

Droplet-Routing-Aware Module Placement for Cross-Referencing Biochips

Zigang XIAO and Evangeline F.Y. Young
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{zgxiao,fyyoung}@cse.cuhk.edu.hk

ABSTRACT

Digital Microfluidic Biochip (DMFB) is a revolutionary technology for performing lab-on-a-chip experiments. Comparing to traditional direct-addressing design of DMFB, Cross-Referencing Biochip is a flexible design which not only helps to reduce pin number on chip but also brings down manufacturing cost. Following the generally accepted DMFB top-down design methodology, namely task scheduling, resource binding, module placement, droplet routing, previous works that focus on cross-referencing biochip routing are all based on the placement result generated for direct-addressing biochip. In this paper, we present an ILP-based placement method that first utilizes the property of cross-referencing for the purpose of optimizing routing. Furthermore, one previously ignored electrode interference problem on modules (blocks) is addressed in this paper. Real-life bioassay protocol based benchmarks are used to evaluate the proposed method. Experimental results show that the placement result generated by our placer yields better routing result comparing with those from placer for direct-addressing DMFB.

Categories and Subject Descriptors

B.7.2 [Hardware, Integrated Circuit]: Design Aids - Placement and routing

General Terms

Algorithms, Design, Performance

Keywords

Biochip, Cross-Referencing, DMFB, Microfluidics, Placement, Synthesis

1. INTRODUCTION

1.1 Digital Microfluidic Biochip

Digital Microfluidic Biochip has been receiving more and more attention today [1]. It shows great advantages in med-

ical, pharmaceutical and environmental monitoring applications [2]. In contrast to previous continuous-flow systems, DMFB offers an important property, i.e., *dynamic reconfigurability* [3]. With the flexible design, many tasks can be performed without using expensive equipment and human resource. However, highly complicated computer-aided design support is strongly in demand for DMFB as in traditional VLSI design. Paper [4] introduced a novel top-down design methodology for DMFB. The behavioral model for a biochemical assay is first acquired from the laboratory protocol. Then architectural-level synthesis follows to generate the macroscopic structure of the chip, which contains the *task scheduling* and *resource binding* information. Finally, geometry-level synthesis is performed to generate the detailed layout of the chip, which includes the module *placement* and *droplet routing*.

Figure 1(a) gives a schematic diagram of a DMFB [5]. The DMFB can be viewed as a 2D-array, which allows various types of biochemical reaction taking place. There are dispensing reservoirs and optical detectors in a DMFB, which allows many kinds of laboratory research to be carried out on the chip as in a miniaturized laboratory, i.e., lab-on-a-chip. The *discrete* liquid objects that containing chemical materials and being operated on chip are called *droplets*. They are the basic units to carry out biochemical operations, such as *mixing*, *detection* and *diluting*.

1.2 Cross-Referencing DMFB

In a DMFB, droplets' movements are controlled by the electro-hydrodynamic forces generated by the electrodes, which causes the droplets to move horizontally or vertically. Hence, all the droplets can be transported to their destinations in a given amount of time. Currently, there exists two kinds of technology for the manipulation of droplet movement. The first one is called *direct-addressing*. In this scheme,

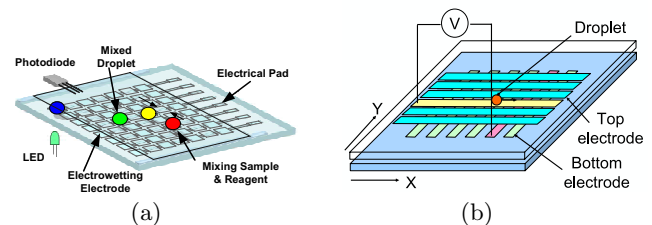


Figure 1: (a) Schematic view of DMFB. (b) A Cross-referencing biochip.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISPD'10, March 14–17, 2010, San Francisco, California, USA.
Copyright 2010 ACM 978-1-60558-920-6/10/03 ...\$10.00.

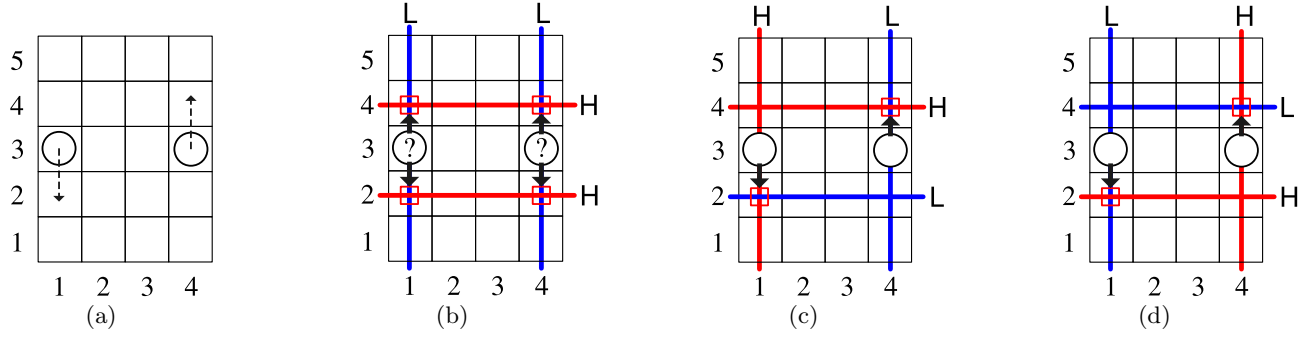


Figure 2: (a) The two droplets are moving from current cell to the arrow-pointed cell. (b) Electrode interference happens, droplets’ movements are intervened. (c) Solution: apply voltage appropriately. (d) An alternative solution.

an electrode is fixed under each cell of the chip. Each cell can thus be addressed independently to bring about the droplets’ movement, i.e., pulling a droplet to it from one of the adjacent cells. It is simple, yet expensive and rather limited for large scale production. Another more promising method is called *cross-referencing*, which applies high and low voltages to a row (column) and a column (row) electrode so as to *activate* the cell at the corresponding intersection point. Figure 1(b) gives an illustration of a cross-referencing biochip [6].

1.3 Electrode Interference

In cross-referencing biochips, the control of the activation of a cell is in a row-column manner, *extra cells* may be activated when more than one droplets are going to be manipulated at the same time. Thus some droplets may be affected by the *extra activated cells* unintentionally. This unwanted effect is referred as *electrode interference* and *electrode constraint* should be imposed to avoid such erroneous cases during droplet routing. Cross-referencing imposes tighter constraints to simultaneous manipulation of droplets’ movements than direct-addressing. An example is given in Figure 2. Suppose the left droplet needs to be moved from (1,3) to (1,2) and another from (4,3) to (4,4). We assign high voltage to row 2 and 4, as in Figure 2(b), while assigning low voltage to column 1 and 4, so as to activate both cells (1,2) and (4,4). However, cells (1,4) and (4,2) will also be activated, and they are called *extra activated*. Hence, the droplets may not be moved as planned. If appropriate voltage assignment can be performed as in Figure 2(c), correct movements can be guaranteed. Figure 2(d) is an alternative solution by flipping all the electrodes in Figure 2(c) to its opposite value. For rows and columns which are not marked as ‘H’ or ‘L’, ground voltage is assigned and no cells will be activated along those rows and columns. Note that those extra activated cells do not necessarily imply electrode interference. If no droplet is around the extra activated cell, no electrode interference will ever happen. The main challenge of routing droplet in cross-referencing DMFB is to avoid electrode interference. In this sense, a cross-referencing DMFB has much tighter constraint than a direct-addressing one, which means the throughput may be severely decreased. Nevertheless, one can maximize the parallelism by carefully arranging voltages, and achieves routing result as good as routing in direct-addressing DMFB. In paper [6] and [7] it is shown that all the benchmarks can be routed by both proposed cross-referencing router. These routers are designed

dedicatedly for routing in cross-referencing biochip. Furthermore, the routing time and cells used¹ is also competitive comparing with the direct-addressing router. Hence cross-referencing biochip can still achieve high-throughput with the help of well-designed router.

1.4 Blocks in Cross-Referencing Droplet Routing

Another important issue not addressed in previous papers is the potential electrode interference inside blocks. The blocks in droplet routing is totally different from the blocks in traditional routing. In fact, they are *on-going operations*, such as *mixing*, *dilution*, *storage*, etc. Due to the reconfigurability of DMFB, these operations can be done anywhere on the chip. For instance, the *mixing* operation is accomplished by moving two droplets to the same location first, and then driving them together around some pivot point. In order not to violate the fluidic constraint [5], a *guarding ring* is imposed around the module. During the routing, if the droplet inside the module is pulled outside unintentionally, the correctness of the whole bioassay will be ruined. Hence, it is very important to ensure the droplet being inside the block when the operation is still in progress. However, as introduced in Section 1.3, there may be extra-activated cell appearing on chip during droplet routing. They are allowed if and only if no electrode interference will happen to the routing. Nevertheless, previous works did not address this issue explicitly, which will cause serious problem. In this paper, because the droplet routing time is relatively small comparing with the operation time (e.g., 0.2s vs. 2s), we assume that extra-activated cells inside a module itself will not cause any problem, since the regular activation sequence will eventually steer the droplet onto the correct track. But it is forbidden in the guarding ring because it is the only way that the droplet will be pulled out. An example is given in Figure 3. We also assume that the module operation is temporarily stopped during droplet routing. Finally, the router in paper [7] is improved by including this new constraint.

1.5 Placement Problem of DMFB

Placement is a key problem in the synthesis of DMFB computer-aided design. Given scheduling result and resource binding result from architectural-level synthesis, the placement algorithm involves placing the microfluidic modules in order to optimize some design metrics while satisfying

¹A metric used to evaluate the fault-tolerance of the routing result.

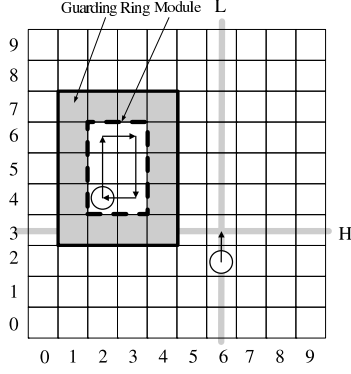


Figure 3: The gray rectangle denotes the guarding ring of the module. Note that low voltage cannot be assigned to column 1~4, otherwise, cells in the guarding ring will be extra activated.

various constraint. Due to the reconfigurability of DMFB, placement problem of DMFB is different from the traditional placement in electronic design [8]. There are various types of modules that can be placed in the same location during different time intervals, while some of them has to be fixed after placed, such as detectors and waste reservoirs. Moreover, since the bioassay has a timing span and precedence over the operations, it can be divided into different subproblems according to the scheduling result. Each subproblem is related with its predecessor and descendant, and the placement result directly affects the hardness and quality of the corresponding droplet routing. In this paper, we propose a comprehensive method that solves the placement problem by considering the properties of cross-referencing DMFB. The major part of the problem is handled using ILP-based formulation. The motivation is that the properties observed from cross-referencing is easy to model and naturally fitted into ILP objective function. Moreover, feasible result can be obtained and improved in an iterative manner when using ILP.

The remaining of the paper is organized as follows. In Section 2.1, related prior works are discussed. A formulation of the problem is given in Section 3. The details of our placement tool is introduced in Section 4. In Section 5, several real-world bioassay is used to evaluated the placement results generated by our tool. Finally, conclusion are drawn in Section 6.

2. PREVIOUS WORKS AND OUR CONTRIBUTION

2.1 Previous Works

Module placement is a critical step before the droplet routing step in biochip automated design. It directly affects the routability and routing quality during the routing step. A placement with inappropriate positioning of modules and reservoirs may lead to a harder routing. One of the consequence is the placement result is totally unroutable, as in Figure 4(a). Even if it is routable, it may be a bad placement in terms of throughput. For example, as in Figure 4(b), the modules might be placed in an over-congested way that many stalling steps are introduced, which makes the routing very sequential.

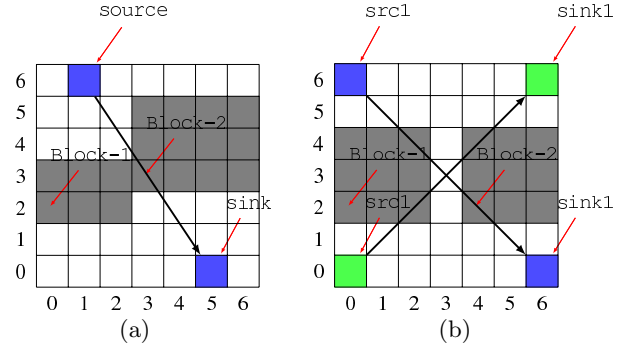


Figure 4: (a) Two modules block the path of the net. This placement is unroutable. (b) Although the path is not blocked. The two droplet has to coordinate which one should go first. Otherwise, they will form a deadlock in the critical path.

There are several papers that try to solve the placement problem, but so far, none of them focus on cross-referencing biochip. Su *et al.* presented a simulated annealing-based algorithm in [9]. They also try to take the fault tolerance issue into consideration by computing the ratio between fault-free cells and total cells. A unified synthesis algorithm is proposed by Su *et al.* in [10], where placement is one of the steps. In their paper, a parallel recombinative simulated annealing method which combines scheduling, resource binding and placement together is proposed. Later, they even include routing into the annealing process [11]. They claimed that the synthesis and placement of DMFB is very similar to the operation of dynamically reconfigurable FPGAs (DRFPGAs) [12]. Yuh *et al.* proposed several representation in simulated annealing-based method for the floorplanning problem, including *3D-subtgcg* [13] and *T-tree* [14]. Later they adopt the T-tree formulation to solve the placement problem [15]. This method is derived from the well-known 2-D placement method for reconfigurable devices. Defect cells are also modeled and considered in their work.

2.2 Our Contributions

In this paper, a placement method is proposed to solve the placement problem in cross-referencing biochip. We try to minimize the *extended covered area* of modules and the overall manhattan distance of routes. The major contributions of our work can includes:

- We propose the first method that solves placement problem in cross-referencing biochip. Comparing with previous papers, our method can generate suitable placement result for the cross-referencing biochip router.
- The electrode interference issue mentioned in Section 1.4 is addressed. The router in [7] is extended in order to satisfy the requirement and constraint. Our placement result is verified and compared with other methods using the modified router. Two criteria, including *stalling steps* and *manhattan difference* are introduced to evaluate the result.
- We evaluated the proposed method on the widely-used benchmarks *in-vitro*. Experimental result shows that

our method can produce better routing result comparing with the placement result generated by other placement methods, in terms of the overall routing time, stalling steps and cells used.

3. PROBLEM FORMULATION

In the architectural synthesis of DMFB, module placement is the step after task scheduling and resource binding. The placement problem can be viewed as a 3D-packing problem where the third dimension is time. The starting time of each operation corresponds to the z -value of the bottom plane of a specific module. Since the module size and time span has been determined during task scheduling and resource binding, the problem can be reduced to a series of 2D-packing problems at different time intervals. We will give a formal description of the problem as follows.

- Input:
 - Scheduling and resource binding result
 - Chip specification, including timing constraint, chip size optical detector number, reservoir/dispenser number
- Output:
 - Placement result, including locations of modules, reservoir and dispenser pin locations for each route
- Objective
 - Minimize the sum of the extended covered area, which will be explained in the next section

4. ROUTING-AWARE PLACEMENT FOR CROSS-REFERENCING BIOCHIPS

In this section, the proposed placement method is discussed. It mainly consists of three stages: dispenser and reservoir location generation, ILP formulation of the placement and pin assignment problem.

4.1 Dispenser and Reservoir Location Generation

This is a step to find the locations of the dispensers and reservoirs. The dispenser and reservoirs should be located on the boundary of the chip. They are not fixed before placement. However, once decided, their location is fixed during the bio-assay. Here, we do not put them into our Integer Linear Programming (ILP) formulation, but use a simple heuristic to place them. An important reasons for implementing this heuristic is that it is reasonable to distribute them using the proposed heuristic rather than giving them flexibility to locate arbitrarily on the boundary of the chip, since some of them may cluster together and make the routing regions congested. This will also reduce the optimality but according to our experiment, the final result is mainly decided by the placement result and not much related to these locations. However, it can significantly reduce the solution space and the overheads by solving the ILP. Following is the details of the algorithm.

First of all, the waste reservoir is placed at the center of one side. A set of cells on the boundary of the chip is then selected evenly according to the number of the reservoirs. Let

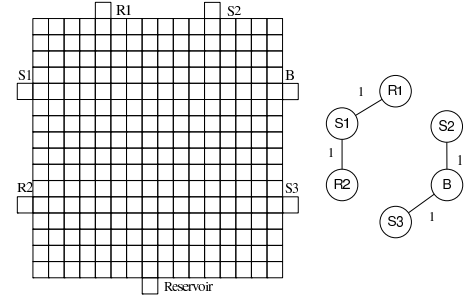


Figure 5: Dispenser/reservoir distribution in *in-vitro*.

there be N dispensers, we construct a graph with N vertices, where each vertex corresponds a dispenser. For each operation in the bioassay, we check whether it is formed by two dispensers, i.e., an operation between the two corresponding droplets. If yes, an edge is added between the two vertices with weight one. If there is already an edge between them, then the weight of the edge is incremented by one. We define the weight of a vertex as the sum of the weight of all the edges linking it with other vertices. After the graph is constructed, there may be several connected components (CC). Next, we arrange the dispensers CC by CC. For each CC, we picked the vertex u_1 that has the largest weight. Then, the vertices u_2, u_3, \dots , that have the largest weight and adjacent to u_1 are picked iteratively and are placed on each side of u_1 to form a list, $\dots u_2, u_1, u_3, \dots$. Finally, all such lists are concatenated. Dispensers then can be arranged to the boundary clockwise (or counter-clockwise) from the waste reservoir according to the final list. This is based on the intuition that for each list, the larger weight a vertex has, the closer to the component center it should be placed, such that the distance between it and the other dispensers are amortized. An example is given in Figure 5. It is from the *in-vitro* benchmark in Section 5.

4.2 ILP Formulation of Placement

We formulate an ILP to solve the module placement problem while considering the properties of cross-referencing biochip. The module placement problem in electronic design is known to be NP-hard [16]. In order to solve it efficiently, we try to model the problem with a scalable size of variables and constraints. The core idea in the formulation is the definition of the objective function. In the ILP, the formulation is the sum of a series of *extended covered area* formed by the routes in each subproblem. The *extended covered area* of a

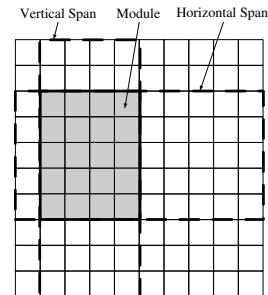


Figure 6: Illustration of extended covered area.

Table 1: Notations used in ILP formulation

M^i	Module i
M_x^i, M_y^i	Lower left coordinate of module M^i
$X(M^i)/Y(M^i)$	Width/Height of a mixing module i
W/H	Width/Height of the array.
$Center(M^i)$	Center point's coordinate of module M^i
L	A large constant
A_{ij}	Extended covered area bound by module/dispenser i and j

rectangle is defined as the vertical and horizontal area span, as illustrated in Figure 6.

Since in cross-referencing DMFB, the droplet movement is controlled by applying different voltages to row and column. If the extended covered area of the bounding box of a route is minimized, the route is shorter. Furthermore, it helps to reduce the interference between routes. The notation used in our ILP formulation is shown in Table 1. Note that the index is starting from 0.

4.3 Constraints

1. *Validity of modules:* The modules should be inside the chip. A further requirement is that the whole module including guarding ring should be at least one space away from the boundary of chip. This constraint is needed in order not to block the route from the dispenser nor to the reservoir. Note that this guarding ring can be shared between different modules. For module i , the above requirement is represented as:

$$M_x^i \geq 2 \quad (1)$$

$$M_x^i + X(M^i) \leq W - 2 \quad (2)$$

$$M_y^i \geq 2 \quad (3)$$

$$M_y^i + Y(M^i) \leq H - 2 \quad (4)$$

2. *Non-overlapping and separation of Modules:* For modules that co-exist at some time, they must not overlap with each other. Furthermore, there should be a separation cell around each module. For a pair of module M^i and M^j which co-exist at the same time, we have the following constraints:

$$(M_x^j < M_x^i - 1) \vee \quad (5)$$

$$(M_x^j > M_x^i + X(i)) \vee \quad (6)$$

$$(M_y^j < M_y^i - 1) \vee \quad (7)$$

$$(M_y^j > M_y^i + Y(i)) \quad (8)$$

Note that the *or* constraint cannot be directly supported in a linear program. Hence we need to *linearize* them. For this particular constraint, two more binary variables are introduced. Let them be c_1 and c_2 , the previous constraint is transformed into:

$$M_x^i - M_x^j + L(c_1 + c_2) > 1 \quad (9)$$

$$M_x^j - M_x^i - X(M^i) + L(c_1 + 1 - c_2) > 0 \quad (10)$$

$$M_y^i - M_y^j + L(1 - c_1 + c_2) > 1 \quad (11)$$

$$M_y^j - M_y^i - Y(M^i) + L(2 - c_1 - c_2) > 0 \quad (12)$$

where L is a large enough number (e.g., $W \times H$). It can be seen that among the four inequalities, only one of them will be left when the value of c_1 and c_2 are decided, others will be automatically satisfied due to the large value of L .

3. *Optical detector resource constraint:* For a set D of modules that are bound to the same optical detector, we have:

$$(M_x^i, M_y^i) = (M_x^f, M_y^f), i \in D \quad (13)$$

where M^f is a module in D that first appear on the time line.

4. *Bounding box of routes:* We use the bounding box as an estimation of the routes generated. Suppose that there is an operation, in which droplet i introduced by some module M^i is needed to be routed somewhere to form as input to M^j , we use the center points of both modules to form a bound box to model the route. Let $Center(M^i) = (x_i, y_i)$ and $Center(M^j) = (x_j, y_j)$, two pairs of variables, (x_{ll}, y_{ll}) and (x_{ur}, y_{ur}) , are introduced to denote the bottom left corner and upper right corner of the bounding box formed by both center points.

$$\begin{aligned} x_{ll} &\leq x_i, & x_{ll} &\leq x_j \\ y_{ll} &\leq y_i, & y_{ll} &\leq y_j \\ x_{ur} &\geq x_i, & x_{ur} &\geq x_j \\ y_{ur} &\geq y_i, & y_{ur} &\geq y_j \end{aligned} \quad (14)$$

Then, we can compute the extended covered area bound by this route, denoted as A_{ij} :

$$A_{ij} = W(x_{ur} - x_{ll}) + H(y_{ur} - y_{ll}) \quad (15)$$

5. *Objective:* The objective function is the sum of all the extended covered area:

$$\min : \sum_k A_k \quad (16)$$

where $A_k, k = 1, 2, \dots$ is a set of all the extended covered area in the subproblems.

4.4 Pin Location Generation

After the exact locations of the modules are obtained by solving the above ILP formulation, the result is not complete enough for routing. According to the placement result from [15], we also implement according to the following rules:

- Mixing module is modeled as a 3-pin net;
- Dilute module is modeled as two 2-pin net;

- Optical detection and storage are modeled as a 2-pin net.

The source pin and sink pin is specified around the modules. Here, we assume that the input and output pin is specified in module description. For instance, for a mixing module, the droplet can enter or leave the mixing area from one of its four side. Note that *fluidic constraint* in [5] must be carefully satisfied. For example, the pin should not overlap with a dispenser or a waste reservoir.

Also, after some operations, e.g., optical detection, some droplets should be routed to the waste reservoir. This is specified in the bio-assay protocol.

5. EXPERIMENTAL RESULTS

The proposed method is implemented in C++ programming language, and the ILP part is solved using lp_solve 5.5 [17] and run on a 2.4GHz Intel Core II Duo machine with 1.5GB memory. We limit the ILP solver to run in a given amount of time. The time limit is set to 15 minutes here, since from our empirical experiments, the quality of the solution is almost fixed and almost no increase. For comparison purpose, we implement the router proposed in [7], and improved it by including the new constraints described in Section 1.3.

We evaluated the proposed method with two sets of the multiplexed *in-vitro* diagnostics [4]. Table 2 summarizes the comparisons. In the experiment, we define the Stalling Step (SS) as the total number of stalling in the route that a droplet has taken. Stalling means that at a certain step, a droplet has to wait for the path to be cleared, either for waiting some other droplet to pass through, or by queuing outside the waste reservoir, etc. The root reason of stalling is from the fluidic constraint, or more possibly, from the electrode constraint for cross-referencing routing. The Manhattan Difference is defined as the sum of the differences between the actual distance a droplet has moved and the Manhattan Distance from the source to the sink. Because the routing has a third dimension of time, rather than the traditional 2D routing, the Manhattan Difference itself is not sufficient to describe the ‘lag’ of droplets. Max/avg cycle refers to the maximum completion time and the average completion time of a benchmark. For comparison, we run the router on the placement generated by [15] and on the placements generated by our approach.

We can see from the result that by using the placement result generated by our proposed method, the Max/avg cycles have been improved. The Stalling steps and the Manhattan Difference are also reduced. Moreover, it is shown that the number of cell used has been reduced by more than a half comparing with the routing result of [15]. According to the experiment result of max/avg cycle, we can see that the router can finish routing in a relative shorter time, which indicates shorter route and less congestion are achieved. The experimental result demonstrates that our placement result can give better routability and simpler routing configurations to the router.

Here we use one instance of the *in-vitro* diagnostics as an example to illustrate the proposed method. Figure 7 and Figure 8 are the sequence graph and the scheduling result of the multiplexed *in-vitro* diagnostics with 3 samples, 2 reagents and 1 buffer. As in [15], we also assume that

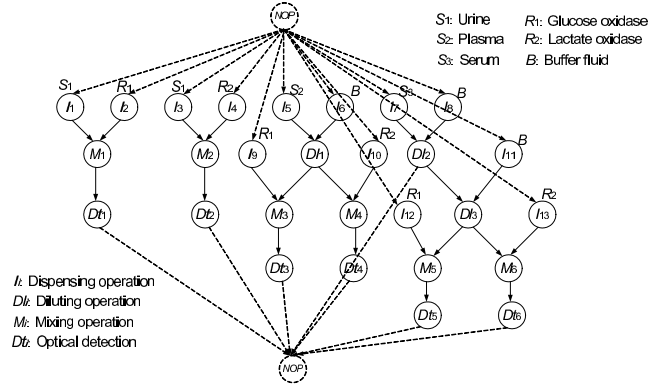


Figure 7: Sequencing graph of *in-vitro* [5].

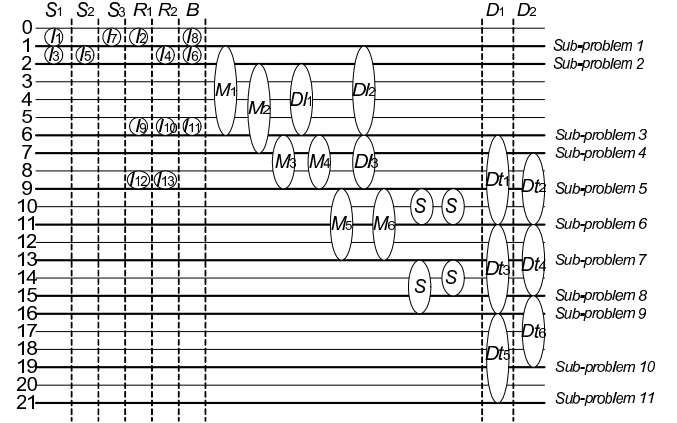


Figure 8: Schedule result obtained from paper [5].

for each type of sample, reagent and buffer, there is one dedicated on-chip reservoir/dispenser. Size of the modules are determined in the previous design stage.

Due to the extended covered area constraint, the ILP formulation gives a highly compacted placement and shorter route between droplets. Firstly, our approach saves more space on chip, which might be wasted if modules are loosely placed. Secondly, our approach is beneficial to cross-referencing routing, since the shorter a route is, the less possible that a droplet might impose electrode constraint on others (remember that the activation of electrodes are in a row-column manner).

6. CONCLUSION AND FUTURE WORK

An ILP-based placement method is presented in this paper. The proposed method tries to minimize the extended covered area and droplet route. This method considers the property of cross-referencing biochip, and can generate placement result that is suitable for cross-referencing biochip routing. Several real-world bioassay cases are used to test the proposed method. The maximum/average cycle time, stalling steps, Manhattan difference and the number of cells used are used as the criteria of placement. The experiment result demonstrates the efficiency and effectiveness of the method. By combining the effort of the placement tool and router, the flexibility of cross-referencing DMFB can be further exploited. More experiments will be conducted on the bench-

Table 2: Routing [7] result upon placement [15] and upon our placement.

Name	# sub. ^a	Routing result of [15]				Routing result of our placement			
		Max/Avg cycle	SS. ^a	MHT diff.	cell used	Max/Avg cycle	SS. ^a	MHT diff.	cell used
<i>in-vitro</i>	11	20/12.09	12	45	246	16/9.64	3	19	151
<i>in-vitro2</i>	15	19/10.73	23	39	250	17/6.40	5	19	104

^aNumber of subproblems

^bStalling steps

marks including the protein bioassay, and more contemporary routers will be tested, e.g., router in [18] and in [6]. In the future, this work is expected to be expanded and included into the automated design tool chain for cross-referencing biochips, which will facilitate scheduler and router to provide dedicate effort in designing cross-referencing biochip.

7. REFERENCES

- [1] M. G. Pollack, A. D. Shenderov, and R. B. Fair, "Electrowetting-based actuation of droplets for integrated microfluidics," *Lab on a Chip*, vol. 2, no. 2, pp. 96–101, May 2002.
- [2] E. Verpoorte and N. F. De Rooij, "Microfluidics meets mems," *Proceedings of the IEEE*, vol. 91, no. 6, pp. 930–953, 2003.
- [3] J. Ding, K. Chakrabarty, and R. Fair, "Scheduling of microfluidic operations for reconfigurable two-dimensional electrowetting arrays," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 12, pp. 1463–1468, 2001.
- [4] F. Su and K. Chakrabarty, "Architectural-level synthesis of digital microfluidics-based biochips," in *Proc. International Conference on Computer-Aided Design*, Nov. 7–11, 2004, pp. 223–228.
- [5] F. Su, W. Hwang, and K. Chakrabarty, "Droplet routing in the synthesis of digital microfluidic biochips," in *Proc. Design, Automation and Test in Europe*, vol. 1, Mar. 6–10, 2006, pp. 1–6.
- [6] P.-H. Yuh, S. Sapatnekar, C.-L. Yang, and Y.-W. Chang, "A progressive-ILP based routing algorithm for cross-referencing biochips," in *Proc. Design Automation Conference*, 2008, pp. 284–289.
- [7] Z. XIAO and E. F. Young, "Crossrouter: A droplet router for cross-referencing digital microfluidic biochips," *Proc. Asia and South Pacific Design Automation Conference*, Jan. 2010, to be published.
- [8] M. Sarrafzadeh and C. Wong, *An introduction to VLSI physical design*. McGraw-Hill Higher Education, 1996.
- [9] F. Su and K. Chakrabarty, "Design of fault-tolerant and dynamically-reconfigurable microfluidic biochips," in *Proc. Design, Automation and Test in Europe*, 2005, pp. 1202–1207.
- [10] F. Su and K. Chakrabarty, "Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips," in *Proc. Design Automation Conference*. ACM New York, NY, USA, 2005, pp. 825–830.
- [11] T. Xu and K. Chakrabarty, "Integrated droplet routing in the synthesis of microfluidic biochips," in *Proc. Design Automation Conference*. ACM Press New York, NY, USA, 2007, pp. 948–953.
- [12] K. Bazargan, R. Kastner, and M. Sarrafzadeh, "Fast template placement for reconfigurable computing systems," *IEEE Design and Test of Computers*, vol. 17, no. 1, pp. 68–83, Jan. 2000.
- [13] P.-H. Yuh, C.-L. Yang, Y.-W. Chang, and H.-L. Chen, "Temporal floorplanning using 3D-subTCG," in *Proc. Asia and South Pacific Design Automation Conference*, Jan. 27–30, 2004, pp. 725–730.
- [14] P. Yuh, C. Yang, and Y. Chang, "Temporal floorplanning using the T-tree formulation," in *Proc. International Conference on Computer-Aided Design*, 2004, pp. 300–305.
- [15] P. Yuh, C. Yang, and Y. Chang, "Placement of defect-tolerant digital microfluidic biochips using the t-tree formulation," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 3, no. 3, p. 13, 2007.
- [16] M. Garey, D. Johnson, R. Backhouse, G. von Bochmann, D. Harel, C. van Rijsbergen, J. Hopcroft, J. Ullman, A. Marshall, I. Olkin *et al.*, *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman San Francisco, 1979.
- [17] Ipsolve homepage. [Online]. Available: <http://lpsolve.sourceforge.net>
- [18] T. Xu and K. Chakrabarty, "A Droplet-Manipulation Method for Achieving High-Throughput in Cross-Referencing-Based Digital Microfluidic Biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 11, p. 1905, 2008.