A Reliability-Oriented Placement Algorithm for Reconfigurable Digital Microfluidic Biochips Using 3-D Deferred Decision Making Technique

Ying-Han Chen, Chung-Lun Hsu, Li-Chen Tsai, Tsung-Wei Huang, and Tsung-Yi Ho, *Senior Member, IEEE*

***Abstract*—In recent studies, digital microfluidic biochips (DMFBs) have been a promising solution for lab-on-a-chip and bio-assay experiments because of their flexible application and low fabrication cost. However, the reliability problem is an imperative issue to guarantee the valid function of DMFBs. The reliability of DMFBs decreases when electrodes are excessively actuated, preventing droplets on DMFBs controlled successfully. Because the placement for bio-assays in DMFBs is a key step in generating corresponding actuating signals, the reliability of DMFBs must be considered during biochip placement to avoid excessive actuation. Although researchers have proposed several DMFB placement algorithms, they have failed to consider the reliability issue. In addition, previous algorithms were all based on the simulated-annealing (SA) method, which is time consuming and does not guarantee to obtain an optimal solution. This paper proposes the first reliability-oriented non-SA placement algorithm for DMFBs. This approach considers the reliability problem during placement, and uses the 3-D deferred decision making (3D-DDM) technique to enumerate only possible placement solutions. Large-scale DMFB placement can be synthesized efficiently by partitioning the operation sequential graph of bioassays. Experimental results demonstrate that the proposed technique can achieve reliability-oriented placement for DMFBs without excessive actuation in each electrode, while optimizing bioassay completion time.**

***Index Terms*—3-D placement, digital microfluidic biochips, reliability.**

I. Introduction

DROPLET-BASED digital microfluidic biochips (DMFBs) have received much attention in laboratory experiments due to their portability, high throughput, high sensitivity, limited human intervention, and low sample volume consumption [3]. Practical bio-assays, such as clinical diagnostics, DNA analysis, environmental toxin monitoring, and drug discovery, have been successfully realized on DMFBs. The DMFB users can obtain bio-assay results in a very short execution time by separating liquids into discrete droplets with flexibly controlled signals [4], [12].

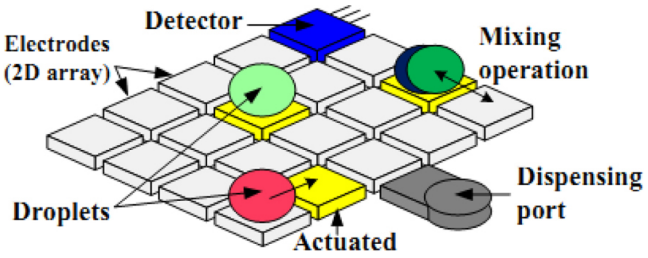


Fig. 1. Schematic view of a DMFB.

A DMFB consists of a 2-D electrode array and peripheral devices (e.g., optical detector, dispensing port) [12] (Fig. 1). Droplets are controlled by underlying electrodes using electrical actuations to generate electrowetting force (i.e., a principle called electrowetting-on-dielectric or EWOD) [8]. By assigning time-varying control signals to actuate and de-actuate electrodes, droplets can be moved around the entire 2-D array to perform fundamental operations (e.g., dispensing and mixing). These operations are carried out under clock control in a reconfigurable manner because of their flexibility in spatial and time domain [1].

In realizing DMFBs, the correctness of the bio-assay results on DMFBs is a crucial requirement, especially for medical applications [8], [1]. Bio-assays are realized on DMFBs by binding, scheduling, and placing each bio-assay operation on corresponding electrodes of DMFBs. To guarantee the high throughput rate and correctness of bio-assays, the reliability of electrodes in DMFBs must be considered, while minimizing the bio-assay completion time. However, the first reliability problem occurs when parts of electrodes are actuated more frequently. The problem gets worse in previous placement algorithms [10], [15], which are all focused on only minimizing bio-assay completion time, and some electrodes in a DMFB placement result may be actuated many more times (highly actuated) than others (lowly actuated). For example, Fig. 2 shows

Manuscript received August 27, 2012; revised December 22, 2012 and February 9, 2013; accepted February 10, 2013. Date of current version July 15, 2013. The work of T.-Y. Ho was supported in part by the Taiwan National Science Council under Grants NSC 101-2220-E-006-016 and NSC 101-2628-E-006-018-MY3 and the Ministry of Education, Taiwan, under the NCKU Aim for the Top University Project Promoting Academic Excellence and Developing World Class Research Centers. This paper was recommended by Associate Editor Y. Chen.

Y.-H. Chen, L.-C. Tsai, and T.-Y. Ho are with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: locke2833@gmail.com, mvptsai@gmail.com, tyho@csie.ncku.edu.tw).

C.-L. Hsu is with the University of California, San Diego, CA 92093, USA (e-mail: doublesluen@gmail.com).

T.-W. Huang is with the Department of Electrical and Computer Engineering, University of Texas, Austin, TX 78713, USA (e-mail: twh760812@gmail.com).

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

Digital Object Identifier 10.1109/TCAD.2013.2249558

0278–0070/$31.00 © 2013 IEEE

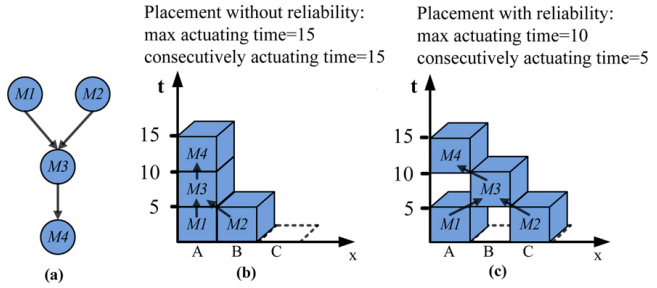


Fig. 2. (a) Sequential graph of a bio-assay with four operations. (b) Corresponding placement of the bio-assay without reliability considerations, and one EA electrode *A* and maximum actuating time at electrode *A*. (c) Corresponding reliability-oriented placement without EA electrode and minimized the maximum actuating time.

two placements for the same bio-assay, and the bio-assay completion times are the same for both placement results. The maximum actuating time of the two placement results for each electrode both occurs in electrode *A*, but the maximum actuating time in Fig. 2(b) is higher than 2(c). Maximum actuating time in Fig. 2(b) is reduced to two-thirds compared to 2(c). Electrode *A* in (b) may break down sooner, causing DMFB failure. Hence, one goal of a reliability-oriented DMFB placement is to minimize the maximum actuating times of all electrodes in a DMFB placement (i.e., to average actuating times of all electrodes).

The second reliability issue is the problem of trapped charges in the dielectric insulator or the actuated electrode metal layer of the chip [10], [12]. This charge problem is the result of excessively actuated (EA) electrodes, which consecutively actuate electrodes without restriction (i.e., electrodes are actuated for too long time without de-actuating). The charge problem eventually causes permanent dielectric degradation [2]. In this scenario, droplet movements are unpredictable because droplets cannot be moved in expected directions. Therefore, the reliability of a DMFB is significantly degraded by EA electrodes. For example, the electrode *A* in Fig. 2(a) is an EA electrode because *A* is actuated for three consecutive times. Fig. 2(b) provides a reliability-oriented placement solution without EA electrodes. Hence, the two major goals of synthesizing reliability-oriented placements for DMFBs are to minimize the maximum actuating time of each electrode and to avoid EA electrodes while minimizing the bio-assay completion time. This approach gaurantees high throughput and correctness of DMFBs simultaneously.

However, only one paper has addressed the reliability problem to eliminate the existence of EA electrodes [5]. Although additional EAs are avoided at this stage based on the results of DMFBs' placements, the inherent EA problem from placement remains unsolved. Highly actuated electrodes represent another reliability issue in the previous paper. The solution of avoiding the reliability problem is to minimize the maximum actuating times, and to eliminate EA electrodes during DMFB placement stage, which is the first stage of synthesizing DMFB.

Current DMFB placement methods [6], [10], [11], [15] have introduced the reliability problem inherently by only minimizing the bio-assay completion time. Hence, the reliability-oriented placement algorithm for DMFB is an essential requirement to ensure the correctness of DMFB applications. This paper proposes a novel reliability-oriented placement algorithm to tackle this problem efficiently and effectively.

This paper offers the following contributions.

1) Reliability-oriented placement algorithm: This paper considers the causes and factors that affect reliability and incorporate the attributes that are favorable for reliability improvement during synthesizing placements.
2) 3-D deferred decision making (3D-DDM) algorithm for 3-D placement: This paper proposes the first non-SA 3-D placement algorithm, called 3D-DDM, for DMFBs. By enumerating possible module combinations and merging the combinations into several feasible placements, this paper can obtain 3-D placement results with minimized assay completion time. Moreover, in order to speed up the computation time, this paper also proposes a novel technique that prunes most redundant computations without affecting solution quality.
3) Minimize the maximum electrode actuating time algorithm: After obtaining several placement results from 3D-DDM, this paper uses two-stage adjustment method and dynamic programming technique to rearrange the position of each module in placements. This approach minimizes the maximum electrode actuating time.
4) Multilevel partition: This paper uses the multilevel partition method on the operation graph of bio-assays to enhance the 3D-DDM efficiency. We first map the original bio-assay graph into a tree structure to reduce the computing complexity. Next, we can efficiently handle the 3-D placement problem by dividing large tree structures into small multilevel groups with minimized edge cuts among each group.

This paper demonstrates the effectiveness of the proposed reliability-oriented 3-D placement algorithm using protein and several kinds of in-vitro assays. Compared with previous DMFB placement algorithms, this proposed algorithm can easily solve the reliability issue, while synthesizing the DMFB placement. Experimental results show that this algorithm can satisfy the reliability consideration and simultaneously optimize bio-assay completion time for DMFBs.

The remainder of this paper is organized as follows. Section II presents the related preliminaries, including reliability issue and the conventional 2-D deferred technique. Section III formulates the reliability-oriented placement problem. Section IV presents the proposed 3-D defer decision making (3D-DDM) placement algorithm, and related methods to enhance the reliability of DMFB placement. Finally, Sections V and VI present experimental results and the conclusion.

II. Preliminaries

A. Review of Reliability-Oriented Design Model

When using a DMFB, correct droplet operations must be ensured to obtain a valid result for bio-assays. However, the reliability problem is encountered when any droplet cannot be controlled by electrodes. This section highlights the causes of

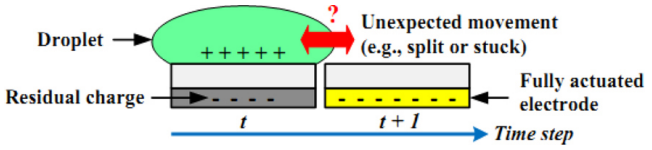


Fig. 3. Residual-charge phenomenon.

degrading reliability and mention the solution of corresponding problems.

1) *Minimizes the Maximum Actuating Time of Electrodes:* The optimizing objective of conventional DMFB placement algorithms is to minimize the bio-assay completion time. However, the compact 3-D placement results from previous algorithms imply that some electrodes are actuated more times (highly actuated) than others (lowly actuated). According to the physical characteristics and fabricating process of electrodes [5], [8], each electrode in a DMFB array is identical. Thus, the lifetime of the DMFB becomes shorter dramatically when some electrodes are highly actuated. Hence, the maximum actuated time of each electrode must be minimized in each DMFB to extend the lifetime of electrodes, while minimizing the bio-assay completion time.

2) *Charge Problem in Excessive Actuated Electrode:* Droplets may be stuck on an electrode or even be wrongly split when charges accumulate on an electrode because of excessive actuation [8], [13] without proper deactivation.

The charge problem occurs when an electrode is actuated over a certain successive period and charges accumulate on the electrode [8], [10] (Fig. 3). In this condition, the behavior of droplets cannot be predicted because of unknown charges exist in the electrode. Although several methods can be used to overcome this problem, they are all based on the results of placement. Physical experiments have shown that only part of the accumulated charges can be removed by additional grounding time [8], preventing the bio-assay from completing in a reasonable time. Therefore, good DMFB placement must be provided to avoid any EA electrodes, while simultaneously minimizing the bio-assay completion time.

Depending on the causes of EA electrode, the reliability can be enhanced during synthesis DMFBs by computer-aided design tools. Because the placement is the first stage in generating the actuating sequences for each electrode, we can guarantee no EA condition occurs in each electrode. Hence, this paper adopts the reliability constraint function $f(r)$ in [8] to show that every continuous executing operation for r seconds in every electrode should be followed by deactivation for at least $f(r)$ seconds. The function $f(r)$ was experimentally set as $f(r) = r/11 + 1$ to avoid charge problems [5], [8].

B. Review of 2-D Deferred Decision Making Technique

Because the DMFB placement problem is similar to the 3-D fixed-outline placement problem, which is an extending problem of 2-D placement in the x–y plane with an additional dimension of t-axis (time domain axis), it is possible to apply 3-D placement algorithms to DMFB placement. However, previous DMFB placement algorithms were implemented using the simulated-annealing (SA) method [11], [15], which is time consuming and cannot guarantee an optimal solution. A non-SA placement algorithm, called the 2-D deferred decision making (DDM) technique, was proposed [14] to obtain an optimal solution efficiently. Unfortunately, the time complexity of that approach increases exponentially with the third dimension when applying 2D-DDM to the 3-D DMFB placement problem. Thus, this paper proposes a novel 3D-DDM algorithm to efficiently obtain reliability-oriented placement.

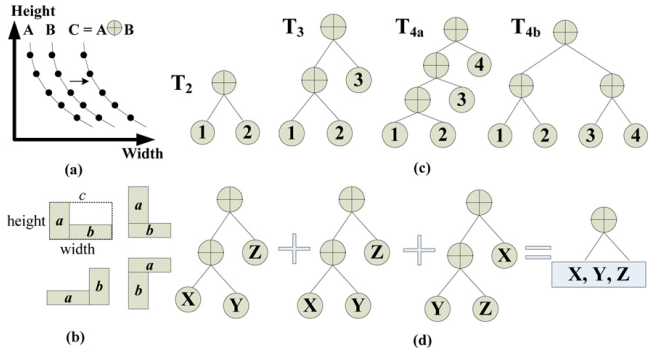


Fig. 4. (a) Obtaining the 2-D curve C from merging curves A and B. (b) 2-D curve C is part of combination results by merging each block from 2-D curves A and B in every possible different direction. (c) List of all combinations of slicing tree structures of T_2, T_3, and T_4. (d) Combining all results with enumerations from the same node into a single node.

Before introducing the 3D-DDM, this section reviews the conventional 2D-DDM algorithm [14], which enumerates every possible module placement combination by dynamic programming and stores the results in a slicing tree structure.

1) *Slicing Tree Structure:* To handle the 2-D placement problem easily, the 2D-DDM algorithm uses a slicing tree structure, in which each leaf denotes a module with all possible orientations, and each internal node denotes how the two modules associated with its child nodes are merged, including horizontal and vertical merges. An internal node in the tree enumerates all possible ways to merge the two subfloorplans associated with its child nodes, and store results in the form of monotonically decreasing curve, known as a 2-D curve. This step significantly decreases computation complexity because only smaller width (height) of merged results with the same height (width) are stored in the 2-D curve. By constructing the slicing tree from leaves to root, more than one 2-D placement result can be obtained and stored in the root of slicing tree, and one optimized placement solution is chosen from the root of the slicing tree.

2) *Enumerative Placements:* To achieve optimal placement, the 2D-DDM algorithm enumerates all block permutations and stores them in a slicing tree. The efficient enumerative packing technique proposed in [15] uses dynamic programming to enumerate all possible slicing permuting structures and to build up the final 2-D shape curve. Fig. 4(a) and (b) represents how two blocks a and b are merged to form a new block c, and new merged 2-D curve C from A and B. Fig. 4(c) also shows the enumerating examples from two to four nodes (i.e., modules in a slicing tree). Because the generalized slicing tree does not differentiate the left-right order, the number of slicing trees is reduced. For example,

there are only two slicing trees with four leaf nodes: T_{4a} and T_{4b} in Fig. 4(c). The advantage of dynamic programming with the data structure of the slicing tree significantly reduces the computation complexity of 2D-DDM [15].

III. Problem Formulation

Based on previous discussion, the reliability-oriented placement problem for DMFB can be summarized as follows.

Input:

1) A module library: The library contains modules for each type of operation, such as mixing and dilute, with different dimension (i.e., width, length, and executing duration).
2) Bioassay protocol: Given an acyclic graph $G = (V, E)$, where the vertex set $V = \{v_1, \cdots, v_m\}$ and the edge set $E = \{e_1, \cdots, e_n\}$ denote the operations and the precedence relations between two operations. The graph G is also called a sequential operation graph.
3) Design specifications: Given the maximum allowable execution time, T, the size of microfluidic array, $W \times H$, and the available number of each type of resources as design constraints.

Constraint:

1) Reliability constraint: An electrode cannot be activated for more than the limited continuously time period to prevent the charge problem.
2) Precedence constraint: The precedence relationship exists between two vertices v_i and v_j. If v_i is a predecessor of v_j in the sequencing graph, v_j can only start after v_i is finished.
3) Resource constraint: Resources include the dispensing port of the specific regent buffer or sample and the optical detector. Based on the available number of resources, there cannot be more than the available number of each type of resource at the same time. In addition, the optical detector is a special resource because its location is decided at the fabrication stage and is fixed.
4) Storage constraint: After all the operations are scheduled and bound into a placement, the storage unit should be inserted between the precedence v_i and succession v_j to guarantee a space for storing the droplet if the successive operation v_j does not start right after the finish of the precedent operation v_i.
5) Fixed-outline constraint: The area occupied by all the modules cannot exceed the dimension of the chip.
6) Nonoverlapping constraint: At any time, no two modules can overlap.

Objective: To synthesize a reliability-oriented placement for DMFBs by scheduling the bio-assay operation, binding assay operation to resources, and creating a 3-D fixed outline layout with optimized biochip assay completion time while minimizing the maximum electrode actuating time.

Fig. 5 shows the 3-D placement flow for DMFBs and a corresponding example.

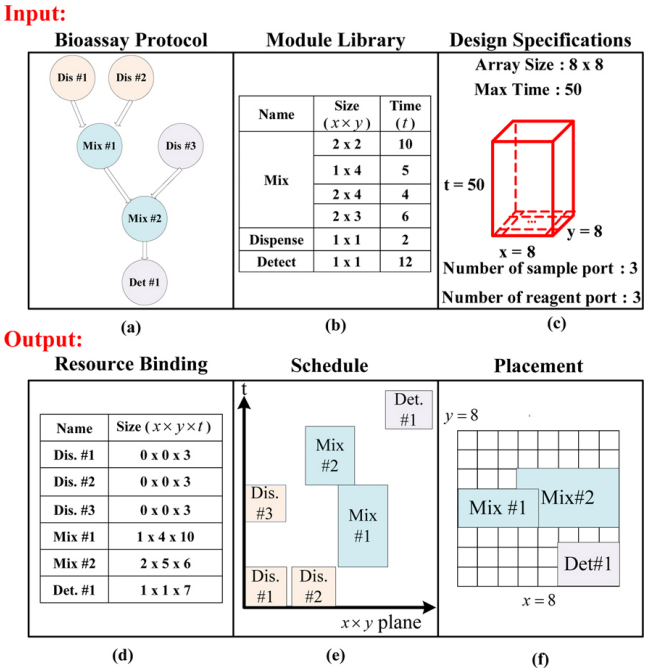


Fig. 5. DMFB placement synthesizing flow. (a) Sequencing graph for a bioassay protocol. (b) Module library. (c) Design specification. (d) Resource binding results. (e) Scheduling of modules. (f) Placement results of modules.

IV. The Proposed Algorithms

The complexity of the 3-D placement problem for DMFB is increased with reliability constraint while optimizing assay completion time, minimizing the maximum electrode actuating time, and avoiding EA electrodes. An intuitive method to incorporate the reliability issue into a given placement results, which can be obtained using recent methods [11], [15], seems to be workable. However, this intuitive method encounters several serious drawbacks, particularly in terms of minimizing assay execution time. We list two major deficiencies as follows.

1) Post incorporating the reliability constraint may insert many deactivated units into a given placement result to solve the charge problem. However, the operation starting time of some modules in this placement must be postponed to produce extra spaces for these deactivated units. Unfortunately, the bio-assay completion time with post processing then becomes much longer than the original one [5].
2) The 3-D placement algorithms in $X \times Y \times T$ should be modified to satisfy the reliability constraint and to minimize the maximum electrode actuating time in each SA iteration. Thus refined algorithms are too complicated to obtain a feasible solution within an acceptable algorithm executing time.

Thus, this paper proposes the 3D-DDM algorithm to obtain the optimized reliability-oriented 3-D placement. First, take operation sequential graph of a bio-assay as input and transfer the graph into a tree structure. Then, partition the graph into multi-level groups by considering the precedent constraint carefully. Next, we propose the 3D-DDM to enu-

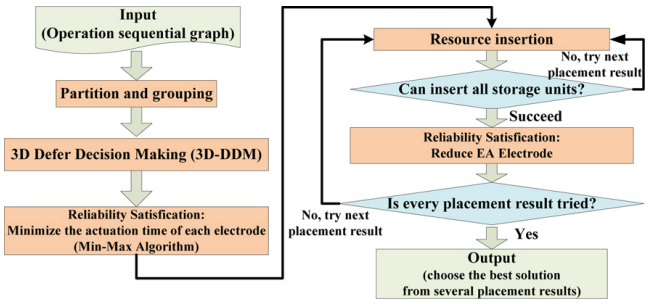


Fig. 6. Overall flow of reliability-oriented placement algorithm for DMFB.

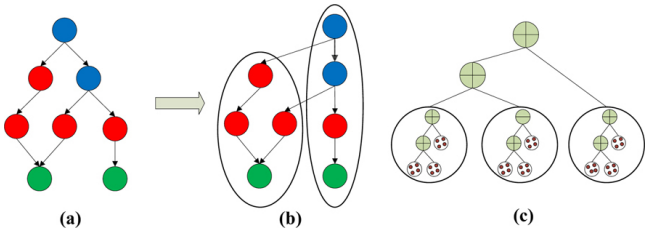


Fig. 7. (a) Original operation sequential graph. (b) Graph is partitioned into two groups with minimum number of precedence constraint (edge cut). (c) Multilevel group result.

merate all possible placements efficiently. During 3D-DDM, satisfy precedence constraint and preserve the storage space simultaneously. The two-step min-max method then minimizes the maximum actuating time of each electrode in DMFBs. Compared with only one possible solution from SA method, 3D-DDM obtains several possible 3-D enumerated placements, and we try to insert resource modules and storage units into the proper spaces in all possible 3-D placements. These modules can insert into 3-D placements easily because they need only small spaces compared to reconfigurable modules. Finally, we can avoid EA electrode and choose the reliability-oriented 3-D placement with optimized assay completion time. Fig. 6 shows the overall flow of the proposed algorithm, and each step is explained in the followings.

A. Partition and Group Operation Graph

After obtaining an operation sequential graph from the bio-assay as an input, use 3D-DDM for a DMFB placement. However, because more than one complicated bio-assay is integrated in one DMFB, the number of operation increases hugely. The protein bio-assay is an example of a large size of operation graph, with 103 operations (internal nodes) [9]. Although the dynamic programming technique is implemented when enumerating all possible placement results, the number of enumerations still grows exponentially [14]. Hence, we propose a multi-level partition method, a divide-and-conquer techniques uses recursively divide groups with a huge number of operations into smaller groups. Thus, the placement problem can be solved hierarchically from low-level groups to top-level groups.

During partition step, the precedence relationships among operation pairs must be considered carefully because the complexity of verifying precedence constraints increases when more precedent relationships exist among groups. On the other hand, groups are formed with the fewest precedent

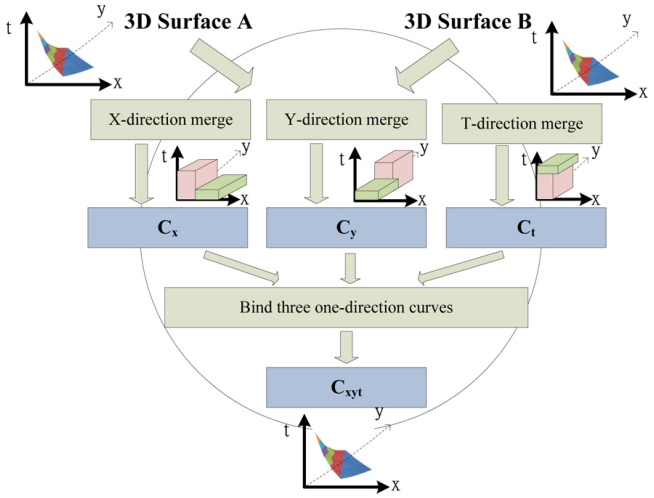


Fig. 8. Merge flow of 3-D generalize slicing tree in the proposed 3D-DDM.

constraints among groups, so the complexity of verifying the precedent constraint is minimized by the 3D-DDM algorithm (Fig. 7). Because of the operation sequential graph of bio-assays, cut one of the two branches of dilution operation because only dilution operations have more than one branch. Moreover, when dividing the sequential operation graph, cut the least number of edges in the graph to form approximately equal sub-trees recursively until each size of sub-trees is solvable.

B. 3-D Deferred Decision Making Technique

Next, the grouped operation graph is used as the input of the 3D-DDM to minimize the DMFB assay completion time. Enumerating the permutations of all three directions directly is the conventional method of transferring the DDM technique from 2-D to 3-D. However, the overhead of this method is unacceptable because the time complexity grow from $O(NlogN)$ to $O(N^4)$ [14], where N is the total number of modules (operations) of a placement problem. The increased complexity is caused by introducing the third axis.

This paper proposes an efficient implementation of 3D-DDM to divide the 3-D merging procedure into three individual directions, and further prevent redundant merging and verifying trials when preserving smaller size of merged permutations. Fig. 8 shows the 3D-DDM algorithm divided into three steps:

1) create three one-direction 3-D surfaces C_x, C_y, C_z, by merging two modules only according to x-, y-, and z-direction, respectively;
2) combine the three 3-D surfaces to form a complete 3-D surface C_{xyz}, which contains all results of three direction merging;
3) obtain all possible placement results for the DMFB by constructing the slicing tree from bottom to top.

Store our solutions as 3-D points, which contains the information of the new merged module size and module's position in 3-D space. These 3-D points form a 3-D surface, the same concept of 2-D curve, with the smaller 3-D module size as possible solutions from 3D-DDM. Hence, the root of the

Algorithm 1: Conventional One-direction 3-D Surface Merge Algorithm

Input : two 3-D surfaces A and B

```
1  begin
2      Create null new 3-D surface C
3      while Not all pair(a_i,b_j) are tried do
           /* a_i ∈ A and b_j ∈ B                    */
4          Create a point c' by combining choosing the pair (a_i,b_j)
5          while Not all points c_k in C are checked do
6              if c'_x ≥ c_kx and c'_y ≥ c_ky and c'_z ≥ c_kz then
7                  Discard c'
8              else if c'_x < c_kx and c'_y < c_ky and c'_z < c_kz then
9                  Discard c_k from Surface C
10         if c' is not discarded then
11             Add c' into Surface C
12 end
```

Output: A merged 3-D surface C

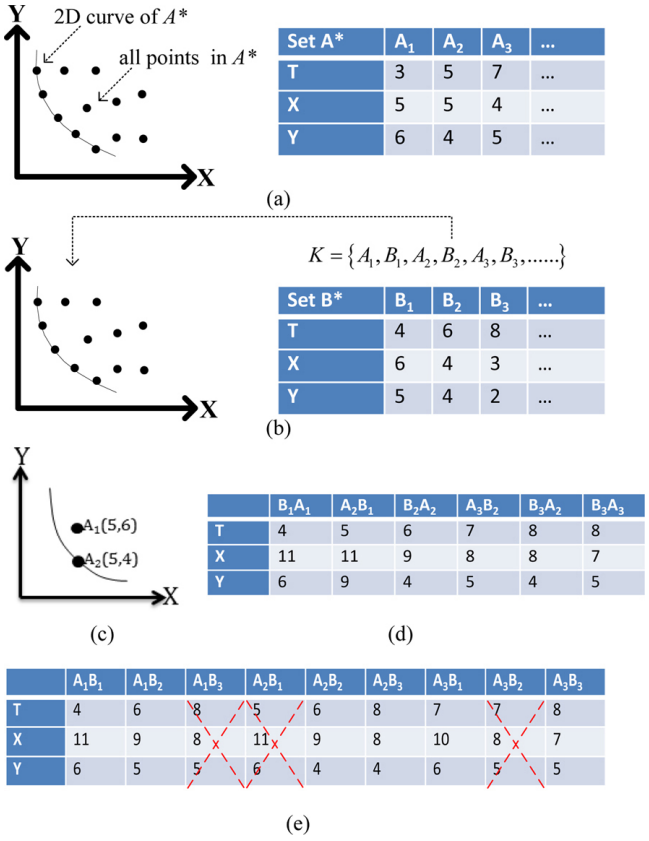
slicing tree is a 3-D surface with all possible DMFB 3-D placement results.

1) *Step 1: One-Direction 3-D Surface Merge:* When two child nodes (3-D surfaces) A and B are merged into one new one-direction 3-D surface C, a one-direction 3-D surface C is created by merging points $a \in A$ and $b \in B$ into a new point c for only one certain merging direction (Fig. 8). The conventional merging method shown in Algorithm 1 creates a new merged point c' by merging a and b, and the point c' either adds to 3-D surface C (lines 12 and 13) or replace previous added point c_k (lines 8–9) in C. Obviously, the conventional algorithm needs unnecessary computations for these kind of redundant points c_k.

The concept of our proposed algorithm is to prevent the redundant points adding into the merged 3-D surface C. Because most merging points c' have larger sizes and will not exist in the new 3-D surface C, the total number of points in C, N_{cut}, is much smaller than the total number of point, N, in A and B. Hence, the time complexity of our proposed algorithm is reduced dramatically from $O(N^4)$ to $O(N_{cut}^4)$. The proposed one-direction 3-D surface merge method is much more efficiency than conventional method.

This section explains our one-direction 3-D surface merge algorithm in Algorithm 2. First, define the reference axis, which is chosen from one of the other two nonmerging directions, to simplify the merging step. This paper uses the terms t-, t-, and x-axis as the reference axes for the x-, y-, and z-direction merge, respectively, for later merging.

This paper uses the x-direction 3-D surface merge to explain Algorithm 2. A new array K consists of all points in A and B, and points $k_i \in K$ are sorted in the increasing order of reference axis (t-axis) (line 4). Next, choose k_i from K accordingly, and add the point k_i to set A^* (or B^*) if k_i is a point from A (or B). The set A^* (or B^*) consists of points in A and B regardless the reference axis, and 2-D curves can be found in set A^* and B^* without considering t-axis. Because point k_i is chosen from K in the increasing order of reference axis (t-axis), the 2-D curves and their set A^* and B^* have following features:



Set A*	A_1	A_2	A_3	...
T	3	5	7	...
X	5	5	4	...
Y	6	4	5	...

Set B*	B_1	B_2	B_3	...
T	4	6	8	...
X	6	4	3	...
Y	5	4	2	...

	B_1A_1	A_2B_1	B_2A_2	A_3B_2	B_3A_2	B_3A_3
T	4	5	6	7	8	8
X	11	11	9	8	8	7
Y	6	9	4	5	4	5

	A_1B_1	A_1B_2	A_1B_3	A_2B_1	A_2B_2	A_2B_3	A_3B_1	A_3B_2	A_3B_3
T	4	6	8	5	6	8	7	7	8
X	11	9	8	11	9	8	10	8	7
Y	6	5	5	6	4	4	6	5	5

Fig. 9. Concept of the 2-D curve and its set. (a) All points in set A^*, and 2-D curve of A^* consisting of the points with smaller sizes in set A^*. (b) Point k_i is selected from sorted array K according to the reference axis, t-axis. If k_i is from set B, then k_i is added to set B^* and update the 2-D curve of B^*. Because k_i is selected according to the t-axis, the points in sets A^* and B^* are also sorted in the increasing order of the t-axis. Once we choose a point k_i, we merge it with other points whose t-axes are no larger than it. For example, when we choose B_2, the points whose t-axes are no larger than B_2 are A_1 and A_2. (c) When choosing B_2 as k_i, we only need to merge B_2 with A_2. Because only A_2, which is on the 2-D curve of A^*, is with the smaller size. Compared to conventional method. (d) We construct the new 3-D surface by sequentially merging B_1 with A_1, A_2 with B_1, B_2 with A_2, A_3 with B_2, B_3 with A_2 and A_3. (e) In the conventional method, total nine permutations are tried and three merged points are added to 3-D curves and then abandoned later.

Features of the 2D curve in set A^* and B^*:

1) The reference axis (t-axis) of all points in set A^* and B^* is less than or equal to the reference axis (t-axis) of the next new point k_i during the merging step.
2) Because the size (regarding to x- or y-axis) of points on the 2-D curves of A^* and B^* is smaller than other points out of curves, the next new point k_i only has to merge with the points on the 2-D curves to get a new smaller point c' (line 11). That is, we prevent unnecessary computation by only merging k_i with points with smaller size.
3) The 2-D curves of A^* and B^* are stored in monotonic form. Therefore, the efficient search method like binary search method can be used to determine whether c' is smaller or not.

Next, merge k_i, which is in A (or B), only with points in the 2-D curve of set B^* (or A^*) into c' (Fig. 9). Add the newly merged point c' to new 3-D surface C_x only when the size

of c' is smaller than x- and y-axis, and any point in C will not be discarded in the following merging step because the t-axis of every point in C is no larger than the newly merged point c'. The possible x-direction merged placement results with smaller size are the stored in 3-D surface C_x. Algorithm 2 shows the complete proposed one-direction 3-D surface merge algorithm, and an example is also in Fig. 9.

Compared to the conventional algorithm, the proposed algorithm offers a speed increase from $O(N^4)$ to $O(N_{cut}^4)$, where $N_{cut} \ll N$, because of the following.

1) In the conventional algorithm, all new points c' may be added into C (Algorithm 1, lines 6 and 7), and in most cases the point c' is discarded later, so called redundant points (Algorithm 1, lines 8 and 9); in contrast, our proposed algorithm does not add any redundant points to the new 3-D surface C (Algorithm 2, lines 10–19) because of the first feature of the 3-D surface. Hence, the proposed algorithm dramatically enhance efficiency without storing and comparing redundant points.
2) The proposed algorithm simplifies the 3-D surfaces A and B into set A^* and B^* and their 2-D curves according to the reference axis. Only points on the 2-D curves of set A^* and B^* are merged (Algorithm 2, line 12) because of the 2nd feature of the 3-D surface, while all the points in A and B are merged and tried in the conventional algorithm.
3) When comparing the module size of a newly merged point c' with points on 3-D surfaces C, only points in C's 2-D curve will be checked because of the 1st feature of the 3-D surface. The time complexity to check the size of point is only $O(log(N))$ by using the binary search method (Algorithm 2, lines 13–18) because the 2-D curves C^* are monotonically sorted, the 3rd feature of 3-D surface.

2) *Step 2: 3-D Surface Merge From Three One-Direction 3-D Surfaces:* After the three one-direction 3-D surfaces with different merging directions, C_x, C_y, and C_z, are created, combine these three surfaces into a complete 3-D merged surface. Because the three surfaces are merged in their individual directions, this step only needs combine all points in these three 3-D surfaces into one surface and discard points with larger sizes.

To reduce the time complexity, use a method similar to the one-direction 3-D surface merge method to reduce redundant the number of checking points: an array K consists of all points in C_x, C_y, and C_z and is then sorted in increasing order according to the t-axis points. Again, a 2-D curve is used to create 3-D surface C_{xyz} during the merging procedure. This produces a new 3-D surface from the two 3-D surfaces of A and B after combining all points in C_x, C_y, and C_z, and removing points with larger sizes.

3) *Step 3: Construct Slicing Tree:* After enumerating all placement permutations from nodes A and B to form a new node C in the slicing tree, we can construct the partition-and-grouped sequential graph into a complete slicing tree. All possible placement results are then stored in the root of the complete slicing tree.

Algorithm 2: Proposed One-direction 3-D Surface Merge Algorithm

Input : two sorted 3-D surfaces A and B

```
1  begin
       /* the concept of reference-axis is introduced
          here to simply 3-D to 2-D curve.          */
       /* for x-direction merge, the reference-axis
          is t-axis.                                */
2      Create a 3-D surface C;
3      Create empty 2-D curve A*, B*, and C*;
       /* the 2-D curves only store the two axes's
          value except the reference-axis           */
4      Create a new array K from all points in A and B in increasing
       order of reference-axis value;
5      for i = 1 to | K | do
6          choose a point k_i from K;
7          this_2D_curve = k_i's 2-D curve;
8          co_2D_curve = the other 2-D curve;
           /* For example, when k_i ∈ A, this_2D_curve
              is A* and co_2D_curve is B*, and vice
              versa                                 */
9          add k_i into this_2D_curve and update this_2D_curve
10         for j = 1 to | co_2D_curve | do
11             choose a point h_j from co_2D_curve;
12             Create a point c' from the pair(k_i, h_j);
               /* k_j is the jth point in co_2D_curve
                  */
13             Binary search C's 2-D Curve
14             if A point c found which (c'_x) ≥ c_x and (c'_y) ≥ c_y then
15                 discard c'
16             else
17                 add c' into C_x;
18 end
```

Output: A merged x-direction 3-D surface C

C. Minimize the Maximum Electrode Actuating Time

The 3D-DDM algorithm produces several 3-D placement results at the root of the slicing tree. The minimized bio-assay completion time of DMFB, which is one of the placement objectives, can be easily obtained by choosing one placement result from the root of the slicing tree. However, the other reliability-oriented objective, to minimize the maximum electrode actuating time, must also be satisfied. Thus, this paper proposes a two-step min-max algorithm that uses dynamic programming to efficiently minimize the maximum actuating time of each electrode in DMFBs. The two-step min-max algorithm is described as follows.

1) *Step 1: Global Adjustment:* The first step of min-max algorithm is the global adjustment of possible vacant positions for each merged block in the slicing tree. All the blocks in the placement result can be divide into two parts recursively from top to bottom because each block in one 3D-DDM placement result is merged to form a slicing tree. Without violating the precedence constraint and keeping the original slicing tree, predict possible locations of every two merged block on vacant space in the global adjustment. This process involves finding all horizontal vacant space S in x-y plane near two merged blocks and dividing the vacant space into two parts according to the x- (or y-) proportion of these two blocks if the two blocks is merged along the x- (or y-) direction in 3D-DDM.

For example, two blocks A and B in Fig. 10 were merged in the x-direction, and all the vacant space S near the two

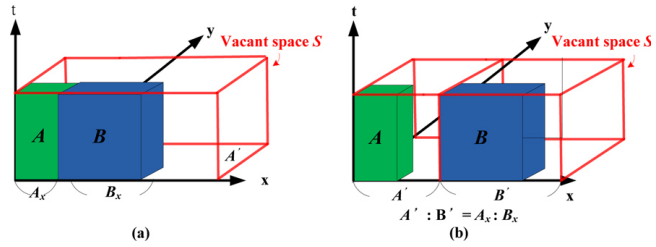


Fig. 10. Example of global adjustment of two x-direction merged blocks A and B in the vacant space S. (a) Original placement result of blocks A and B. (b) Global adjustment result according to the proportion of the x sizes of the two blocks.

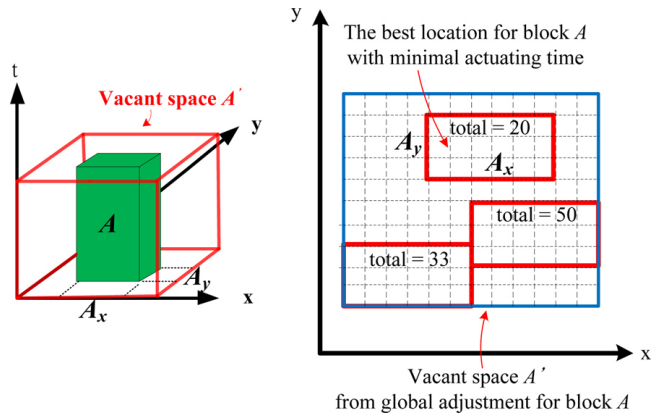


Fig. 11. Detail adjustment for block A in the vacant space A', and choose the square with minimal total actuating time as the precise location of block A.

blocks is divided into two spaces A' and B' according to the proportion of x-axis of blocks A and B. Hence, two blocks can be arranged A and B in any position of A' and B'.

2) *Step 2: Detail Adjustment:* Based on the global adjustment result, the min-max algorithm makes the detail adjustment by finding the minimal actuating rectangle in one vacant space obtained from global adjustment, and then locate the module in its vacant space. Because we only adjust one module's position in a known 3-D placement, the actuating time of each electrode can be summed up only without the actuating time of the module which is currently adjusting. For example, total actuating times in the rectangle of $A_x \times A_y$ are calculated first using dynamic programming. A can locate at the minimal actuating time rectangle in the vacant A', as shown in Fig 11.

Hence, using the global and detail adjustment in the min-max algorithm on every module combinations, the maximum electrode actuating time of each 3-D placement result in the root of the slicing tree are minimized efficiently.

D. Avoid Excessive Actuated Electrode

In addition to minimizing the maximum electrode actuating time, the proposed method prevents the charge problems in electrodes and enhances the reliability of DMFBs. As mentioned in the preliminaries, this paper adopts the reliability constraint function $f(r) = r/11 + 1$ based on the physical characteristics of the electrode [8]. The min-max algorithm avoids most EA electrodes because EA electrodes and maximum actuating electrodes usually occurs in the same space.

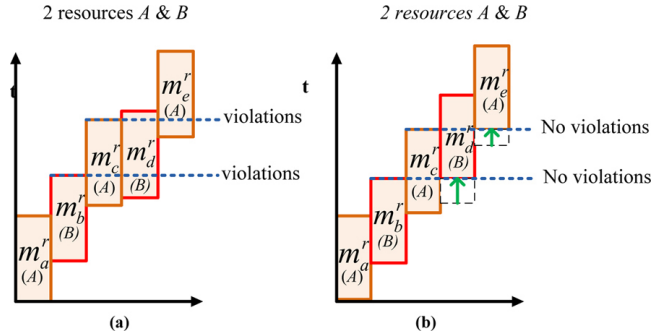


Fig. 12. (a) Violations occur on module m_d^r and m_e^r because m_d^r requests the second resource, while the m_b^r is executing and m_e^r requests the first resource while the m_c^r is executing. (b) No violation exists after resource constraint check.

Next, check the reliability constraint by calculating the continuous actuating time of each electrode in every DMFB placement result. According to the function $f(r)$, supposing a module m_i with duration of t_i seconds exists and other modules m_j are activated for t_j seconds continuously, for every electrode at position (x, y) in module m_i, the constraint verifications can be divided into two conditions.

1) No violation of reliability constraint: This occurs when $t_i + t_j \leq 10$ in every position (x, y).
2) Eliminate violation of reliability constraint: If $a + b > 10$ for any position (x, y) in modules m_i and m_j, the reliability constraint is violated because an electrode is actuated continuously for more than 11 seconds, and tje charge problem occurs.

Note that the violation cannot be eliminated by deactivating electrodes during one certain module m_i because unpredicted droplet reaction on DMFBs may be occurred. The violations of reliability constraint can only be eliminated by deactivating electrodes between two modules.

Therefore, when any reliability violation is found, pull up the module m_i for $f(r)$ scales along the t-axis to deactivate the electrodes for the required time period according to the reliability constraint function $f(r)$. This approach also considers and prevents any violations of other constraints while satisfying the reliability constraints.

E. Consider DMFB's Constraints in 3D-DDM

The 3-D placement results of DMFBs must satisfy several constraints, as mentioned in Section III. The time complexity of eliminating the constraint violations in conventional SA-based algorithms is unacceptable because these constraints must be verified and satisfied in each SA iteration. Conversely, the placement step of 3D-DDM avoisd the violations of DMFB's constraints and obtain several refined and optimized 3-D placement from several results at the root of slicing tree.

1) *Precedence Constraint:* The reliability-oriented 3-D placement algorithm avoid the precedence violation rather than removing violations after they happen greatly reducing the complexity of the proposed algorithm. This paper considers the precedence constraint efficiently during 3D-DDM in two stages.

1) Reduce the precedence constraint in the grouping and partition stage: During the preprocessing stage, divide the operation sequential graph into small sizes of groups. According to the precedence relationships, operations are grouped together and the precedent relationships between each group are reduced. The 3D-DDM stage greatly reduces the complexity of checking precedence constraint.
2) Avoid precedence violation in the 3D-DDM stage: Check every precedence constraint and avoid violations when constructing a one-direction 3-D surface C from surfaces A and B. In this case, the newly merged point $c\prime$ is discarded before adding $c\prime$ to the new 3-D surface C when precedence violation is found in $c\prime$.

Most precedent constraints are avoided when constructing a nonleaf node from leaf node of slicing tree because most precedence constraints are exists between leaves and few relationships between other nodes. Hence, the complexity of precedence constraint is reduced significantly with above considerations, and no precedence violation is occurred during and after placement in 3D-DDM step.

2) *Resource Constraint and Placement:* After 3D-DDM and min-max methods, a set of placement results is at the root node of a slicing tree, which contains several possible placements of all reconfigurable operations. Next, insert all nonreconfigurable operations into these placements for every point of the top surface-set and ensure that the final placement results satisfy the resource and detection constraints. This stage can be separated into two steps: detection operation placement and source operation placement.

Before inserting resources at verifying the resource constraint and placement, some assumptions are introduced according to the function of nonreconfigurable operations: 1) This detection operation can only take one result from the other operations because it uses only one photodiode. 2) Source operation can only give its result to one operation since a dispensing port generates only one droplet at a time. 3) Detection operation requires a 1×1 area because it uses one photodiode, which is fabricated on-chip. 4) Source operation requires 0×0 area because the source is only a dispensing port outside the biochip.

All of these assumptions are based on the property and function of resource operations in DMFBs, and are consistent with benchmarks used in [11], [15] and the current paper. Resource binding is not required for these operations because the dimensions of detection and source operations are fixed (i.e., there is only one choice in module library).

1) Resource constraint: To check the resource constraint, first build sorted lists M^r from operations that require the resource type γ in increasing order of execution start time. It is possible to check for any violation from the sorted list M^r based on the property that all operations in M^r require the same resource type γ. Assume the total number n_r of resources γ can be used by n_r operations at the same time, for the ith point m_i^r in M^r using resource γ, the resource constraint will not be violated if $t(m_i^r) > t(m_j^r) + d_j$ and $t(m_i^r) + d_i < t(m_k^r)$ when m_j^r and m_k^r are the nth precedent and successive modules in sorted list M^r, respectively.
Check the resource constraint from the first to the last node in every sorted list M^r and shift up modules (i.e., postpone the start time of the operations that violate the resource constraint). If the shifted modules overlap with other modules, the overlapped modules also shift up until no overlapped module exist. Example in which $n_r = 2$ shows how the violation of resource constraint occurs, and can be solved. Assume that the five modules using the resource γ, are m_a^r, m_b^r, m_c^r, m_d^r, and m_e^r, as Fig. 12(a) shows. The resource constraint violated modules, m_c^r, m_d^r, and m_e^r, are then solved at the step of checking resource constraint (Fig. 12(b)).
2) Resources placement: This step inserts resource operations into placement results from 3D-DDM to satisfy the resource constraints. First, place detection operations because the detection operations require a 1×1 area and usually require a longer operation time than other resources. Then insert other resource operations. Because the area of resources is only up to 1×1, we can insert resource operations easily into enumerating results without violating the resource constraint and, preserving the topological form of the placement results. If no space exists for resources, the placement result is abandoned.

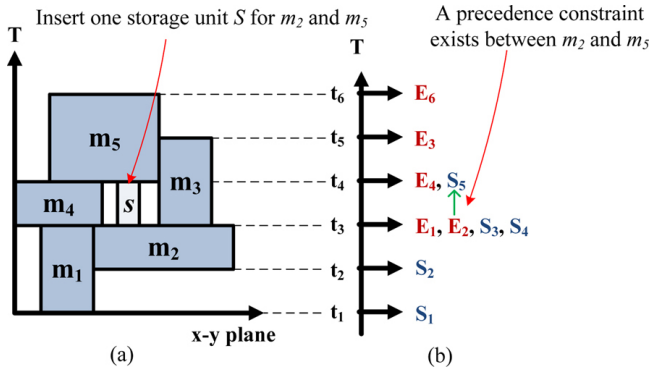


Fig. 13. (a) Example of an operation sequential graph and its placement result. (b) Corresponding project list of the placement that stores and sorts the start and finish times of each placement, where S_i means module i starts and E_i means module i finishes. One storage unit is inserted between E_2 and S_5 to satisfy the storage constraint.

3) *Storage Constraint and Storage Unit Placement:* The last step of DMFB placement is to satisfy the storage constraint. The storage constraint is violated when an operation m_j in the operation sequencing graph does not start to execute immediately when its precedent operation m_i is finished (i.e., the start time t_j of operation m_j is larger than the finished time t_i of operation m_i). Therefore, a storage unit, an extra space, is required to store the product of operation m_i when m_i is finished at t_i, and the product is moved to the operation m_j at m_j's start time t_j.

The 3D-DDM algorithm considers the storage constraint while constructing slicing tree. If the storage constraint is not checked until the result of 3D-DDM, many placement results

TABLE I

COMPARISONS OF PLACEMENT RESULTS BETWEEN [15], AND OURS. OUR RESULTS ARE MUCH BETTER THAN [15], AND THE IMPROVEMENTS OF ASSAY COMPLETION TIME, MAXIMUM ACTUATING TIME, AND CPU TIME, ARE SHOWN IN THE AVERAGE ROWS

Testcase	Design Spec.		T-tree [15]		Our Proposed Algorithm w/o Reliability Constraint		Our Proposed Algorithm w/ Reliability Constraint	
	#Op.	DMFB Size	Assay Completion Time (s)	CPU Time (s)	Assay Completion Time (s)	CPU Time (s)	Assay Completion Time (s)	CPU Time (s)
Protein	103	10 × 10	241	157.42	183	30.54	183	31.04
		10 × 10	231	141.29	183	31.01	183	31.25
		11 × 11	221	143.11	183	30.88	183	31.12
		9 × 9	240	138.94	183	30.11	183	30.66
Average	–	–	233.3	145.19	183 (0.78X)	30.64 (0.20X)	183 (0.78X)	31.02 (0.21X)
In-Vitro1	64	9 × 9	67	19.85	53	3.37	53	3.52
		8 × 8	98	23.21	53	1.92	53	2.07
		7 × 7	96	17.91	53	1.21	56	1.38
Average	–	–	87.0	30.32	53.0 (0.61X)	2.17 (0.07X)	54.0 (0.62X)	2.32 (0.08X)
In-Vitro2	48	8 × 8	74	13.01	41	0.28	41	0.33
		7 × 7	62	11.88	41	0.20	42	0.24
		6 × 6	73	22.67	41	0.14	47	0.17
Average	–	–	69.7	15.85	41.0 (0.59X)	0.21 (0.01X)	43.3 (0.62X)	0.25 (0.01X)
In-Vitro3	36	7 × 7	60	8.92	44	4.03	44	4.22
		6 × 6	61	9.56	44	2.26	45	2.60
		5 × 5	64	5.28	44	1.81	46	2.03
Average	–	–	61.7	7.92	44 (0.71X)	2.70 (0.34X)	45 (0.73X)	2.95 (0.37X)

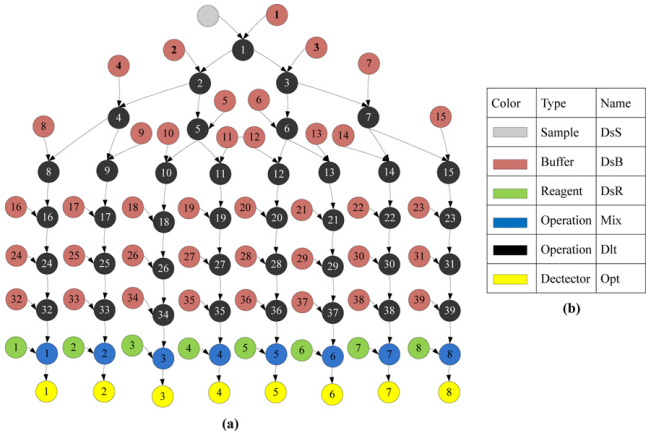


Fig. 14. Real bio-assay case, protein assay, with 103 operations. (a) Sequential graph of the bio-assay. (b) Corresponding operation in the sequential graph.

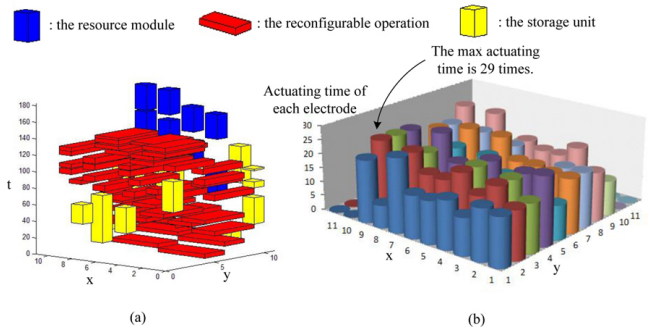


Fig. 15. Protein assay with 103 operations under 11×11 design specification. (a) 3-D placement result of 103 operations and the assay completion time is 183 with the reliability constraint. (b) Histogram shows the actuating times in each electrode on DMFB, and the maximum actuating time is only 29. Because the last two columns and rows in the DMFB are used for resource and storage placement, the electrodes in these places are not actuated.

from the root of the slicing tree fail to satisfy the storage constraint, and the complexity of post-adjustment is increased or some placement results are discarded. Therefore, we check the storage constraint when obtaining 3-D placement C from merging 3-D curves A and B, and make sure enough vacant if the storage unit is required between A and B.

Next, insert the storage unit to the placements from the result of previous stage. Fig. 13 shows an example of this. Because 3D-DDM preserves enough vacant space in advance, we can insert a storage unit easily in the vacant space to satisfy the storage constraint.

V. EXPERIMENTAL RESULTS

This paper implements our proposed reliability-oriented placement algorithm using the 3D-DDM technique in C++ language on a 2-GHz 64-bit Linux machine with 16 GB of memory. Two commercial bio-assays, which are colorimetric protein assay [9] and multiplexed in-vitro diagnostics [5], are used for experimental evaluation. The two biochips are typically used for point-of-care testing, and involve typical microfluidic droplet protocols that are used for many healthcare-related assays. To demonstrate the effectiveness of the proposed reliability-oriented placement algorithm for optimal assay completion time and minimal maximum electrode actuating time, this paper compares the current placement results with the up-to-date placement algorithm without defects [15], which was also implemented on our Linux machine.

Table I shows the placement results obtained from our proposed algorithm. Column 1 lists four kinds of testcases: protein, in-vitro1, in-vitro2, and in-vitro3. Column 2 shows the number of operations for each testcase, and Column 3 shows the different design specifications for each testcase.

TABLE II

ANALYSIS OF PLACEMENT RESULTS OF TABLE I. THE RELIABILITY CONSTRAINTS ARE ALL SATISFIED IN OUR PLACEMENTS RESULTS WITH ALMOST THE SAME ASSAY COMPLETION TIMES WHILE VIOLATIONS OCCURRED IN [15]

Testcase	Design Spec.		T-tree [15]		Our Proposed Algorithm w/o Reliability Constraint		Our Proposed Algorithm w/ Reliability Constraint	
	#Op.	DMFB Size	Maximum Actuating Time (s)	# of EA Electrode	Maximum Actuating Time (s)	# of EA Electrode	Maximum Actuating Time (s)	# of EA Electrode
Protein	103	10×10	115	52	108	20	36	0
		10×10	178	57	108	20	36	0
		11×11	141	59	108	20	29	0
		9×9	145	63	108	20	62	0
Average	–	–	144.8	57.8 (violated)	108.0 (0.75X)	20 (violated)	41.0 (0.28X)	0 (meet)
In-Vitro1	64	9×9	58	18	14	0	14	0
		8×8	64	20	14	0	14	0
		7×7	68	20	18	0	16	0
Average	–	–	63.3	19.3 (violated)	15.3 (0.24X)	0 (meet)	14.7 (0.23X)	0 (meet)
In-Vitro2	48	8×8	46	4	10	0	10	0
		7×7	47	5	11	0	11	0
		6×6	51	7	23	4	21	0
Average	–	–	48.0	5.3 (violated)	14.7 (0.31X)	1.3 (violated)	14.0 (0.29X)	0 (meet)
In-Vitro3	36	7×7	45	11	8	0	8	0
		6×6	48	9	12	0	12	0
		5×5	53	7	24	0	24	0
Average	–	–	48.7	9 (violated)	14.7 (0.30X)	0 (meet)	14.7 (0.30X)	0 (meet)

In order to prove the efficiency and effectiveness of our proposed algorithm, the assay completion time and CPU of our placement results without and with reliability considerations (i.e., avoiding EA electrodes and minimizing the maximum electrode actuating time). Although the results from [15] in Columns 4–5 meet the designed specification dimensions (i.e., *Width* $\times$ *Height*), the placement results in Columns 6–7 without reliability consideration can achieve an assay completion time average that is 27% shorter than that in [15], fulfilling the major objective of DMFB placement. Even under the reliability considerations, the reliability-oriented placement results in Columns 8 and 9 also achieve a 26% shorter assay completion time. By considering reliability during 3D-DDM and obtaining multiple 3-D placement results, the proposed method achieved can get almost the same assay completion times between the conditions without and with reliability constraint. Also, our reliability-oriented algorithm is much more efficient in terms of CPU running time, which is only 21% CPU time of [15] for the large protein bio-assay.

The reliability issue is the most critical requirement as any fluidic error or physical defect directly affects medical diagnosis. An analysis of the reliability-oriented placement results shows that this method avoids EA electrodes and minimizes the maximum electrode actuating time. Because the 3D-DDM algorithm produces more than one placement result, the best reliability-oriented placement result can be chosen from several results with shorter assay completion time. The experimental results shown in Table II also prove that the proposed algorithm can eliminate all EA electrodes and minimize the maximum actuating times, compared to [15] and our own algorithm without reliability consideration. Fig. 15 shows the reliability-oriented placement result for the protein assay [9] obtained from the proposed placement algorithm with 11×11 design specification. Fig. 14(a) shows the sequential graph of protein assay, including reconfigurable operation, resource module, and storage unit. The completion time of the protein assay is only 183 s. Fig. 15(a) shows the 3-D histogram of actuating time for each electrode, and the maximum actuating time among all 11×11 electrodes is reduced to only 29 without any EA electrodes. Hence, we can obtain reliability-oriented 3-D placement results for DMFBs with a much shorter assay completion time and CPU runtime using the proposed algorithm.

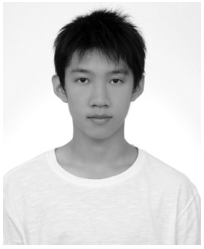
VI. CONCLUSION

This paper represented a novel reliability-oriented 3-D placement algorithm to deal with the involved reliability problem in DMFBs. This paper proposed a new 3-D placement algorithm using the 3D-DDM technique. This paper identified the causes of reliability degradation and introduced a new and practical formulation of reliability-oriented 3-D placement problem. By incorporating the properties that were favorable for reliability enhancement in the placement algorithm, the reliability-oriented 3-D placement problem can be effectively solved with the optimal bio-assay completion time, while minimizing the electrode actuating time condition and avoiding excessive actuated electrodes. Several commercial DMFBs of point-of-care testing, including protein and in-vitro bioassays, were used to evaluate the effectiveness and efficiency of the proposed reliability-oriented placement algorithm in preventing charge problem, and in minimizing the maximum actuating times of all electrodes.

REFERENCES

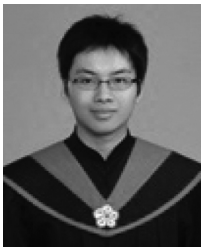
[1] K. Chakrabarty, "Toward fault-tolerant digital microfluidic lab-on-chip: Defects, fault modeling, testing, and reconfiguration," in *Proc. IEEE ICBCS*, Nov. 2008, pp. 329–332.

[2] A. I. Drygiannakis, A. G. Papathanasiou, and A. G. Boudouvis, "On the connection between dielectric breakdown strength, trapping of charge, and contact angle saturation in electrowetting," *ACS J. Langmuir*, vol. 1, no. 25, pp. 147–152, 2009.
[3] R. B. Fair, "Digital microfluidics: Is a true lab-on-a-chip possible?" *Microfluidics Nanofluidics*, vol. 3, no. 3, pp. 245–281, 2007.
[4] T.-Y. Ho, J. Zeng, and K. Chakrabarty, "Digital microfluidic biochips: A vision for functional diversity and more than Moore," in *Proc. IEEE/ACM ICCAD*, Nov. 2010, pp. 578–585.
[5] T.-W. Huang, T.-Y. Ho, and K. Chakrabarty, "Reliability-oriented broadcast electrode-addressing for pin-constrained digital microfluidic biochips," in *Proc. IEEE/ACM ICCAD*, Nov. 2011, pp. 223–228.
[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.
[7] R. H. J. M. Otten, "Automatic floorplan design," in *Proc. IEEE/ACM DAC*, Jun. 1982, pp. 261–267.
[8] J. K. Park, S. J. Lee, and K. H. Kang, "Fast and reliable droplet transport on single-plate electrowetting on dielectrics using nonfloating switching method," *J. Biomicrfluidics*, vol. 2, no. 4, pp. 2–9, 2010.
[9] V. Srinivasan, V. Pamula, P. Paik, and R. Fair. "Protein stamping for MALDI mass spectrometry using an electrowetting-based microfluidic platform," in *Proc. SPIE*, 2004, pp. 26–32.
[10] F. Su and K. Chakrabarty, "Architectural-level synthesis of digital microfluidics-based biochips," in *Proc. IEEE/ACM ICCAD*, Nov. 2004, pp. 223–228.
[11] F. Su and K. Chakrabarty. "Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips," in *Proc. IEEE/ACM DAC*, Jun. 2005, pp. 825–830.
[12] F. Su, K. Chakrabarty, and R. B. Fair, "Microfluidics based biochips: Technology issues, implementation platforms, and design-automation challenges," in *Proc. IEEE TCAD*, 2006, pp. 211–223.
[13] H. J. J. Verheijen and M. W. J. Prins, "Reversible electrowetting and trapping of charge: Model and experiments," *ACS J. Langmuir*, vol. 20, no. 15, pp. 6616–6620, 1999.
[14] J. Z. Yan and C. Chu, "DeFer: Deferred decision making enabled fixed-outline floorplanner," in *Proc. IEEE/ACM DAC*, Jun. 2008, pp. 161–166.
[15] P. Yuh, C. Yang, and Y. Chang, "Placement of defect-tolerant digital microfluidic biochips using the T-tree formulation," *ACM JETC*, vol. 3, no. 3, article 13, 2007.



Ying-Han Chen received the B.S. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2012.

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

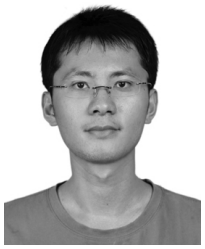


Chung-Lun Hsu received the B.S. and M.S. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2007 and 2009, respectively. He is currently pursuing the Ph.D. degree at the University of California, San Diego, CA, USA.

His current research interests include digital microfluidic biochips and analog circuitry, especially related issues in sigma-delta modulator.

Li-Chen Tsai received the B.S. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2012.

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



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

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

Mr. Huang was a recipient of the First Prize from the ACM SIGDA Student Research Competition (SRC), awarded at ACM/IEEE DAC in 2011. He was also the Second Prize Winner in the ACM Student Research Competition (SRC) Grand Final in 2011. He was a recipient of the Award of Microelectronics and Computer Development Fellowships.

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

Since 2007, he has been with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, where he is currently an Associate Professor. From 2003 to 2004, and in 2005 and 2008, he was a Visiting Scholar at the University of California, Santa Barbara, CA, USA, Waseda University, Tokyo, Japan, Synopsys, Inc., Mountain View, CA, USA, respectively. He has published several papers in top journals and conferences, such as IEEE TCAD, ACM TODAES, ACM/IEEE DAC, IEEE/ACM ICCAD, and ACM ISPD. His current research interests include design automation for digital microfluidic biochips and nanometer integrated circuits.

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