IEEE Access®

Multidisciplinary : Rapid Review : Open Access Journal

Unified Contamination-Aware Routing Method Considering Realistic Washing Capacity Constraint in Digital Microfluidic Biochips

ZHIPENG HUANG[1], XIQIONG BAI[1], TINGSHEN LAN[1], XINGQUAN LI[2], AND GENG LIN[3]

[1]Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou 350000, China (e-mails: m185410003@fzu.edu.cn, baixiqiong@163.com, lantingshen@163.com)
[2]School of Mathematics and Statistics, Minnan Normal University, Zhangzhou 363000, China (e-mail: xqli@mnnu.edu.cn)
[3]College of Mathematics and Data Science, Minjiang University, Fuzhou 350000, China (e-mail: lingeng413@163.com)

Corresponding author: Geng Lin (e-mail: lingeng413@163.com).

ABSTRACT To fully utilize the dynamic reconfigurability of digital microfluidic biochips, most of electrodes would be shared by different droplets. Thus, contaminations caused by liquid residues among droplets are inevitable which lead to lethal errors in bioassays. To remove the contaminations, washing operations are introduced as an essential step to ensure the correctness of bioassay. However, existing works have oversimplified assumptions on the washing droplet's behavior and constraints which cannot clean all contaminations with erroneous outcomes. Moreover, straightforward integration of washing operations with droplet routing may increase the execution time of a bioassay which is not feasible for timing-critical bioassay. To effectively remove contaminations and minimize the execution time of a bioassay, this paper proposes a unified contamination-aware routing method, which addresses the above issues simultaneously. Firstly, we present a top-down scheme to generate candidates of routing paths, then construct a shortest-path model to select desirable routing solution for all subproblems. With a decision diagram of droplets, we further propose an integer linear programming (ILP) formulation to compact the execution time. Finally, contamination removal by washing droplets with realistic washing capacity is considered for all subproblems. Tested on real-life benchmarks, our proposed method can significantly reduce 72% contamination spots and save 11% execution time.

INDEX TERMS Digital microfluidic biochip, dynamic reconfigurability, washing operations, contamination aware routing, integer linear programming.

I. INTRODUCTION

Microfluidic laboratories-on-chip (LoCs) are replacing the conventional biochemical analyzers and are able to integrate the necessary functions for biochemical analysis on chip [1]. In the automation and miniaturization applications of biochemical laboratories, the digital microfluidic biochips (DMFBs) are being widely used as a revolutionary technique to realize the lab-on-a-chip (LoC) [2], [3]. Compared with the traditional biochemical analyzer, microfluidic biochips have various advantages, such as lower cost, smaller size, larger throughput, increased automation, higher sensitivity, and flexibility [4], [5]. Therefore, DMFBs are widely applied in real-time DNA sequencing, the development of new drugs,

and point-of-care clinical diagnostics, etc [6].

Based on the electrowetting technology [3], DMFBs control the wetting behavior of polarizable or conductive droplets through an electric field, thereby controlling the movement of the droplets. As shown in Fig. 1(a), a DMFB of the sandwich structure consists of a top plate, a bottom plate, and a flow layer between these two plates [7], [8]. The flow layer constituting DMFB mainly consists of three parts: a 2-D array of controllable electrodes (cells), the peripheral devices (optical detector, heater, etc.), and the dispensing reservoirs/ports [9], [10]. By applying a voltage to activate adjacent electrode, the discrete reagent or sample droplets between the bottom and top plates can be moved horizontally
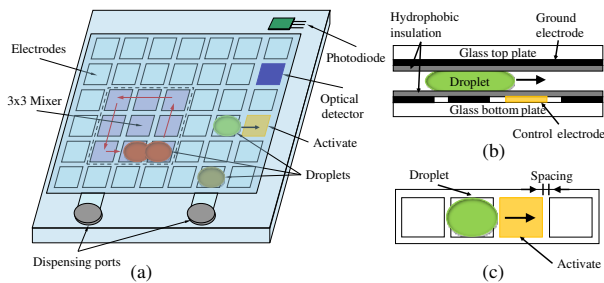
IEEE Access

Unified Contamination-Aware Routing Method Considering Realistic Washing Capacity Constraint in Digital Microfluidic Biochips

FIGURE 1: A DMFB. (a) Schematic of a DMFB. (b) Side view of the 2-D electrode array. (c) Top view of the 2-D electrode array.

FIGURE 2: Physical-level synthesis. (a) Module placement. (b) 3-D module placement. (c) Droplet routing.

or vertically to its adjacent activated electrode [11], [12]. As shown in Fig. 1(b) and Fig. 1(c), when the electrode on the right side of the green droplet is activated, the droplet will move to the right electrode. The principle of controlling droplet movement by activating the corresponding control electrode is called electrowetting-on-dielectric (EWOD) [13], [14]. Therefore, according to the principle of EWOD, we can activate the electrodes around the dispensing ports. And then the discrete droplets of microliter volumes can be easily separated from the dispensing ports and fed into the chip.

In the platform of DMFB, a rectangular structure covering several adjacent electrodes is called a module. According to the EWOD principle of the electrode, if we apply a timing-vary voltage sequence to each electrode inside the module, the droplets inside the module will move repeatedly inside the module. As shown in Fig. 1(a), we can apply a clockwise or counterclockwise voltage inside the module (3×3 Mixer) to activate the electrodes inside the module, so that the red droplets inside this module will be mixed. Similarly, we can use a virtual module to implement a series of basic operations in the bioassay, such as splitting, diluting, and detection [15]. When the operation is completed, the module used for the operation will be canceled, and the electrodes used by this module can be recombined into other modules for other operations. Therefore, the module for performing basic operations is dynamically reconfigurable [12], [16].

In the traditional DMFB design methodology, it mainly includes two phases: architecture-level synthesis and physical-level synthesis [17], [18]. Architecture-level synthesis contains resource binding and operation scheduling [19]. During this phase, the module specifications corresponding to each basic operation and the times which each operation starts to be performed are determined [20]. After that, the physical-level synthesis which contains module placement and droplet routing [17], [21]–[23] will determine the specific location of each basic operation on the electrode array and the specific path of droplet transport. For example, as shown in Fig. 2(a), the specific locations of each module in the electrode array will be specified based on the results of the operational scheduling. The problem of module placement for DMFB is a 3-D packing problem due to the virtual and reconfigurable features of the module [6] which is shown in Fig. 2(b).
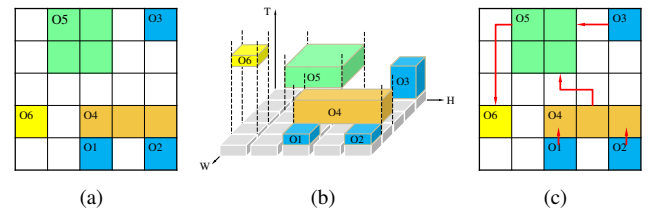
At different times, there may be different module placement results on the electrode array. Furthermore, we can divide a complete bioassay into a series of successive subproblems to perform according to the results of the module placement. Within each subproblem, Pin is the input (or output) port of the module, indicates an electrode around the module. For the following operation to be performed sequentially according to the results of the operational scheduling, the droplets are transported in advance to the pin of the module. The process of transporting droplets between the pin of the module and the port of chip is called droplet routing in Fig. 2(c), and it is also a crucial step for ensuring the automatic process of DMFB. As the scale of DMFB becomes larger and the bioassay becomes more complicated, automatic droplet path calculation becomes more and more important, which determines the final routing paths for droplets, thereby determining the correctness and performance in implementing the assays [24].

A. PREVIOUS WORKS

As a functional (reagent or sample) droplet moves forward along its path, it will leave liquid residue on the passed cells (electrodes). Two kinds of cross-contamination spots including intra-contamination spot and inter-contamination spot will occur when different functional paths have an intersection. Based on the results of the module placement, the droplet routing problem is divided into a series of successive subproblems. If the two functional paths are from the same subproblem, then these cross-contamination spots are called intra-contamination spots Conversely, if they are from two adjacent subproblems, then these spots are called inter-contamination spots. In DMFB, cross-contamination of droplets with different biomolecules is a major issue, which causes the inevitable erroneous reaction [25]. However, there are usually no available disjoint functional paths since the electrodes in the array are limited. Thus, to effectively avoid cross-contamination, we have to introduce washing droplets to wash the contaminated cells [11], [26]. To reduce the effect of liquid residue on final bioassay results, several routing and washing algorithms have been proposed to minimize cross-contamination [24], [27], [28].

Most of the previous works only washed contamination spots within a separated subproblem, i.e., intra-contamination spots. Actually, contamination spots are mainly generated between inter-contamination and intra-
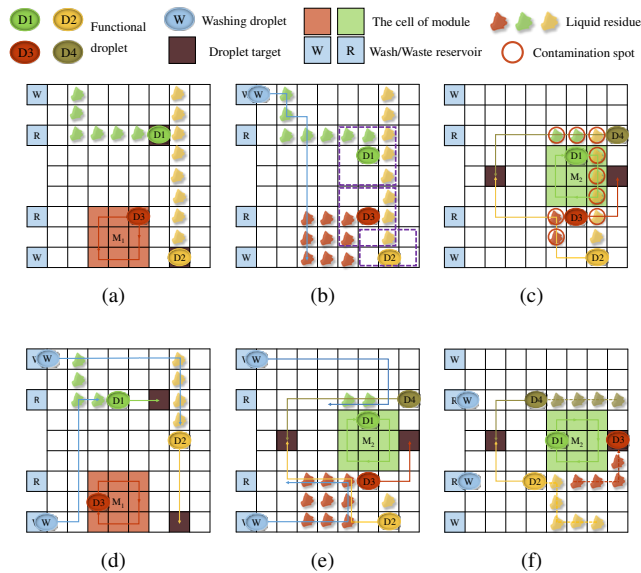
FIGURE 3: (a) In subproblem SP_{k-1} , $D1$ and $D2$ reach their target cells, respectively, and $D3$ is performing a corresponding operation inside $M1$. (b) After SP_{k-1} has been routed, many residues are left on the electrode array. Then W is introduced to wash these residues. To satisfy the fluidic constraint, a cell interval must be maintained between $D1$ ($D2$ or $D3$) and W. Thus, many residues cannot be washed. (c) Subproblem SP_k begins routing, many contamination spots have formed in SP_k since the residues left by SP_{k-1} are not completely washed. (d) If simultaneous subproblems are considering, some generated residues by the droplets in SP_{k-1} will be assigned to wash in SP_{k-1}. (e) Other generated residues by the droplets in SP_{k-1} will be assigned to wash in SP_k. (f) In SP_k, the residues left by SP_{k-1} is completely washed. Meanwhile, since both the washing droplets and the functional droplets are routed synchronously during the washing process, some functional droplets have been routed to toward their target cell, which saves the execution time of a bioassay.

contamination spots. In order to consider the inter-contamination spots, Huang et al. presented a contamination-aware droplet routing algorithm [27]. However, they ignored the cross-contaminations between reconfigurable modules and droplet routing paths, and many contamination spots have not been washed. In addition, they did not consider the crucial capacity constraint of washing droplets. In fact, the washing capacity of a washing droplet will decrease when residues are washed away from the electrodes [24]. Hence, the realistic washing capacity constraint need to be considered.

Further, Yao et al. presented an integrated functional and washing droplet routing flow, which considers the washing capacity constraint [24]. However, they did not consider the washing of liquid residue between successive subproblems. In [28], to consider both of the washing capacity and the liquid residue between successive subproblems, extra washing operations were added to clean the residue between successive subproblems. The extra washing operations may increase the execution time of a bioassay. Moreover, [28] ignored the routing conflicts between functional and washing

droplets. For example, in Fig. 3(b), to satisfy the fluidic constraint, the residues around the functional droplets can not be washed. As a result, many contamination spots are formed as Fig. 3(c). However, if we consider all subproblems simultaneously, then these residues will be assigned to its adjacent subproblems as Fig. 3(d) and (e). As a result, these contamination spots can be washed completely. In addition, the washing and functional droplets are routed at the same time in Fig. 3(f), thus the execution time can be saved.

B. OUR WORKS

To effectively remove contaminations and minimize the execution time of a bioassay while all the fluidic, timing, contamination, and washing capacity constraints are satisfied, it is desirable to consider all subproblems simultaneously. In this paper, we propose a unified contamination-aware routing method. The major contributions of this paper are summarized as follows.

- Unlike previous works, we present an effective routing flow to practically resolve both intra-contaminations and inter-contaminations with washing capacity constraint by simultaneously considering all subproblems in DMFBs.
- A top-down scheme is proposed to generate candidate routing paths and a new shortest-path formulation is proposed to select desirable routing paths for all the subproblems.
- A new scheduling model is constructed, and an ILP-based scheduling algorithm is presented to compact the time of scheduling.
- A reasonable contamination spots assignment principle is proposed to assign the contamination spots to their adjacent subproblems, then all washing paths are generated based on the A* routing algorithm [29] with considering the washing capacity constraint.
- Experimental results validate the effectiveness of our proposed method. Compared with conventional washing method, our unified routing and washing method can significantly reduce 72% contamination spots and save 11% execution time. Particularly, our proposed method achieves 48% shorter total execution time than the existing work [24].

The remainder of this paper is organized as follows. Section II describes the problem statement and the proposed algorithm flow. Section III gives the simultaneous routing scheme of subproblems. Section IV introduces the ILP-based scheduling approach. Section V presents our washing algorithm. Section VI shows the experimental results. Finally, conclusions are made in Section VII.

II. PROBLEM STATEMENT AND THE PROPOSED ALGORITHM FLOW

A. PROBLEM STATEMENT

In the phase of droplet routing, a droplet form a net from its source cell to its target cell. The droplets that need to be

IEEE *Access*

Unified Contamination-Aware Routing Method Considering Realistic Washing Capacity Constraint in Digital Microfluidic Biochips

routed inside each subproblem and the start position (source cell) and end position (target cell) of these droplets can be obtained from the result of the module placement, and then the electrodes inside modules will be treated as obstacles to avoid during the droplet routing process. The objective of droplet routing is to find a path from its source cell to its target cell without passing obstacles for each net. There are four constraints in the unified contamination-aware routing problem: fluidic constraint, timing constraint, contamination constraint, and washing capacity constraint.

1) Fluidic Constraint

During droplet transportation, the fluidic constraint can ensure accidental mixing is forbidden between two droplets of different paths, which includes static constraint and dynamic constraint. Let (x_i^t, y_i^t) and (x_j^t, y_j^t) represent the locations of droplets d_i and d_j in the array at time t, respectively. Then, the fluidic constraint can be described as follows:

- Static constraint: $|x_i^t - x_j^t| \geq 2$ or $|y_i^t - y_j^t| \geq 2$,
- Dynamic constraint: $|x_i^{t+1} - x_j^t| \geq 2$ or $|y_i^{t+1} - y_j^t| \geq 2$ or $|x_i^t - x_j^{t+1}| \geq 2$ or $|y_i^t - y_j^{t+1}| \geq 2$.

2) Timing Constraint

Given maximum execution time T_{max}, the timing constraint can ensure every droplet from its source to target should be not more than T_{max}. Timing constraint is mainly used for timing critical bioassays.

3) Contamination Constraint

The contamination constraint can ensure contaminations left by the previous droplet should be washed before a new droplet reaches this spot.

4) Washing Capacity Constraint

The washing capacity constraint means that the capacity of each washing droplet is limited. That is, when a washing droplet reaches its washing threshold value W_{max}, it cannot be used to further wash any residue [24].

The unified contamination-aware routing while considering realistic washing capacity constraint problem of a DMFB can be formulated as follows.

Input: A DMFB array, a series of successive subproblems, a set of nets to be connected, a set of washing droplets, the locations of reservoirs, a list of locations of modules, the timing constraint, and the washing capacity constraint.

Objective: All functional droplets are routed from their sources to their targets without violating any constraints, such that the number of contamination spots, the number of used cells[1], and the execution time of a bioassay are minimized.

Constraints: All the fluidic, timing, contamination, and washing capacity constraints are satisfied.
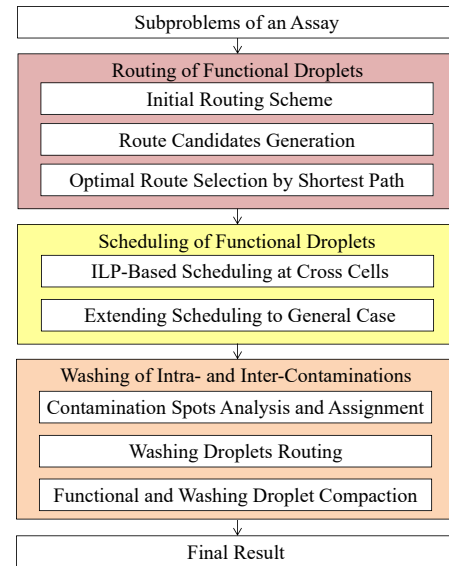

FIGURE 4: Our proposed unified contamination-aware routing method.

B. THE PROPOSED ALGORITHM FLOW

Fig. 4 shows the overall flow of the proposed approach, which consists of three major stages: routing of functional droplets, scheduling of functional droplets, and washing of intra- and inter-contaminations. In the routing of the functional droplets stage, we first generate routing candidates for each subproblem. Then, we propose an algorithm based on the shortest path to determine a specific path for each net, while minimizing the number of contamination spots and the length of the path. In the scheduling of functional droplets stage, we construct a new scheduling model, and present an ILP-based approach to compact the total execution time. Finally, at the stage of washing of intra- and inter-contaminations, we first assign the contamination spots in reasonable assignments of adjacent subproblems, and then adopt an A*-based washing algorithm to wash these contamination spots, which considers the washing capacity constraint in the meantime. Finally, a compact algorithm based on greedy strategies is used to simultaneously compact all droplets, while minimizing the execution time of bioassay. The details of each part are elaborated in the following sections.

III. ROUTING OF FUNCTIONAL DROPLETS

In this section, we first propose a top-down routing scheme to generate several route candidates for each subproblem. Then, to achieve a desirable route design for every subproblem, we simultaneously consider all subproblems based on the shortest path method.

---

[1]During the activation of the electrode, the errors in control pins may be occur. In addition, each used cell needs to activate the corresponding electrode. Therefore, for better reliability, the used cells should be minimized.
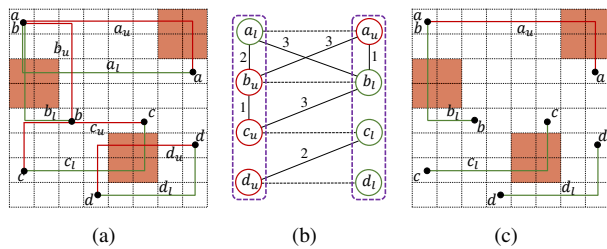
IEEE *Access*

FIGURE 5: (a) Bounding boxes and their possible L-shape routes for all nets. (b) Cross graph constructed. (c) Initial routing result by solving the BLP problem.

A. INITIAL ROUTING SCHEME

In this work, we first concurrently assign an L-shape route for all two-pin nets. The bounding box of a two-pin net consists of an upper L-shape route (n_u) and a lower L-shape route (n_l). The first step of our top-down routing scheme is deciding one possible L-shape route between n_u and n_l for every two-pin net such that the total intersecting cells are minimized. Before that, we construct a cross graph for all n_u and n_l of nets.

Definition 1 (cross graph): A cross graph $CG = (V, NE, CE, W)$ is an undirected and weighted graph, where vertex $v_i \in V$ represents n_u (or n_l) of a net. If vertices v_i and v_j belong to the same net, then there exists a net edge $ne_{ij} \in NE$ between them. If vertices v_i and v_j belong to two different nets and cross at some spot, then there exists a cross edge $ce_{ij} \in CE$. $w_{ij} \in W$ represents the weight of cross edge $ce_{ij} \in CE$, and w_{ij} equals the number of intersecting cells between v_i and v_j.

Fig. 5 shows an example of cross graph construction, where the red cells represent modules. Given the possible n_u and n_l of all nets in Fig. 5(a), according to the definition of cross graph, we can construct a cross graph of Fig. 5(a) as Fig. 5(b). The solid lines and dashed lines represent cross edges and net edges, respectively. The number on every edge represents its weight.

Since every net has two possible L-shape candidates (n_u and n_l), we must decide which is the better one. Thus, if two vertices v_i and v_j are connected by two different net edges, then only one of v_i and v_j can be chosen. In addition, the weight of a cross edge is the number of intersecting cells, and the objective of this step is to minimize the number of intersecting cells. Therefore, this problem is equal to dividing the cross graph into a bipartite graph, where minimizing the sum of all the edge weights within one part of the bipartite graph and making the net edges fall between the bipartite graph. Let binary variable u_i denote whether vertex v_i is selected. If $u_i = 1$, then v_i is selected, and vice versa. Further, we formulate this problem to binary linear programming (BLP) problem as follows:

$$\min \quad \sum_{i \in V} \sum_{j \in V} z_{ij} \cdot w_{ij} \qquad (1)$$

$$\text{s.t.} \quad u_i + u_j \leq 1 + z_{ij}, \qquad \forall e_{ij} \in CE; \qquad (1a)$$

$$u_i + u_j \leq 1, \qquad \forall e_{ij} \in NE; \qquad (1b)$$

$$u_i, z_{ij} \in \{0, 1\}, \qquad \forall i, j \in V. \qquad (1c)$$

In the above BLP, constraint (1a) establishes a correspondence between vertices and edges. That is, if vertices v_i and v_j are selected, then weight w_{ij} is added to the objective value. Constraint (1b) ensures that, if vertices v_i and v_j are connected by a net edge, then only one of v_i and v_j can be selected. We can quickly solve this BLP problem by calling LP solver. Fig. 5(c) shows the initial routing result by solving the BLP problem.

B. ROUTE CANDIDATES GENERATION

After initial routing by solving BLP (1), we obtain an L-shape route for every net in subproblems. However, these L-shape routes may be illegal, since they may overlap with modules. Therefore, we should modify these illegal L-shape routes by local flipping of L-shape routes such that all droplet routes do not overlap with modules.

These illegal routes can be grouped into two classes. The first one is the routes that overlap with the corners of modules, as the solid line route shown in Fig. 6(a). The second one is the routes that horizontally or vertically go through modules, as the solid line route shown in Fig. 6(b). Furthermore, to handle these two different types of illegal routes, we first define two different rules as follows:

1) Rule-1. We can easily find two cells (corner cell) outside the overlapping module, and then we can fix the overlapping by flipping the sub-route between the two cells. As the dashed line shown in Fig. 6(a).

2) Rule-2. We can easily find two cells (corner cell) outside the overlapping module, and then we can fix the overlapping by detouring the route around the modules. As the dashed line shown in Fig. 6(b).
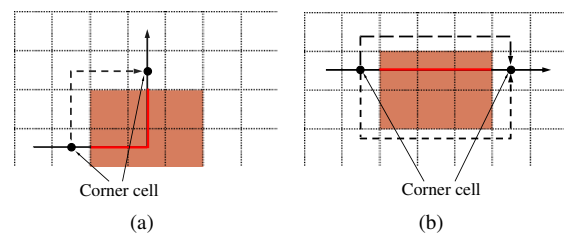

FIGURE 6: Two approaches of blockage-aware path legalization. The solid line and dashed line indicate respectively the original route and the modified route. (a) Rule-1. (b) Rule-2.

Under Rule-1 and Rule-2, we can achieve various legal route candidates by choosing different two corner cells and locally modifying the shapes of route candidates. We set a

IEEE *Access*

Unified Contamination-Aware Routing Method Considering Realistic Washing Capacity Constraint in Digital Microfluidic Biochips
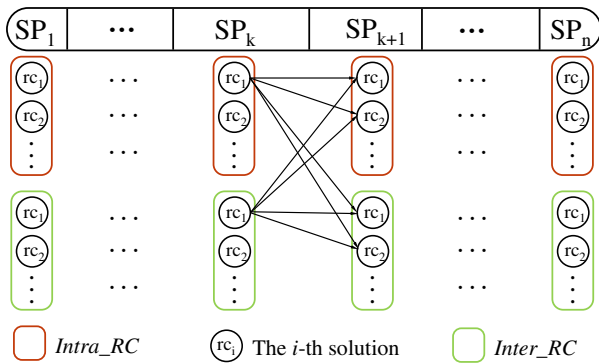

FIGURE 7: Optimal route selection by solving the shortest path problem of a directed graph.

cost criterion to weight the quality of selected route candidates in a subproblem as

$$cost_{Intra} = N_{ic} + \alpha_1 \cdot L_p \qquad (2)$$

where N_{ic} denotes the number of intersecting cells between paths, L_p represents the total length of all paths, and α_1 is a user-defined parameter. After the above legalization operation, we may obtain several route candidates with minimum $cost_{Intra}$, which are added to the set $Intra\_RC$ of a subproblem. However, $cost_{Intra}$ ignores the contaminations from its adjacent subproblems.

To simultaneously consider all subproblems, we set another cost criterion to select desirable route candidates as follows:

$$cost_{Inter} = N_{inter} + \alpha_2 \cdot N_{ic} + \beta_2 \cdot L_p \qquad (3)$$

where N_{inter} represents the number of inter-contamination spots, α_2 and β_2 are the user-defined parameters. By minimizing $cost_{Inter}$, we may obtain some other route candidates, which are added to the set $Inter\_RC$ of a subproblem.

C. OPTIMAL ROUTE SELECTION BY SHORTEST PATH

In the above subsection, we generate several route candidates for every net in each subproblem. After that, we should select a desirable route for every net such that the total numbers of contaminations (includes intra-contaminations and inter-contaminations) are minimized.

To achieve this goal, we construct a shortest path model based on the directed graph DG as shown in Fig. 7. In the directed graph DG, SP_k, $k = 1, ..., n$, represent the k-th subproblem, where n is the number of subproblems. Each subproblem contains two route candidate sets $Intra\_RC$ and $Inter\_RC$, and there are several route candidates in the sets $Intra\_RC$ and $Inter\_RC$. A vertex in the directed graph represents a route candidate solution of a subproblem. For every two vertices from two successive subproblems, there exists a directed edge between them. Moreover, the weight of an edge is calculated by the number of cross-contaminations of two route candidates.

Finally, by solving the shortest path problem on the constructed DG, we can achieve desirable routes for all subproblems.

IV. SCHEDULING OF FUNCTIONAL DROPLETS

After obtaining a path for every functional droplet from its source cell to its target cell, we arrange a schedule for every functional droplet along its routed path. In the scheduling process, we need to ensure that these walking functional droplets satisfy the fluidic constraint and the timing constraint. To achieve this objective, we first analyze cross cells that will affect the functional droplet scheduling, and propose a method based on ILP to determine the time that a functional droplet arrives at its corresponding decision cell. Then, the ILP-based method is extended to handle the general case.

A. ILP-BASED SCHEDULING AT CROSS CELLS

After the routing process, two different routing paths of functional droplets may cross at some cells, namely cross cells. The formal definition of cross cell is described as follows.

Definition 2 (cross cell): A cross cell is a cell in an array where two different routing paths crossed.

In addition, to better describe the relationships between cells, we introduce the concept of adjacent cell. Let $C(i, k)$ be the k-th cell of path p_i from the source to target.

Definition 3 (adjacent cell): Cells $C(i, k)$ and $C(j, l)$ $(i \neq j)$ are adjacent cells mutually if

$$|x_{ik} - x_{jl}| \leq 1 \ and \ |y_{ik} - y_{jl}| \leq 1 \qquad (4)$$

where (x_{ik}, y_{ik}) and (x_{jl}, y_{jl}) represent the positions of cells $C(i, k)$ and $C(j, l)$, respectively.

Apparently, if two different routing paths of functional droplets are mutually disjoint (satisfying fluidic constraint), then the functional droplets along these two paths can walk at the same time without fluidic constraint violation; otherwise, the functional droplets along these two paths may violate the fluidic constraint at a cross cell. As a result, a decision problem should be resolved at the cross cell. That is, we should decide the order of passing every cross cell of two cross routing paths. We also call this problem as scheduling at cross cells.

To satisfy the fluidic constraint at cross cell, an apparent observation is made as follows:

Observation 1: If droplet d_i on path p_i reaches a cross cell ($C(i, k)$ or $C(j, l)$) before droplet d_j on path p_j, then d_i should reach $C(i, k-1)$ and $C(i, k+1)$ before d_j reaching $C(j, l-1)$ and $C(j, l+1)$.

Therefore, at each cross cell ($C(i, k)$ or $C(j, l)$) of two paths p_i and p_j, we only need to decide which droplet between d_i and d_j first reaches cell $C(i, k-1)$ or $C(j, l-1)$ on the respective paths. We call cell $C(i, k-1)$ or $C(j, l-1)$ as decision cell. The formal definition is stated as follows:

Definition 4 (decision cell):

Both cells $C(i, k)$ on path p_i and $C(j, l)$ on path p_j are decision cells when the following conditions are satisfied:

1) $C(i, k)$ and $C(j, l)$ are adjacent cells;
2) $C(i, k-1)$ and $C(j, l)$ are not adjacent cells;
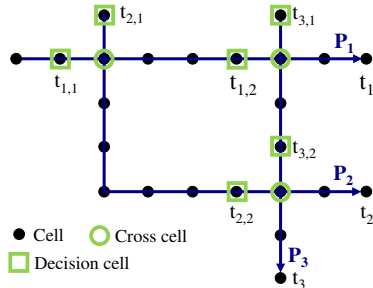3) $C(i, k)$ and $C(j, l-1)$ are not adjacent cells.

FIGURE 8: Model for scheduling at cross cells.

Our scheduling model of droplets at cross cells is shown in Fig. 8. The droplet scheduling problem aims to determine the time of a droplet arriving at each cell on its routing path. We handle the droplet scheduling problem by two steps. First, according to *Observation* 1, we should decide the order of passing a cross cell among two cross routing paths, and further decide the times of droplets arriving at the decision cell.

Let $t_{i,k}$ denotes the time of droplet d_i reaching a decision cell $C(i, k)$, and let $P_{i,k}$ represents the set of cells from the decision cell $C(i, k)$ on path p_i to its next decision cell. Specially, $P_{i,0}$ represents the set of cells from the source cell to its first decision cell on path p_i, and P_{i,n_i} represents the set of cells from the last decision cell $C(i, n_i)$ to its target cell on path p_i.

As shown in Fig. 8, we first give the values of $t_{1,1}$ and $t_{2,1}$, $t_{3,1}$ and $t_{1,2}$, $t_{3,2}$ and $t_{2,2}$. Then, for other cells between successive decision cells, we only need to let the droplets pass directly to determine its time.

To determine the time of droplets to decision cells, we formulate the problem to an ILP. The detailed constraints and objective are demonstrated in the following.

The most crucial constraint is the fluidic constraint (including static constraint and dynamic constraint). In Fig. 9, the blue and red arrow-lines denote paths p_i and p_j, respectively. The blue and red numbers on the cells represent the time of droplets d_i and d_j reaching the cell, respectively. As shown in Fig. 9 (a), the droplets d_i and d_j on paths p_i and p_j may reach the cross cell CC at the same time, which may violate fluidic constraint. As Fig. 9 (b) shown, delaying the time of one droplet to its corresponding decision cell is intuitive to avoid violating fluidic constraint.

From Observation 1, we know that fluidic constraint may be violated between adjacent decision cells. Therefore, the fluidic constraint can be rewritten as follows:

$$t_{i,k} - t_{j,l} \geq w_{jl,ik} \ or \ t_{j,l} - t_{i,k} \geq w_{ik,jl} \quad (5)$$

where the decision cells $C(i, k)$ and $C(j, l)$ are adjacent cells, and the constant $w_{ik,jl}$ represents the minimum time interval between the droplets reaching $C(i, k)$ and $C(j, l)$ on the basis
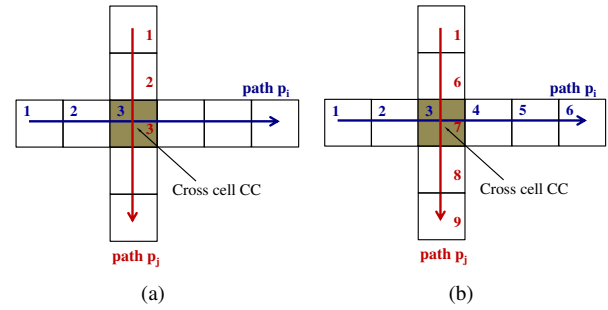

FIGURE 9: An example of fluidic constraint that may be violated at the cross cell CC.

that the fluidic constraint is not violated if droplet on path p_i first pass $C(i, k)$.

Note that the "or" operation in constraint (5) cannot be directly supported in a linear program. Let constant M be a large number, then the previous constraint can be transformed equivalently to

$$
\begin{aligned}
t_{i,k} - t_{j,l} &\geq w_{jl,ik} - M \cdot b_{ik,jl}, \\
t_{j,l} - t_{i,k} &\geq w_{ik,jl} - M \cdot (1 - b_{ik,jl}), \quad (6) \\
b_{ik,jl} &\in \{0, 1\}.
\end{aligned}
$$

Furthermore, the timing constraint can be rewritten as:

$$t_i \leq T_{max} \quad (7)$$

where $t_i = t_{i,n_i} + |P_{i,n_i}|$ denotes the time of droplet d_i from its source cell to its target cell.

For all decision cells on a path, we know that the closer to target cell, the later droplets arrive. Therefore, for any two successive decision cells $C(i, j-1)$ and $C(i, j)$ on a path p_i, we have:

$$t_{i,j} \geq t_{i,j-1} + |P_{i,j-1}| + 1 \quad (8)$$

For a droplet from its source cell, we initialize $t = 1$. Then we can formally formulate the corresponding ILP model as follows:

$$\min \ \max_i \{t_i\} \quad (9)$$
$$\text{s.t.} \quad (6)(7)(8).$$

B. EXTENDING SCHEDULING TO GENERAL CASE

Unlike the scheduling problem at cross cells, in the general case, it is possible that fluidic constraint may also be violated at adjacent cells of different paths. Thus, the decision problem not only occurs at the cross cells, but also occurs among adjacent cells on two paths. In this section, we extend our ILP based scheduling at cross cells in Subsection IV-A to the general case. To solve the general case scheduling problem, we introduce the concepts of sub-path and adjacent path.

Definition 5 (sub-path): A set of successive cells in path p_i is called a sub-path sp_i of path p_i.

Definition 6 (adjacent path): Let $C(p_i)$ represents the set of all cells in path p_i. Paths p_i and p_j $(i \neq j)$ are adjacent paths mutually if the following two conditions are satisfied:

IEEE *Access*

Unified Contamination-Aware Routing Method Considering Realistic Washing Capacity Constraint in Digital Microfluidic Biochips

1) for all $C(i,k) \in C(p_i)$, there exists $C(j,l) \in C(p_j)$, such that $C(i,k)$ and $C(j,l)$ are adjacent cells;

2) for all $C(j,l) \in C(p_j)$, there exists $C(i,k) \in C(p_i)$, such that $C(j,l)$ and $C(i,k)$ are adjacent cells.

According to the definition of adjacent path, for source cells of any two adjacent paths, we have the following claim.

Claim 1: If cells c_i and c_j are the source cells of adjacent paths p_i and p_j respectively, then c_i and c_j are decision cells.

Similar to Observation 1, for each pair of adjacent sub-paths, we can draw the following claim.

Claim 2: If paths sp_i and sp_j are sub-paths of path p_i and p_j respectively, and the droplet d_i on p_i reaches the source cell of sp_i first. To avoid fluidic constraint being violated, we need: for all $C(i,k) \in C(sp_i)$ and for all $C(j,l) \in C(sp_j)$, $t_{i,k} > t_{j,l}$, where $C(i,k)$ and $C(j,l)$ are adjacent cells.

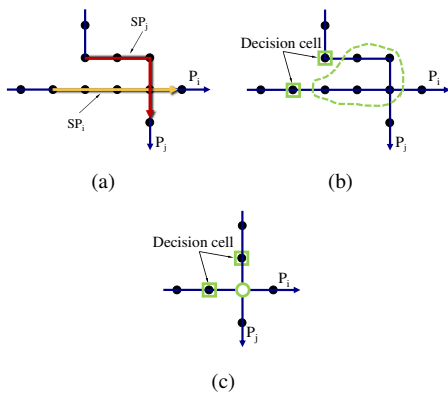FIGURE 10: (a) Adjacent sub-paths sp_i and sp_j. (b) Contraction process of adjacent sub-paths sp_i and sp_j. (c) The results of contraction.

Therefore, by Claim 1 and Claim 2, for adjacent sub-paths of any two paths, we can contract the cells between their source cell and target cell into a cross cell, as shown in Fig. 10. Finally, the droplet scheduling problem in the general case can be transformed into a scheduling problem at cross cells.

Specially, when the source cell of path p_i is adjacent to a cell in the path p_j or the target cell of path p_j is adjacent to a cell in the path p_i, letting the droplet on the path p_i first pass through these adjacent cells can prevent the fluidic constraint from being violated.

Therefore, if the target cell of the path p_j is adjacent to the cell $C(i,k)$ in the path p_i, we need the following inequality

$$t_{j,n_j} - t_{i,k} \geq w_{jn_j,ik} \tag{10}$$

where n_j denotes the last decision cell of path p_i.

If the source cell $C(i,1)$ of the path p_i is adjacent to a cell in the path p_j, then $C(i,1)$ is a decision cell by Claim 1, and we initialize the droplet d_i on path p_i at the cell on $t=1$. Therefore, we only need to find the decision cell $C(j,l)$ on the path p_j that is adjacent to $C(i,1)$, so that the relationship between $C(i,1)$ and $C(j,l)$ satisfies constraint (5).

As a result, the scheduling model of the general case is obtained by adding constraint (10) into the ILP model (9).

V. WASHING OF INTRA- AND INTER-CONTAMINATIONS

After the functional droplets routing and scheduling, the positions and probable times of contamination spots are known. Then the washing stage is proposed to clean the contamination spots left by the previous droplets. In this section, we first analyze different types of inter-contamination spots, and then we assign contamination spots to be washed to adjacent subproblems to reduce the execution time. Furthermore, we design a desirable washing path for each washing droplet for each subproblem to improve efficiency. Finally, a compact algorithm is presented to simultaneously compact all droplets, while minimizing the execution time of bioassay.

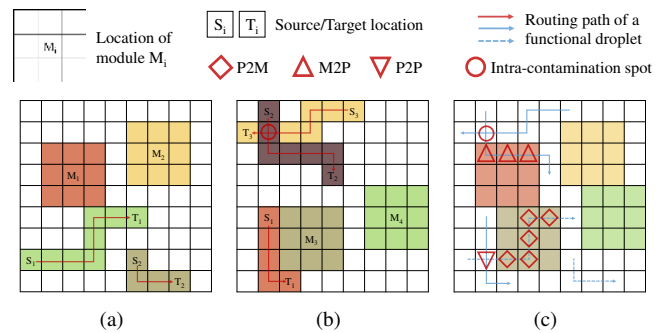### A. CONTAMINATION SPOTS ANALYSIS AND ASSIGNMENT

FIGURE 11: Situations where a subproblem SP_k may be contaminated. (a) Routing solution of SP_{k-1}. (b) Routing solution of SP_k. (c) All potential contamination spots of SP_k.

In this paper, we not only consider the intra-contamination spots within a subproblem, but also consider the inter-contamination spots in successive subproblems. Unlike that intra-contamination spots can be washed in the corresponding subproblems directly, it is much difficult to handle the inter-contamination spots since we must consider subproblems unify and thus large solution space. In the washing stage, we focus on handling the inter-contamination spots.

For successive subproblems, as shown in Fig. 11, we know that the subproblem SP_{k-1} is bounded to leave liquid residues at used cells. If these cells will be used again in the subproblem SP_k, they are inter-contamination spots. According to different residue sources, we can more specifically classify inter-contamination spots in SP_k into the following three categories:

- M2P: formation of inter-contamination by the residues on modules of SP_{k-1} and the cells in paths of SP_k;
- P2M: formation of inter-contamination by the residues on paths of SP_{k-1} and the cells in modules of SP_k;
- P2P: formation of inter-contamination by the residues on paths of SP_{k-1} and the cells in paths of SP_k;

As illustrated in Fig. 3, if we schedule an extra washing operation between successive subproblems to clean the residues, there still exist many residues that cannot be cleaned

due to the fluidic constraint. However, if these residues are assigned to adjacent subproblems, the number of contamination spots would be reduced and washed. For example, the P2Ms of subproblem SP_k can be cleaned in SP_{k-1}. Similarly, we can clean M2Ps of subproblem SP_k in SP_k, and clean P2Ps of subproblem SP_{k+1} in SP_k or SP_{k+1}, respectively.

Therefore, a reasonable contamination spot assignment principle can effectively improve the washing effect of contamination spots. And in subproblem SP_k, we only need to wash the following four types of contamination spots: (1) Intra-contamination spots in SP_k; (2) M2Ps of SP_k; (3) P2Ms of SP_{k+1}; (4) P2Ps of SP_{k+1}.

After the contamination spots being assigned, we can know the specific contamination spots to be washed in each subproblem. By analyzing the types of contamination spots, we can know which subproblems are related to the formation of the current contamination spots. At a contamination spot, suppose the arrival time of the previous droplet is t_{early}, while the arrival time of the latter droplet is t_{late}. Then the arrival time $t_{washing}$ of washing droplet should be between t_{early} and t_{late}, i.e., $t_{washing} \in (t_{early}, t_{late})$.

For the contamination spots formed by the previous subproblems, we assume its $t_{early} = 0$. Similarly, for the contamination spot that will be formed within the next subproblem, we assume $t_{late} = T_{max}$. For the contamination spots formed within the previous subproblem (include the intra-contamination spots), similar to the previous work [24], we can relax the washing duration (t_{early}, t_{late}) to achieve better washing effect. The maximum relaxation time allowed for contamination spot is $t_{relax} = T_{max} - t_{used}$, where t_{used} represents the routing length of the second droplet from its source cell to the contamination spot. Therefore, for a contamination spot , its relaxed washing duration is

$$t_{washing}^{relax} \in (t_{early}, t_{late} + t_{relax}) \quad (11)$$

B. WASHING DROPLETS ROUTING

To improve the efficiency of washing droplets, we design a desirable washing path for each washing droplet. If the t_{late} of contamination spot cs_i is smaller than that of contamination spot cs_j, then cs_i is more urgent to be cleaned. In addition, the distance between a washing droplet wd and a contamination spot is also a crucial factor. It is obvious that, the closer the distance, the more priority is considered washing by wd. It must be noted that, these distances t_{trans}, e.g., the times of transporting washing droplet to contamination spots, can be obtained by the A* routing algorithm [30].

Therefore, for each washing droplet wd, we first construct a feasible contamination spots candidate set in a square search area around wd. The radius of the square search area will gradually get larger until feasible contamination spots are found. Here, a feasible contamination spot refers to a cell at which the washing droplet wd can arrive in time in its relaxed washing duration. Then, a contamination spot that will be washed is selected from this candidate set according
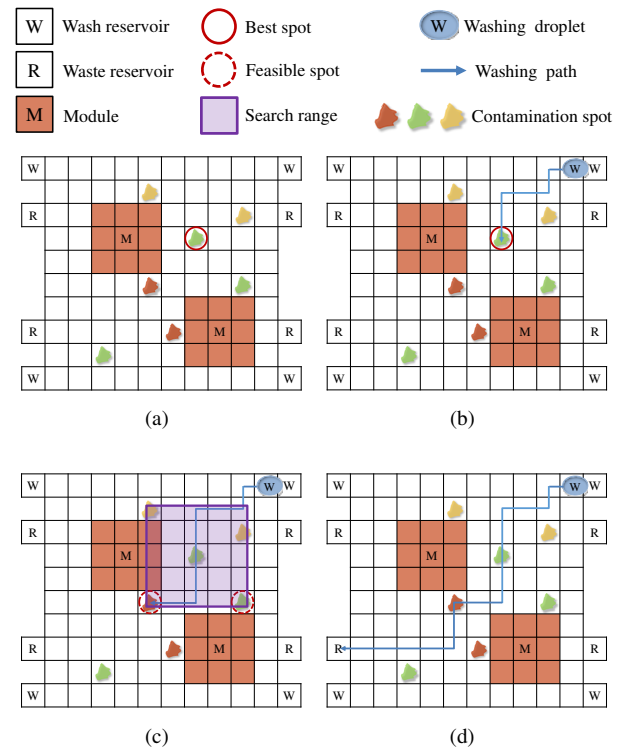

FIGURE 12: Describes the process of washing droplet routing. (a) The first contamination spot to be cleaned is selected. (b) The best washing port is selected while the corresponding washing path is generated. (c) Two feasible contamination spots were found. (d) When the washing capacity of the washing droplet is exhausted, the washing droplet will exit the chip and its corresponding washing path is generated.

to

$$Cost_{spot} = t_{trans} + \alpha_3 \cdot t_{late} \quad (12)$$

where t_{trans} indicates the time of the washing droplet from the current position to the feasible contamination spot, t_{late} denotes the arrival time of the second functional droplet at this feasible contamination spot, and α_3 is a user-defined parameter.

The washing droplet routing is demonstrated as shown in Fig. 12. Here the red, green and orange contamination spot represent the contamination spots of the residue in the previous subproblem, the intra-contamination spots and the contamination spots that will contaminate the next subproblem, respectively. Firstly, we initialize a washing droplet for each wash reservoir. Then, the feasible contamination spot candidate set is constructed by simultaneously searching their own search areas of initialized washing droplets. As shown in Fig. 12(a), we select a contamination spot cs based on the lowest $cost_{spot}$, and start to wash it using the corresponding washing droplet wd. Meanwhile, as shown in Fig. 12(b), wd is routed from its wash reservoir to the position of cs by using the A* routing algorithm. Then, as shown in Fig. 12(c), a search area of square is constructed to generate a feasible contamination spots candidate set based

IEEE *Access*

Unified Contamination-Aware Routing Method Considering Realistic Washing Capacity Constraint in Digital Microfluidic Biochips

---

Algorithm 1 Functional and washing droplet compaction

Input: Set of functional and washing paths $P = \{p_1, p_2, ..., p_m\}$ and the timing constraint T_{max}.

Output: The scheduled paths set P does not violate any constraints.

1   Set m to be the number of set P;

   for $t = 0$ *to* T_{max} **do**

2   Each droplet moves forward along its path;

     $restart := false$;

     for $i = 0$ *to* m **do**

3   **if** *path p_i violates fluidic constraints at time t* **then**

4   Find the set of paths $\bar{P}$ that violate the fluidic constraint with the path p_i at time t;

       foreach *path $p_j \in \bar{P}$* **do**

5   **if** *path p_j takes precedence over path p_i at time t* **then**

6   Delay the droplet on the path p_i before the time $t - 1$;

         $restart := true$;

         $t := 0$;

         break;

7   **if** $restart = true$ **then**

8   break;

9   Reset all the paths in P;

---

on the current position of wd, and Equation (12) is used to select the next target position for washing droplet wd. For each time of washing, as shown in Fig. 12(d) we update the remaining washing capacity of wd until the washing capacity is exhausted. wd will be routed to one of the waste reservoirs by using the A* routing algorithm. As a result, the washing path of wd is generated.

Secondly, the above process is repeated until all contamination spots are washed or the remaining contamination spots cannot be washed for the limited washing capacity of washing droplets. Finally, all washing paths of this subproblem are generated.

C. FUNCTIONAL AND WASHING DROPLET COMPACTION

After all the washing paths have been generated, the following is to schedule the function droplets and the washing droplets along their paths, such that they do not violate the fluidic constraint and contamination constraint while minimizing the time of scheduling.

Algorithm 1 gives an overview of our functional and washing compaction algorithm. In our compaction algorithm, every step of the droplet moves without any constraint. If the path p_i violates the fluidic constraint at time t, then the set $\bar{P}$ consisting of paths that violate fluidic constraints at time t is first constructed (Line 4 in Algorithm 1). Further, for any path $p_j \in \bar{P}$, path p_i passes before p_j at time t, if and only if one of the following four conditions is satisfied (Line 5 in

Algorithm 1).

- If paths p_i and p_j are both functional paths, and the path p_i passes preferentially during the functional path scheduling stage.
- If path p_i is a functional path and path p_j a is washing path, and the washing droplet in the path p_j is not responsible for the contamination spot left by path p_i after time t.
- If path p_i is a washing path and path p_j is a functional path, and the washing droplet in the path p_i is not responsible for the contamination spot left by path p_j after time t.
- If paths p_i and p_j are both washing paths, and the length of path p_i is greater than path p_j.

At time t, if path p_i is passed after path $p_j \in \bar{P}$, path p_i will be delayed to find a position before the time $t - 1$ such that path p_i does not violate fluidic constraints with any other paths (Line 5-6 in Algorithm 1). Thus, by using this iterative approach, the fluidic constraints between droplets can be eliminated. Finally, the functional and washing droplets are simultaneously scheduled.

VI. EXPERIMENTAL RESULTS

We implemented the proposed algorithm for the unified contamination aware routing optimization in the C++ programming language on a 2.67 GHz 64-bit Linux machine with 6GB memory. The commonly used bioassay provided by [24] was used to verify the effectiveness of our proposed algorithm. Table 1 lists the basic statistics result of these bioassays. "Size" gives the size of DMFBs array, "#Sub" represents the number of subproblems, "#Net" records the number of nets, "#D_{max}" denotes the maximum number of functional droplets within one subproblem, "#WP" lists the number of wash reservoirs. Washing capacity is 4.

TABLE 1: Statistics of the routing benchmarks

Benchmark	Size	#Sub	#Net	#D_{max}	#WP
In-vitro_1	16x16	11	28	5	4
In-vitro_2	14x14	15	35	6	4
Protein_1	21x21	64	181	6	4
Protein_2	13x13	78	178	6	4

In the experiments, α_1 in Equation (2) was set as 0.25. α_2 and β_2 in Equation (3) were set as 0.4 and 0.1. α_3 in Equation (12) was set as 0.25. We conducted three experiments to verify the performance of our routing and washing algorithm. The corresponding experimental results are illustrated in the following subsections.

A. PERFORMANCE OF UNIFIED CONTAMINATION-AWARE ROUTING

In the first experiment, to evaluate the performance of simultaneous consideration of subproblem, we compared the results obtained by simultaneous consideration ("Ours w. SS") with the results achieved by separately processing the subproblems ("Ours w.o. SS"). The comparisons are listed

TABLE 2: Simulation results with versus whether to consider washing the inter-contamination spots and washing capacity constraint is four

Benchmarks	Ours w.o. SS							Ours w. SS					
	#CSS	#Intra	#W	#UC	T_r	CPU(s)	T_r^*	#CSS	#Cont	#W	#UC	T_r	CPU(s)
In-vitro_1	12	7	5	444	239	0.24	443	6	78	38	1112	389	0.58
In-vitro_2	20	5	4	444	221	0.20	491	7	69	34	997	442	0.74
Protein_1	100	14	9	2433	1253	0.79	3366	28	283	151	6350	2967	1.68
Protein_2	203	27	17	1587	940	0.63	2138	54	272	149	3618	1992	1.36
Total	335	53	35	4908	2653	1.86	6438	95	702	372	12077	5790	4.36
Ratio	3.53	0.08	0.09	0.41	0.46	0.43	1.11	1.00	1.00	1.00	1.00	1.00	1.00

T_r^*: The total execution time of a bioassay, which equals T_r in "Ours w.o. SS" plus the execution time spending on washing inter-contamination spots between all successive subproblems.

TABLE 3: Comparisons of our method without and with applying shortest-path algorithm

Benchmarks	Ours (non-SP)						Ours (SP)					
	#Intra	#Inter	#Cont	#UC	T_r	CPU(s)	#Intra	#Inter	#Cont	#UC	T_r	CPU(s)
In-vitro_1	7	108	115	224	162	0.10	7	71	78	274	177	0.15
In-vitro_2	5	85	90	275	182	0.20	5	64	69	291	182	0.24
Protein_1	15	399	414	1720	1055	0.51	14	269	283	1837	1068	0.90
Protein_2	23	357	380	1023	831	0.49	27	245	272	1098	816	0.73
Total	50	949	999	3262	2230	1.30	53	649	702	3500	2243	2.02
Ratio	1.00	1.00	1.00	1.00	1.00	1.00	1.06	0.68	0.70	1.07	1.01	1.55

in Table 2. In Table 2, "#CCS" indicates the number of contamination spots that have not been washed; "#Intra" records the number of intra-contamination spots; "#Cont" records the total number of cross-contamination spots, which includes both the intra-contamination and inter-contamination spots; "#W" represents the total number of used washing droplets; "#UC" denotes the number of used cells; "T_r" represents the execution time of a bioassay; and "CPU(s)" denotes the runtime in second.

From Table 2, the number of intra-contamination spots in "#Intra" only occupies 8% of the total contamination spots in "#Cont". This fully demonstrates that contaminations are mainly formed between successive subproblems. In addition, compared with "Ours w.o. SS", "Ours w. SS" needs more washing droplets to wash contamination spots. Thus, there must be more "#UC" in "Ours w. SS" than in "Ours w.o. SS".

From the "#CSS" in "Ours w.o. SS" and "Ours w. SS", the contamination spots are all successfully washed. In addition, from the comparison between "T_r^*" of "Ours w.o. SS" and "T_r" of "Ours w. SS", the total execution time of "Ours w.o. SS" is 11% more than that of "Ours w. SS". This improvement also shows the effectiveness of our simultaneous routing and washing of all subproblems. Consequently, the key benefit of simultaneous consideration of subproblems is that: it can not only significantly reduce 72% contamination spots, but also save the execution time of a bioassay.

B. PERFORMANCE OF DROPLET ROUTING AND SCHEDULING

In the second experiment, to evaluate the reduction of inter-contamination spots by using the shortest-path algorithm, we compared the results obtained without using shortest-path algorithm ("Ours (non-SP)") with the results achieved by shortest-path algorithm ("Ours (SP)"). The comparisons are listed in Table 3. In this table, "#Inter" indicates the number of inter-contamination spots and other parameter names have the same meaning as in Table 2.

From Table 3, our shortest-path algorithm can reduce "#Inter" by 32% at the cost of a small increase in "#Intra" and "T_r". The significant reduction in "#Inter" fully demonstrates the effectiveness of the proposed algorithm.

In the third experiment, we compare the performance of our droplet routing algorithm considering cross-contamination spots with the works [24], [26]. The comparisons are listed in Table 4. Compared with the work [26], our method reduces the respective "#UC" and "T_r" by 44% and 32% with a small increase of "#Intra". Compare with work [24], with small increase in "CPU(s)", our method reduces the "#Intra", "#UC" and "T_r" by 76%, 23% and 14%, respectively. Therefore, our droplet routing method can significantly reduce the used cells and the execution time of bioassays in an acceptable range of intra-contamination spots.

C. PERFORMANCE OF OUR WASHING ALGORITHM

To further verify the effectiveness of our proposed algorithm, we also compared our proposed method with the work in [24]. Since the work [24] did not consider simultaneous subproblem optimization, we compare the results in [24] with "Ours w.o. SS" in Table 5. In this table, "#Fail" indicates the number of contamination spots that have not been washed and other parameter names have the same meaning as in Table 2. Thanks to binary code provided by [24], the experiment is executed on the same platform for fair comparison.

Under the same configuration, "#Cont" and "#UC" in [24] are 3.94× and 35% more than those in "Ours w.o. SS",

IEEE Access

Unified Contamination-Aware Routing Method Considering Realistic Washing Capacity Constraint in Digital Microfluidic Biochips

TABLE 4: Comparison of cross-contamination aware routing results

Benchmarks	DAC'10 [26]				TCAD'16 [24]				Ours (non-SP)			
	#Intra	#UC	T_r	CPU(s)	#Intra	#UC	T_r	CPU(s)	#Intra	#UC	T_r	CPU(s)
In-vitro_1	4	621	268	0.06	31	326	186	0.05	7	224	162	0.10
In-vitro_2	0	423	224	0.03	34	348	219	0.05	5	275	182	0.20
Protein_1	18	3215	1508	0.23	69	2121	1176	0.39	15	1720	1055	0.51
Protein_2	11	1574	1287	0.14	75	1463	1005	0.16	23	1023	831	0.49
Total	33	5833	3287	0.46	209	4258	2586	0.65	50	3262	2230	1.30

TABLE 5: No consideration of inter contamination spots and the washing capacity constraint is four

Benchmarks	TCAD'16 [24]						Ours w.o. SS					
	#Cont	#Fail	#W	#UC	T_r	CPU(s)	#Cont	#Fail	#W	#UC	T_r	CPU(s)
In-vitro_1	31	14	9	571	444	0.12	7	2	5	444	239	0.24
In-vitro_2	34	7	12	582	432	0.17	5	1	4	444	221	0.20
Protein_1	69	9	40	3438	1724	0.66	14	4	9	2433	1253	0.79
Protein_2	75	7	32	2046	1318	0.30	27	3	17	1587	940	0.63
Total	209	37	93	6637	3918	1.25	53	10	35	4908	2653	1.86
Ratio	3.94	3.70	2.66	1.35	1.48	0.67	1.00	1.00	1.00	1.00	1.00	1.00

respectively. The significant improvements verify the effectiveness of our routing algorithm. Moreover, compared with [24], "Ours w.o. SS" achieves 3.70× less "#Fail" and 2.66× "#W". But above all, the total execution time in [24] is 48% more than that in "Ours w.o. SS". These fully demonstrate that our routing, scheduling and washing algorithms are very effective.

D. PERFORMANCE OF WASHING CAPACITY

We also evaluated the impact of different capacities of washing droplets. The comparison of the experiments is shown in Fig. 13. From the figure, as the capacity of the washing droplet increases, more contamination spots can be washed by the washing droplets. Many contamination spots cannot be successfully washed when the capacity of the washing droplet is small. The reason is that, there may be too many ordinary residues (not contamination spots) around certain contamination spots, the washing droplet consumes all of its washing capacity before it reaches these contamination spots [24]. Therefore, in order to obtain a better washing effect, we can appropriately increase the washing capacity of washing droplets (e.g., increase the capacity of washing droplets). As shown in Fig. 13, the numbers of unwashed contamination spots in benchmarks In-vitro_1, In-vitro_2, Protein_1 and Protein_2 are nearly 0 if the washing capacities of the washing droplets are greater than 7.

VII. CONCLUSIONS

Unlike previous works, we presented a routing flow in this paper to practically resolve both the intra-contaminations and inter-contaminations with washing capacity constraint by simultaneously considering all subproblems in DMFBs. We first presented a top-down scheme to generate candidates of routing paths, then constructed a shortest-path model to select desirable routing paths for all subproblems. With the decision diagram of droplets at cross cells, we further proposed an ILP formulation to compact the execution time. Finally,

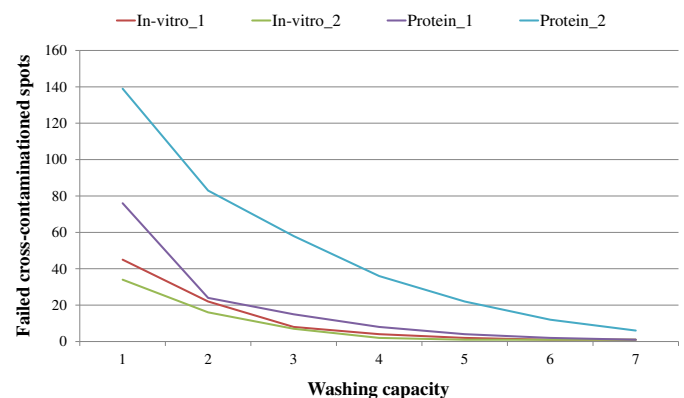
Using different capacities of washing droplets

FIGURE 13: Experimental results on different capacities of the washing droplets.

contamination removal by washing droplets with washing capacity was considered for all subproblems. Experimental results validated the effectiveness of our proposed method. Particularly, simultaneous consideration of subproblems can not only significantly reduce contamination spots completely, but also save the execution time of a bioassay.

ACKNOWLEDGMENTS

REFERENCES
[1] W. H. Minhass, J. McDaniel, M. Raagaard, P. Brisk, P. Pop, and J. Madsen, "Scheduling and fluid routing for flow-based microfluidic laboratories-on-a-chip," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 3, pp. 615–628, 2018.
[2] V. Srinivasan, V. K. Pamula, and R. B. Fair, "An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids," Lab on a Chip, vol. 4, no. 4, pp. 310–315, 2004.

Unified Contamination-Aware Routing Method Considering Realistic Washing Capacity Constraint in Digital Microfluidic Biochips

[3] R. B. Fair, A. Khlystov, T. D. Tailor, V. Ivanov, R. D. Evans, V. Srinivasan, V. K. Pamula, M. G. Pollack, P. B. Griffin, and J. Zhou, "Chemical and biological applications of digital-microfluidic devices," IEEE Design & Test of Computers, vol. 24, no. 1, pp. 10–24, 2007.

[4] I. Barbulovic-Nad, H. Yang, P. S. Park, and A. R. Wheeler, "Digital microfluidics for cell-based assays," Lab on a Chip, vol. 8, no. 4, pp. 519–526, 2008.

[5] C.-H. Kuo, G.-R. Lu, T.-Y. Ho, H.-M. Chen, and S. Hu, "Placement optimization of cyber-physical digital microfluidic biochips," in Proceedings of IEEE Biomedical Circuits and Systems Conference, pp. 448–451, 2016.

[6] Y.-H. Chen, C.-L. Hsu, L.-C. Tsai, T.-W. Huang, and T.-Y. Ho, "A reliability-oriented placement algorithm for reconfigurable digital microfluidic biochips using 3-d deferred decision making technique," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 32, no. 8, pp. 1151–1162, 2013.

[7] F. Su and K. Chakrabarty, "Architectural-level synthesis of digital microfluidics-based biochips," in Proceedings of IEEE/ACM International Conference on Computer-Aided Design, pp. 223–228, 2004.

[8] Y. Luo, K. Chakrabarty, and T.-Y. Ho, Hardware/Software Co-design and Optimization for Cyberphysical Integration in Digital Microfluidic Biochips. Springer, 2015.

[9] K. Chakrabarty and T. Xu, Digital Microfluidic Biochips: Design Automation and Optimization. CRC press, 2010.

[10] Y. Zhao and K. Chakrabarty, Design and Testing of Digital Microfluidic Biochips. Springer Science & Business Media, 2012.

[11] Y. Zhao and K. Chakrabarty, "Cross-contamination avoidance for droplet routing in digital microfluidic biochips," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 6, pp. 817–830, 2012.

[12] P. Roy, P. Howladar, R. Bhattacharjee, H. Rahaman, and P. Dasgupta, "A new cross contamination aware routing method with intelligent path exploration in digital microfluidic biochips," in Proceedings of International Conference on Design & Technology of Integrated Systems in Nanoscale Era, pp. 50–55, 2013.

[13] S. K. Cho, S. K. Fan, H. Moon, and C. J. Kim, "Towards digital microfluidic circuits: creating, transporting, cutting and merging liquid droplets by electrowetting-based actuation," in Proceedings of IEEE International Conference on Micro Electro Mechanical Systems, pp. 32–35, 2002.

[14] 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, 2002.

[15] Z. Xiao and E. F. Y. Young, "Droplet-routing-aware module placement for cross-referencing biochips," in Proceedings of ACM International Symposium on Physical Design, pp. 193–199, 2010.

[16] S. Windh, C. Phung, D. T. Grissom, P. Pop, and P. Brisk, "Performance improvements and congestion reduction for routing-based synthesis for digital microfluidic biochips," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, no. 1, pp. 41–54, 2017.

[17] M. Cho and D. Z. Pan, "A high-performance droplet routing algorithm for digital microfluidic biochips," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 10, pp. 1714–1724, 2008.

[18] F. Su and K. Chakrabarty, Digital Microfluidic Biochips: Synthesis, Testing, and Reconfiguration Techniques. CRC Press, 2006.

[19] F. Su and K. Chakrabarty, "Unified high-level synthesis and module placement for defect-tolerant microfluidic biochips," in Proceedings of ACM/IEEE Design Automation Conference, pp. 825–830, 2005.

[20] P. Pop, M. Alistar, E. Stuart, and J. Madsen, Fault-Tolerant Digital Microfluidic Biochips. Springer, 2016.

[21] T. Xu and K. Chakrabarty, "Integrated droplet routing in the synthesis of microfluidic biochips," in Proceedings of ACM/IEEE Design Automation Conference, pp. 948–953, 2007.

[22] P. H. Yuh, C. L. Yang, and Y. W. Chang, "Bioroute: A network-flow-based routing algorithm for the synthesis of digital microfluidic biochips," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 11, pp. 1928–1941, 2008.

[23] P. H. Yuh, S. S. Sapatnekar, C. L. Yang, and Y. W. Chang, "A progressive-ilp-based routing algorithm for the synthesis of cross-referencing biochips," in Proceedings of ACM/IEEE Design Automation Conference, pp. 284–289, 2009.

[24] H. Yao, Q. Wang, Y. Shen, T.-Y. Ho, and Y. Cai, "Integrated functional and washing routing optimization for cross-contamination removal in digital microfluidic biochips," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 8, pp. 1283–1296, 2016.

[25] F. Su, K. Chakrabarty, and R. B. Fair, "Microfluidics-based biochips: technology issues, implementation platforms, and design automation challenges," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 2, pp. 211–223, 2006.

[26] Y. Zhao and K. Chakrabarty, "Synchronization of washing operations with droplet routing for cross-contamination avoidance in digital microfluidic biochips," in Proceedings of ACM/IEEE Design Automation Conference, pp. 635–640, 2010.

[27] T.-W. Huang, C.-H. Lin, and T.-Y. Ho, "A contamination aware droplet routing algorithm for the synthesis of digital microfluidic biochips," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 29, no. 11, pp. 1682–1695, 2010.

[28] D. Mitra, S. Ghoshal, H. Rahaman, K. Chakrabarty, and B. B. Bhattacharya, "On residue removal in digital microfluidic biochips," in Proceedings of the on Great Lakes Symposium on VLSI, pp. 391–394, 2011.

[29] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Transactions on Systems Science and Cybernetics, vol. SSC-4, no. 2, pp. 100–107, 1968.

[30] G. W. Clow, "A global routing algorithm for general cells," in Proceedings of ACM/IEEE 21st Design Automation Conference, 1984.

ZHIPENG HUANG received the B.Sc. degree in information and computing sciences from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, in 2015, and the M.Sc. degree in applied mathematics from the Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou, China, in 2018.

Currently, he is working toward the Ph.D. degree in the Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou, China. His research interests include optimization theory and applications, and VLSI placement.

XIQIONG BAI received the B.Sc. Degree in mathematics and information science, Jiangxi Normal University, Nanchang, China, in 2016 and the M.Sc. degree in applied mathematics from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, in 2019.
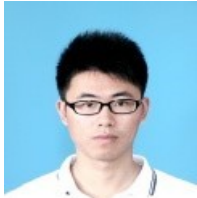
Currently, she is working toward the Ph.D. degree in the Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou, China. Her research interest includes VLSI routing.

TINGSHEN LAN received the B.Sc. degree in Mathematics and Applied Mathematics from the School of Mathematics and Statistics, South-Central University for Nationalities, Wuhan, China, in 2016, and the M.Sc. degree in Operations Research and Cybernetics from the Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou, China, in 2019. His research interests include optimization theory and applications, and biochip.

IEEE *Access*

Unified Contamination-Aware Routing Method Considering Realistic Washing Capacity Constraint in Digital Microfluidic Biochips

XINGQUAN LI received the B.Sc. and the Ph.D. degrees in applied mathematics from Fuzhou University, Fuzhou, China, in 2013, and 2018, respectively. He is currently a Associate Professor at the School of Mathematics and Statistics, Minnan Normal University, Zhangzhou, China. His current research interests include VLSI physical design and design for manufacturability, and graph neural network. He has authored over 20 papers in refereed journals and conferences. Dr. Li was recipients of the First Place in the ICCAD 2017 Contest and the First Place in the ICCAD 2018 Contest, He received Excellent Doctor Degree Dissertation Award in Fujian and Graduate National Scholarship.

GENG LIN received the B.S. degree in Information and Computing Sciences, the M.S. degree in Operations Research and Cybernetics, and the Ph. D. degree in Applied Mathematics, all from Fuzhou University, Fuzhou, China, in2004, 2007, and 2010, respectively. He is currently a Professor with the College of Mathematics and Data Science, Minjiang University. His research interests include combinatorial optimization and artificial intelligence. He is the author or coauthor of scientific papers in refereed journals including IEEE Transactions on Evolutionary Computation, IEEE Transactions on Computers, Journal of Global Optimization, Computers and Operations Research, Information Sciences, Annals of Operations Research, and INFORMS Journal on Computing.

. . .