even critical when battery-driven applications are involved, where it is desirable to incorporate power-saving issues into the droplet routing for these kinds of applications.

## REFERENCES


[1] http://minisat.se/.

[2] F. A. Aloul, A. Ramani, I. L. Markov, and K. A. Sakallah. Generic ILP versus specialized 0-1 ILP: an update. In *ICCAD*, pages 450–457, 2002.

[3] K. F. Böhringer. Modeling and controlling parallel tasks in droplet-based microfluidic systems. *TCAD*, 25(2):334–344, 2006.

[4] M. Cho and D. Z. Pan. A high-performance droplet router for digital microfluidic biochips. In *ISPD*, pages 200–206, April 2008.

[5] J. Gong, S.-K. Fan, and C.-J. Kim. Portable digital microfluidics platform with active but disposable lab-on-chip. In *MEMS*, pages 355–358, 2004.

[6] E. J. Griffith, S. Akella, and M. K. Goldberg. Performance characterization of a reconfigurable planar-array digital microfluidic system. *TCAD*, 25(2):435–357, 2006.

[7] L. McMurchie and C. Ebeling. Pathfinder: a negotiation-based performance-driven router for fpgas. In *FPGA*, pages 111–117, 1995.

[8] S. Russel and P. Norvig. *Artificial Intelligence - A Modern Approach*. Prentice Hall, 1995.

[9] F. Su, K. Chakrabarty, and R. B. Fair. Microfluidic-based biochips: Technology issues, implementation platforms, and design-automation challenges. *TCAD*, 25(4):211–223, 2006.

[10] F. Su, W. Hwang, and K. Chakrabarty. Droplet routing in the synthesis of digital microfluidic biochips. In *DATE*, pages 323–328, 2006.

[11] E. Verpoorte and N. F. D. Rooij. Microfluidics meets MEMS. *Proceedings of IEEE*, 1(6):930–953, June 2003.

[12] T. Xu and K. Chakrabarty. Droplet-trace-based array partitioning and a pin assignment algorithm for the automated design of digital microfluidic biochips. In *CODES+ISSS*, pages 112–117, Oct. 2006.

[13] T. Xu and K. Chakrabarty. A cross-referencing-based droplet manipulation method for high-throughput and pin-constrained digital microfluidic arrays. In *DATE*, pages 552–557, Apr. 2007.

[14] P.-H. Yuh, S. S. Sapatnekar, C.-L. Yang, and Y.-W. Chang. A progressive-ILP based routing algorithm for cross-referencing biochips. In *DAC*, pages 284–289, June 2008.

[15] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang. BioRoute: A network-flow based routing algorithm for digital microfluidic biochips. In *ICCAD*, pages 752–757, Nov. 2007.