PathDriver-Wash: A Path-Driven Wash Optimization Method for Continuous-Flow Lab-on-a-Chip Systems

Xing Huang[1], Jiaxuan Wang[1], Zhiwen Yu[2,1*], Bin Guo[1], Tsung-Yi Ho[3], Ulf Schlichtmann[4], and Krishnendu Chakrabarty[5]
[1]School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]College of Computer Science and Technology, Harbin Engineering University, Harbin, China
[3]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong
[4]Chair of Electronic Design Automation, Technical University of Munich, Munich, Germany
[5]School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe AZ, USA
{xing.huang, zhiwenyu, guob}@nwpu.edu.cn, wjx99@mail.nwpu.edu.cn, tyho@cse.cuhk.edu.hk, ulf.schlichtmann@tum.de, krishnendu.chakrabarty@asu.edu

***Abstract*—Rapid advances in microfluidics technologies have facilitated the emergence of highly integrated lab-on-a-chip (LoC) biochip systems. With such coin-sized biochips, complicated bioassay procedures can be executed efficiently without any human intervention. To ensure the correctness and precision of assay outcomes, however, cross-contamination among different fluid samples/reagents needs to be dealt with separately during assay execution. As a consequence, wash operations have to be introduced and a wash path network needs to be established on the chip to remove the residues left in flow channels. To realize optimized assay procedures with efficient wash operations, we propose *PathDriver-Wash* in this paper, a path-driven wash optimization method for continuous-flow LoC biochip systems. The proposed method includes the following three key techniques: 1) The necessity of contamination removals is analyzed systemically to avoid unnecessary wash operations, 2) wash operations are integrated with the regular removal of excess fluids, so that extra path occupations caused by wash can be minimized, and 3) optimized wash paths and time windows are computed and assigned to wash operations, so that the completion time of assays can be minimized. Experimental results demonstrate that the proposed method leads to highly efficient wash procedures as well as minimized assay completion times.**

***Index Terms*—Continuous-flow LoC biochip systems, contamination removal, microfluidics technologies, wash optimization.**

I. Introduction

The emergence of continuous-flow lab-on-a-chip (LoC) biochip systems has revolutionized laboratory procedures in biology and biochemistry [1]. By precisely controlling fluid samples/reagents in nanoliter volumes, various bioassays such as protein crystallization [2] and nucleic-acid isolation [3], can be performed automatically within a coin-sized chip area based on pre-customized assay plans. Besides, since LoC systems also have the advantages of high efficiency, high reliability, and low cost [1], they are expected to be a practical substitute for traditional manual laboratory procedures.

Fig. 1(a)-(b) show the schematic of a continuous-flow LoC biochip system, where a flow layer and a control layer form the basic structure of the chip [4]. In the flow layer, fluid samples/reagents can be injected into the chip via flow ports, and then transported to the target devices through etched flow channels to perform various biochemical operations such as mixing and heating. In the control layer, air pressure is generated by external pressure sources connecting to control ports and prorogated in the control channels, so that the elastomer membranes deployed in the overlapping area of the two layers, referred to as microvalves, can be pushed down into flow channels, thereby turning off fluid flows [5]. In contrast, microvalves return to their original positions after air pressure is released, thus resuming the transportation of fluids.

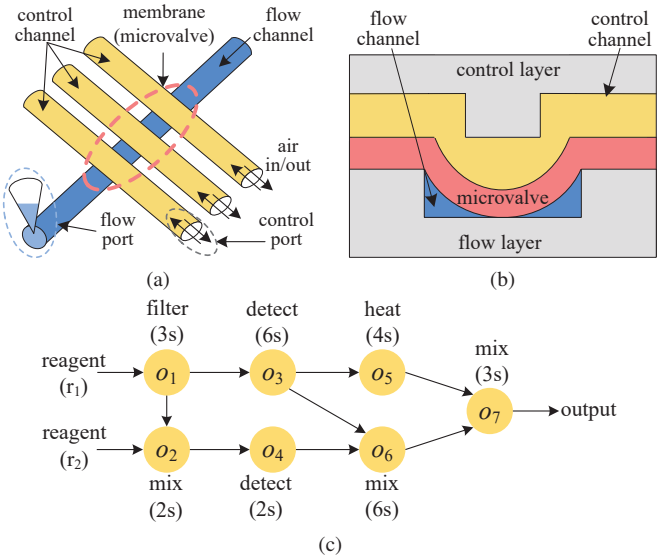

Fig. 1: (a) Schematic of a LoC biochip system, (b) the front view of a closed valve, and (c) sequencing graph of a bioassay.

Despite the aforementioned advantages of biochips, experimental precision is still a major concern during assay execution since cross-contamination among fluids can occur anywhere on the chip. If this problem is not addressed, incorrect assay outcomes can be generated and this, undoubtedly, will cause serious consequences in outcome-sensitive applications such as medical diagnosis. Chemiluminescence Immune Assay, for example, is a detection measure widely used in in-vitro diagnosis, through which several medical indexes such as tumormarker can be identified [6]. When two fluidic flows carrying different luminescence agents flow through a channel sequentially for tumormarker detections, the second luminescence agent can be contaminated due to the residue left behind by the first flow. During the luminous reaction, abnormal luminescent intensity will then be displayed, leading to an incorrect detection result.

As a consequence, wash operations are introduced to the LoC biochip systems, so that cross-contamination among fluids can be avoided [8]. This can be implemented by injecting buffer fluids from flow ports into the chip and flushing contaminated

979-8-3503-4859-0/DATE24/© 2024 EDAA
Authorized licensed use limited to: Fuzhou University. Downloaded on December 17,2024 at 03:26:50 UTC from IEEE Xplore. Restrictions apply.

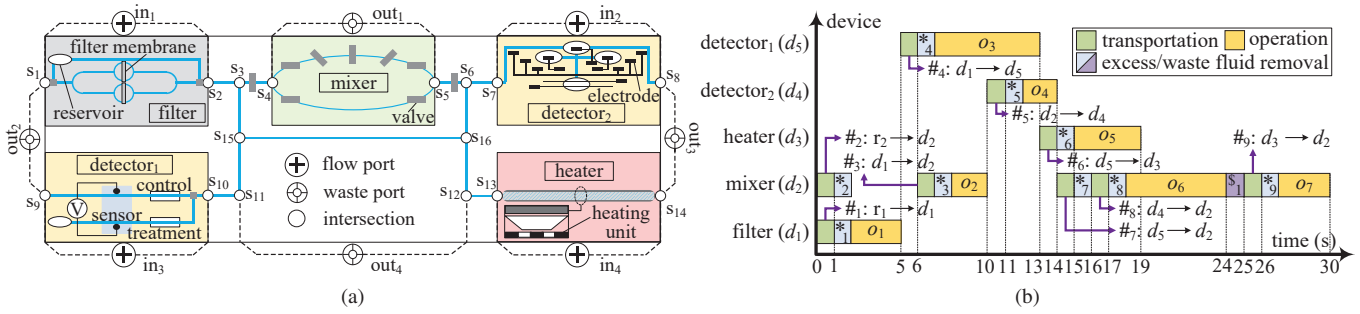

Fig. 2: (a) The chip layout constructed for the assay shown in Fig. 1(c) and (b) the corresponding assay scheduling scheme [7].

spots with predefined wash paths. Moreover, waste ports need to be deployed at the end of wash paths, so that waste fluids and the air already present in channels can be released. Thus, the flow port, contaminated spots, waste port, as well as flow channels connecting them form a complete wash path, i.e., ⌈flow port→contaminated spots→waste port⌋. Note that since wash operations need to be considered jointly with biochemical operations, reagent transportation, and excess fluid removal [7], the design of wash schemes is very complex.

Up to now, only a few automation methods have been proposed to address the wash optimization problem. In [8], a graph-based method is proposed by formulating wash optimization as a hitting-set problem. This method, however, is performed statically based on a given set of contaminated spots. In other words, wash operations are not incorporated into assay executions in a real-time manner. In [9], a demand-driven wash heuristic is implemented during application mapping. Since this method tries to postpone wash operations as long as possible until the contaminated resources need to be reused, conflicts between wash operations and fluid transportation occur frequently, leading to serious delay in assay completion. Recently, a sweepline-based heuristic is proposed in [10] to realize efficient wash in a distributed channel storage architecture. However, wash paths are established independently in this method without considering resource sharing, leading to high chip cost and prolonged assay execution.

In view of the importance of wash optimization as well as the drawbacks suffered by existing approaches, we propose a practical path-driven wash optimization method called *PathDriver-Wash* (PDW). Our contributions are listed below.

- PDW takes the complete wash paths into account and considers wash requirements, biochemical operations, fluid transportation/removal jointly, leading to a practical wash optimization tool for continuous-flow LoC systems.
- PDW conducts an in-depth analysis of wash necessity regarding contaminated resources to avoid unnecessary wash operations, thereby reducing the extra cost, e.g., wash paths and buffer fluids, introduced by wash.
- PDW integrates wash operations with the regular removal of excess fluids during assay execution, thus effectively reducing the extra channel occupation caused by wash.
- PDW computes optimized wash paths and time windows for wash operations, leading to minimized completion time of bioassays.

The remainder of this paper is organized as follows. In Section II, motivation, design challenges, and problem formulation of this work are introduced. Section III discusses the proposed method in detail. Experimental results are reported in Section IV. Finally, conclusions are drawn in Section V.

II. Motivation, Challenges, and Problem

The protocol of a bioassay is usually modeled as a sequencing graph $G(O, E)$, where O is a set of biochemical operations with specific execution times and E indicates the dependencies between these operations. For example, Fig. 1(c) shows the sequencing graph of a bioassay, which takes two input reagents r_1 and r_2 and processes them with seven biochemical operations o_1-o_7 to generate the corresponding assay outcome. To automatically perform this assay using an LoC system, Fig. 2(a)-(b) show the biochip layout and the scheduling scheme (i.e., execution procedure) generated by the synthesis tool in [7]. Moreover, the complete flow paths used in Fig. 2(b) are listed in Table I. It can be seen that all the biochemical operations are completed in 30 seconds.

A. Necessity Analysis of Wash Operations

Although the above assay can be performed efficiently using the chip layout and assay scheduling shown in Fig. 2, it does not take the cross-contamination between fluids into account, and thus may lead to incorrect assay outcomes. On the other hand, however, if all the contaminated resources are washed immediately during assay execution, a large number of channels will be occupied by buffer fluids and the completion of the assay will be delayed significantly. In fact, there are some channels/devices that do not need to be washed even if they are contaminated, including the following three categories:

- Type 1: *Contaminated channels/devices that are not used anymore in the subsequent assay procedure.* For example, after operation o_3 is finished Fig. 2(b), device detector$_1$ is contaminated. Then, the resulting fluid of o_3 is transported from detector$_1$ to mixer using flow path $\#_7$ in the time slot (14 s, 15 s), further leading to a contaminated flow path $s_{10} \rightarrow s_{11} \rightarrow s_{15} \rightarrow s_3 \rightarrow s_4$. However, since detector$_1$ will not be used for any biochemical operation and the contaminated path will not be used by any transportation task in the subsequent steps, the corresponding wash operations can be avoided.
- Type 2: *Contaminated channels/devices that are used to transport/manipulate the same type of fluids.* For example,

Authorized licensed use limited to: Fuzhou University. Downloaded on December 17,2024 at 03:26:50 UTC from IEEE Xplore. Restrictions apply.

TABLE I: The complete flow paths used in Fig. 2(b) and Fig. 3

flow paths of transportation			
$\#_1$	$in_1 \rightarrow s_2 \rightarrow$ filter $\rightarrow s_1 \rightarrow out_2$	$\#_2$	$in_2 \rightarrow s_7 \rightarrow s_6 \rightarrow s_5 \rightarrow$ mixer $\rightarrow s_4 \rightarrow out_1$
$\#_3$	$in_1 \rightarrow s_1 \rightarrow$ filter $\rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow$ mixer $\rightarrow s_5 \rightarrow out_1$	$\#_4$	$in_1 \rightarrow s_1 \rightarrow$ filter $\rightarrow s_2 \rightarrow s_3 \rightarrow s_{15} \rightarrow s_{11} \rightarrow s_{10} \rightarrow$ det.$_1 \rightarrow s_9 \rightarrow out_2$
$\#_5$	$in_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow$ mixer $\rightarrow s_5 \rightarrow s_6 \rightarrow s_7 \rightarrow$ det.$_2 \rightarrow s_8 \rightarrow out_3$	$\#_6$	$in_3 \rightarrow s_9 \rightarrow$ det.$_1 \rightarrow s_{10} \rightarrow s_{11} \rightarrow s_{15} \rightarrow s_{16} \rightarrow s_{12} \rightarrow s_{13} \rightarrow$ heater $\rightarrow s_{14} \rightarrow out_3$
$\#_7$	$in_3 \rightarrow s_9 \rightarrow$ det.$_1 \rightarrow s_{10} \rightarrow s_{11} \rightarrow s_{15} \rightarrow s_3 \rightarrow s_4 \rightarrow$ mixer $\rightarrow s_5 \rightarrow out_1$	$\#_8$	$in_2 \rightarrow s_8 \rightarrow$ det.$_2 \rightarrow s_7 \rightarrow s_6 \rightarrow s_5 \rightarrow$ mixer $\rightarrow s_4 \rightarrow out_1$
$\#_9$	$in_4 \rightarrow s_{14} \rightarrow$ heater $\rightarrow s_{13} \rightarrow s_{12} \rightarrow s_{16} \rightarrow s_6 \rightarrow s_5 \rightarrow$ mixer $\rightarrow s_4 \rightarrow out_1$		

flow paths of excess (*)/waste fluid removal ($)			
$*_1$	$in_1 \rightarrow s_1 \rightarrow out_2$; $in_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow out_1$	$*_2, *_3, *_7, *_8, *_9$	$in_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow out_1$; $in_2 \rightarrow s_7 \rightarrow s_6 \rightarrow s_5 \rightarrow out_1$
$*_4$	$in_3 \rightarrow s_9 \rightarrow out_2$; $in_3 \rightarrow s_{10} \rightarrow s_{11} \rightarrow out_4$	$*_5$	$in_2 \rightarrow s_8 \rightarrow out_3$; $in_2 \rightarrow s_7 \rightarrow s_6 \rightarrow s_5 \rightarrow out_1$
$*_6$	$in_4 \rightarrow s_{14} \rightarrow out_3$; $in_4 \rightarrow s_{13} \rightarrow s_{12} \rightarrow out_4$	$\$_1$	$in_2 \rightarrow s_7 \rightarrow s_6 \rightarrow s_5 \rightarrow$ mixer $\rightarrow s_4 \rightarrow out_1$

flow paths of wash operations			
w_1	$in_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow out_1$	w_2	$in_2 \rightarrow s_7 \rightarrow s_6 \rightarrow s_5 \rightarrow out_1$
w_3	$in_4 \rightarrow s_{13} \rightarrow s_{12} \rightarrow s_{16} \rightarrow s_{15} \rightarrow s_{11} \rightarrow out_4$		

after operation o_2 is finished in Fig. 2(b), the resulting fluid is transported from mixer to detector$_2$ using flow path $\#_5$, leading to a contaminated flow path $s_5 \rightarrow s_6 \rightarrow s_7$. This path, however, will be then used to transport the result of o_4, i.e., the same fluid of the previous one, from detector$_2$ to mixer in the time slot (16 s, 17 s). Thus, the corresponding wash operation can be avoided. In particular, if the residue left in a device has the same type as the subsequent input fluid, wash operation for the device can also be avoided.

- Type 3: *Contaminated channels that are used to transport waste fluids.* For example, after operation o_4 is finished in Fig. 2(b), the resulting fluid is transported from detector$_2$ to mixer using flow path $\#_8$, leading to a contaminated path $s_4 \rightarrow$ out$_1$. However, since this path will only be used for the removal of waste fluid generated by o_6, i.e., $\$_1$ in Fig. 2(b), the corresponding wash operation can be avoided.

B. Integration of Wash with Excess Fluid Removal

As mentioned above, the introduction of wash operations can cause extra channel occupations. If dealt with inappropriately, conflicts between wash operations and fluid transportation may occur frequently, thus prolonging the completion of bioassays. On the other hand, when a fluid is transported to a target device, excess fluids will be cached at the two ends of the device due to the need for air release [7]. As a result, these fluids need to be removed using separated flow paths such as $*_1$ in Fig. 2(b) [7]. However, if we integrate wash operations with those excess fluid removals, i.e., using wash operations to deal with the removal of contaminants and excess fluids simultaneously, both resource occupation and execution efficiency can be optimized.

Let us revisit the chip layout and scheduling described in Fig. 2. Reagent r_2 is first injected into a mixer using path $\#_2$. Consequently, some excess fluid will be cached in channel segment $s_5 \rightarrow s_6$ and thus need to be removed independently using the second path listed in $*_2$. Afterwards, since path $s_5 \rightarrow s_6$ will be reused to transport the mixture of o_2 in the time slot (10 s, 11 s), it needs to be further washed after excess fluid removal, leading to two separate fluidic manipulations. In contrast, if we integrate the wash operation with excess fluid removal, i.e., introducing a wash operation with flow path $in_2 \rightarrow s_7 \rightarrow s_6 \rightarrow s_5 \rightarrow out_1$, both excess fluids and contaminants can be removed at the same time.

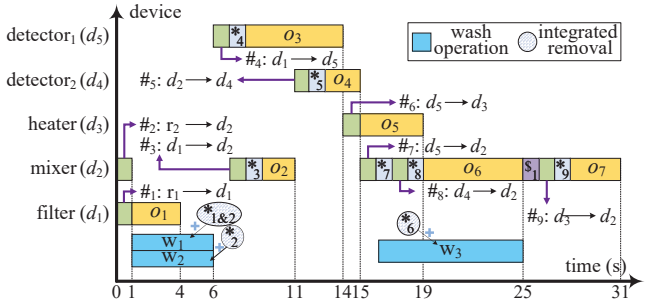

Fig. 3: An optimized scheduling with efficient wash operations.

C. Assignment of Optimized Wash Paths and Time Windows

To reduce the assay delay caused by wash as much as possible, optimized wash paths and time windows need to be computed, so that wash operations can be performed efficiently without introducing additional conflicts.

We take the chip layout and scheduling described in Fig. 2 as an example. After operation o_3 is finished in Fig. 2(b), the resulting fluid is transported from detector$_1$ to heater using path $\#_6$, leading to a contaminated path $s_{16} \rightarrow s_{12} \rightarrow s_{13}$. This path needs to be washed before the resulting fluid of o_5 is moved to the mixer. Hence, the best start point of the wash path is in_4 and there are three candidates that can be selected as its end point, namely out_1, out_2, and out_4. If out_1 is selected, the generated wash path passes through $s_6 \rightarrow s_5$ and it overlaps with the paths of transportation tasks using $\#_7$ and $\#_8$, thus significantly postponing the execution of operation o_6. Moreover, if out_2 is selected, a very long wash path passing though filter will be generated, while a conflict between wash and transportation task using $\#7$ is introduced. In contrast, if we select out_4 as the end point, a relatively short wash path, i.e., $in_4 \rightarrow s_{13} \rightarrow s_{12} \rightarrow s_{16} \rightarrow s_{15} \rightarrow s_{11} \rightarrow out_4$, can be generated. In the meantime, the wash operation can be executed concurrently with the transportation task using $\#8$.

Fig. 3 shows an optimized assay scheduling with efficient wash operations, where only three wash operations are introduced and they are executed concurrently with several other fluidic tasks. Moreover, excess fluid removals using $*_1$, $*_2$, and $*_6$ are integrated with wash operations, leading to only one second delay in assay completion.

D. Problem Formulation

Based on the above analysis, the wash optimization problem solved in this paper can be formulated as follows:

Given a bioassay modeled as a sequencing graph $G(O, E)$, a biochip architecture used for assay execution, and a scheduling of the assay, compute an optimized assay execution procedure with efficient wash operations such that the number of wash operations, the total length of wash paths, and the completion time of bioassays are minimized.

Authorized licensed use limited to: Fuzhou University. Downloaded on December 17,2024 at 03:26:50 UTC from IEEE Xplore. Restrictions apply.

III. Details of the Proposed Method

In this section, we discuss PDW in detail, taking all the design challenges described in Section II into account. The proposed method is formulated in terms of integer linear programming (ILP) problems and tackled using an ILP solver.

Given a chip architecture used for assay execution, PDW uses a virtual grid R of size $W_G \times H_G$ to represent the chip layout, where devices and channels are placed on the cells of R. Since the original scheduling of an assay can be disrupted once wash operations are introduced, the start times of fluidic manipulations such as biochemical operations need to be recomputed, so that the correctness of assay execution can be ensured.

For each operation $o_i \in O$, its execution should be adequate for the target protocol, and this can be constrained as

$$t^e_{o_i} - t^s_{o_i} \geq t(o_i),\ \forall o_i \in O, \tag{1}$$

where $t^s_{o_i}$ and $t^e_{o_i}$ represent the start time and end time of o_i, respectively. Note that $t(o_i)$ is the exsection time of o_i specified in G.

For each edge $e_{j,i} \in E$, o_i needs to be executed after the completion of o_j, constrained as

$$(1-\lambda_{j,i})M + t^s_{o_i} \geq t^e_{o_j},\ \forall e_{j,i} \in E, \tag{2}$$

where $\lambda_{j,i}$ is a 0-1 variable representing whether o_i is a child node of o_j in G. M is a very large constant used to transform the two situations indicated by $\lambda_{j,i}$ into linear constraints.

For simplicity, hereafter we use d_i and out_i to represent the device bound to o_i and the resulting fluid of o_i, respectively. When two operations o_i and o_j are bound to the same device, they cannot be executed concurrently. We therefore have

$$\begin{matrix}(1-\kappa)M + t^s_{o_j} \geq t^e_{o_i} \\ \kappa M + t^s_{o_i} \geq t^e_{o_j}\end{matrix},\ \forall o_i, o_j \in O, d_i = d_j \tag{3}$$

where κ is 0-1 variable representing the execution order of o_i and o_j, i.e., o_j (o_i) is first executed when $\kappa = 0$ ($\kappa = 1$).

Then, for each edge $e_{j,i} \in E$, the transportation of out_j from d_j to d_i, denoted by $p_{j,i,1}$, needs to be completed before the start time of o_i. This can be constrained as

$$t^s_{p_{j,i,1}} \geq t^e_{o_j},\ t^e_{p_{j,i,1}} \leq t^s_{o_i}\ ,\ \forall e_{j,i} \in E, \tag{4}$$

where $t^s_{p_{j,i,1}}$ and $t^e_{p_{j,i,1}}$ are the start time and end time of transportation task $p_{j,i,1}$, respectively.

After operation $p_{j,i,1}$ is completed, excess fluids cached at the two ends of d_i need to be removed. Similarly, we use $p_{j,i,2}$ to represent the removal of excess fluids, and this needs to be completed before o_i starts. Thus, we have

$$t^s_{p_{j,i,2}} \geq t^e_{p_{j,i,1}}, t^e_{p_{j,i,2}} \leq t^s_{o_i}\ ,\ \forall e_{j,i} \in E, \tag{5}$$

where $t^s_{p_{j,i,2}}$ and $t^e_{p_{j,i,2}}$ are the start time and end time of removal task $p_{j,i,2}$, respectively.

To ensure correct execution of tasks $p_{j,i,1}$ and $p_{j,i,2}$, they should satisfy the durations specified in the given assay scheduling (denoted by $T_{j,i,1}$ and $T_{j,i,2}$). Moreover, since an excess fluid removal task may be integrated into a wash operation as discussed in Section II-B, a 0-1 variable $\psi_{j,i,2}$ is used to represent whether $p_{j,i,2}$ is integrated into a wash operation. Note that the value of $\psi_{j,i,2}$ can be determined after wash paths are established. Then, we have

$$t^e_{p_{j,i,1}} - t^s_{p_{j,i,1}} \geq T_{j,i,1},\ \forall e_{j,i} \in E, \tag{6}$$

$$t^e_{p_{j,i,2}} - t^s_{p_{j,i,2}} \geq (1-\psi_{j,i,2})T_{j,i,2},\ \forall e_{j,i} \in E. \tag{7}$$

In addition, conflicting transportation/removal tasks using the same grid cells cannot be performed concurrently. We have

$$\begin{matrix}(1-\varepsilon)M + t^s_{p_{j,i,z}} \geq t^e_{p_{v,u,r}} \\ \varepsilon M + t^s_{p_{v,u,r}} \geq t^e_{p_{j,i,z}}\end{matrix}\ ,\ \forall e_{j,i}, e_{v,u} \in E, z, r = 1, 2,\ l_{j,i,z} \cap l_{v,u,r} \neq \varnothing, \tag{8}$$

where ε is a 0-1 variable representing the execution order of the two transportation/removal tasks. $l_{j,i,z}$/$l_{v,u,r}$ is the flow path of task $p_{j,i,z}$/$p_{v,u,r}$ specified in the given scheduling.

After a fluid transportation/removal task is completed, grid cells that are contaminated can be determined. We use R_c and $t^c_{x,y}$ to represent the set of contaminated grid cells and the time point that a cell $(x, y) \in R_c$ is contaminated, respectively. Then, the analysis of wash necessity can be performed based on the discussion in Section II-A.

For a contaminated cell (x, y), we use a 0-1 variable $a^1_{x,y}$ to represent whether the cell satisfies the requirements of Type 1 described in Section II-A. That is to say, if cell (x, y) is not used for any transportation task $p_{j,i,1}$ in the subsequent steps, washing (x, y) can be avoided. Thus, we have

$$a^1_{x,y} = \begin{cases} 1,\ (x,y) \notin l_{p_{j,i,1}} & \forall e_{j,i} \in E, \forall (x,y) \in R_c, \\ 0,\ otherwise & t^s_{p_{j,i,1}} \geq t^c_{x,y}, \end{cases} \tag{9}$$

Similarly, we use 0-1 variables $a^2_{x,y}$ and $a^3_{x,y}$ to represent whether a contaminated cell (x, y) satisfies the requirements of Type 2 and Type 3, respectively. Then, we have

$$a^2_{x,y} = \begin{cases} 1,\ S_T = 1 \\ 0,\ otherwise \end{cases},\ a^3_{x,y} = \begin{cases} 1,\ (x,y) \in l_{p_{j,i,1}} \wedge Q_{p_{j,i,1}} = 1 \\ 0,\ otherwise \end{cases},$$
$$\forall e_{j,i} \in E, \forall (x,y) \in R_c, t^s_{p_{j,i,1}} \geq t^c_{x,y}, \tag{10}$$

where S_T is a 0-1 variable indicating whether the fluid of $p_{j,i,1}$ is the same type as the contaminant and $Q_{p_{j,i,1}}$ is a 0-1 variable representing whether $p_{j,i,1}$ is used for waste fluid removal.

Then, we use a 0-1 variable $r_{x,y}$ to represent whether a contaminated cell (x, y) needs to be washed, constrained as

$$r_{x,y} = \begin{cases} 1,\ a^1_{x,y} = 0 \wedge a^2_{x,y} = 0 \wedge a^3_{x,y} = 0 \\ 0,\ otherwise \end{cases}, \forall (x,y) \in R_c. \tag{11}$$

After wash-necessity analysis of transportation/removal tasks, a set W_T of wash targets and a set W of wash operations are determined. It is then the task to establish an efficient wash path for each operation in W. For a wash operation w_j, its path l_{w_j} should start with a flow port and end with a waste port, constrained as

$$\sum_{fp_i \in F_p} u^j_{fp_i} = 1,\ \sum_{wp_i \in W_p} u^j_{wp_i} = 1,\ \forall w_j \in W, \tag{12}$$

where F_p and W_p are a set of flow ports and a set of waste ports on the given chip architecture, respectively. $u^j_{fp_i}$/$u^j_{wp_i}$ is a 0-1 variable representing whether a flow port fp_i/a waste port wp_i is allocated to the wash path l_{w_j}.

Authorized licensed use limited to: Fuzhou University. Downloaded on December 17,2024 at 03:26:50 UTC from IEEE Xplore. Restrictions apply.

With a flow port fp_i and a waste port wp_i allocated to l_{w_j}, only one adjacent cell of the two ports can be occupied by l_{w_j}. Thus, we have the following constraints

$$\sum_{(x,y)\in AC_{fp_i}} u^j_{x,y} = 1, \sum_{(x,y)\in AC_{wp_i}} u^j_{x,y} = 1, \ \forall w_j \in W, \quad (13)$$

where AC_{fp_i} and AC_{wp_i} are adjacent cells of fp_i and wp_i on the grid, respectively. In addition, $u^j_{x,y}$ is a 0-1 variable indicating whether cell (x, y) is occupied by l_{w_j}.

Next, for any cell (x, y) on path l_{w_j} except for the one adjacent to the flow/waste port, exactly two adjacent cells of (x, y) should also be on the path. We have

$$\sum_{(x',y')\in AC_{x,y}} u^j_{x',y'} = 2, \ \forall w_j \in W, (x,y) \notin AC_{fp_i} \cup AC_{wp_i}, \quad (14)$$

where $AC_{x,y}$ is a set of adjacent cells of (x, y) on the grid.

We assume that $wt_i \subseteq W_T$ is a set of cells that needs to be washed by w_j. Then, all these cells should be covered by the established wash path l_{w_j}. Hence we have the following constraint

$$\sum_{(x,y)\in wt_i} u^j_{x,y} \geq |wt_i|, \ \forall w_j \in W. \quad (15)$$

After establishing path l_{w_j}, a time window used for executing w_j needs to be computed. We use a variable $t_{j,e}$ to represent the time point that all the cells in wt_i are contaminated. Moreover, $t_{j,s}$ is the earliest time that wash requirements of wt_i is proposed. Then, we have

$$t^s_{w_j} \geq t_{j,e}, t^e_{w_j} \leq t_{j,s}, \ \forall w_j \in W, \quad (16)$$

where $t^s_{w_j}$ and $t^s_{w_j}$ are the start time and end time of wash operation w_j, respectively.

Furthermore, the duration of w_j is computed based on the flush time of buffer fluid and the dissolution time of contaminants [11], computed as

$$t(w_j) = L(l_{w_j})/v_f + t_d(w_j), \ \forall w_j \in W, \quad (17)$$

where $L(l_{w_j})$, v_f, and $t_d(w_j)$ are the length of l_{w_j}, the flow velocity, and the dissolution time, respectively.

To ensure that the wash is adequate, the execution of w_j should last for a sufficient time. Thus, we have

$$t^e_{w_j} - t^s_{w_j} \geq t(w_j), \ \forall w_j \in W. \quad (18)$$

To avoid resource conflicts, cells occupied by w_j cannot be used by other fluidic manipulations simultaneously, e.g., a transportation/removal task or another wash operation. Thus, we have

$$\begin{matrix}(1-\mu)M + t^s_{w_j} \geq t^e_{p_{v,u,r}} \\ \mu M + t^s_{p_{v,u,r}} \geq t^e_{w_j}\end{matrix}, \ \begin{matrix}\forall e_{v,u} \in E, r = 1, 2, \\ l_{w_j} \cap l_{p_{v,u,r}} \neq \varnothing,\end{matrix} \quad (19)$$

$$\begin{matrix}(1-\eta)M + t^s_{w_j} \geq t^e_{w_k} \\ \eta M + t^s_{w_k} \geq t^e_{w_j}\end{matrix}, \ \forall w_k \in W, l_{w_j} \cap l_{w_k} \neq \varnothing, \quad (20)$$

where μ is a 0-1 variable representing the execution order of w_j and a transportation/removal task, and η is a 0-1 variable representing the execution order of w_j and another wash operation.

For a removal task $p_{j,i,2}$ in (7) and a wash operation w_k, assume that $p_{v,u,1}$ is the next task that needs to use $l_{p_{j,i,2}}$. The variable $\psi_{j,i,2}$ in (7) can be expressed as

$$\psi_{j,i,2} = \begin{cases} 1, \ l_{p_{j,i,2}} \subseteq l_{w_k} \wedge t^s_{w_k} \geq t^e_{p_{j,i,1}} \wedge t^e_{w_k} \leq t^s_{p_{v,u,1}} \\ 0, \ otherwise \end{cases},$$
$$\forall e_{j,i} \in E, w_k \in W, l_{p_{j,i,2}} \cap l_{p_{v,u,1}} \neq \varnothing \wedge t^e_{p_{j,i,1}} \leq t^s_{p_{v,u,1}}. \quad (21)$$

After the execution of all the biochemical operations in O is completed, the completion time of the assay, denoted by T_{assay}, can be bounded by the following constraint

$$T_{assay} \geq t^e_{o_i}, \ \forall o_i \in O. \quad (22)$$

The number of wash operations, denoted by N_{wash}, can be computed as

$$N_{wash} = \sum_{z=1}^{2} \sum_{p_{j,i,z}} a^z_{j,i}, \ \forall e_{j,i} \in E, \quad (23)$$

where $a^z_{j,i}$ is a 0-1 variable representing whethe the path of $p_{j,i,z}$ needs to be washed and can be formulated as

$$a^z_{j,i} = \begin{cases} 1, \ \exists (x,y) \in l_{p_{j,i,z}}, r_{x,y} = 1, z = 1, 2 \\ 0, \ otherwise \end{cases}. \quad (24)$$

In addition, the total length of wash paths, denoted by L_{wash}, can be computed as

$$L_{wash} = \sum_{l_{w_j}\in L} \sum_{(x,y)\in l_{w_j}} u^j_{(x,y)}. \quad (25)$$

Finally, an optimized wash scheme minimizing the assay completion time, the total length of wash paths, and the number of wash operations can be determined by solving the optimization problem below:

$$\begin{aligned} &\text{minimize } \alpha \times N_{wash} + \beta \times L_{wash} + \gamma \times T_{assay} \quad (26) \\ &\text{subject to } (1) - (25). \end{aligned}$$

where α, β and γ are three weighting factors.

IV. Experimental Results

The proposed PathDriver-Wash was implemented and tested on a PC with 3.6-GHz CPU and 32 GB memory. The Gurobi tool was used to solve the ILP problems formulated in Section III. Eight benchmarks were used to verify PDW, including five real-life bioassays and three synthetic benchmarks. The details are listed in the Column 2 of Table II, where $|O|$, $|E|$, and $|D|$ are the numbers of biochemical operations and dependencies in the sequencing graph as well as the number of devices in the library, respectively. The chip architectures and assay schedulings are generated using the synthesis tool in [12]. Moreover, the parameters used in the experiments are set as follows: $\alpha = 0.3$, $\beta = 0.3$, $\gamma = 0.4$, and $v_f = 10$ mm/s [13]. The runtime on each benchmark of PDW was limited to 15 minutes to return the best-effort results.

To evaluate the performance of PDW, we compare it with the delay-aware wash optimization (DAWO) method proposed in [10]. In this method, wash operations are first introduced based on the positions of contaminated spots. Next, the breadth-first-search algorithm is employed to compute wash paths on

Authorized licensed use limited to: Fuzhou University. Downloaded on December 17,2024 at 03:26:50 UTC from IEEE Xplore. Restrictions apply.

TABLE II: Comparison results between PathDriver-Wash (PDW) and EDW regarding wash optimization

Benchmark	$\|O\|/\|D\|/\|E\|$	N_{wash}			L_{wash}(mm)			T_{delay} (s)			T_{assay}(s)		
		DAWO	PDW	I_m (%)	DAWO	PDW	I_m (%)	DAWO	PDW	I_m (%)	DAWO	PDW	I_m (%)
PCR	7/5/15	4	3	25.00	110	80	27.27	10	7	30.00	33	30	9.09
IVD	12/9/24	10	6	40.00	200	150	25.00	21	16	23.81	51	46	9.80
ProteinSplit	14/11/27	12	10	16.67	220	160	27.27	15	7	53.33	110	102	7.27
Kinase act-1	4/9/16	3	3	0.00	80	60	25.00	5	3	40.00	38	36	5.26
Kinase act-2	12/9/48	17	13	23.53	250	190	24.00	33	25	24.24	87	79	9.20
Synthetic1	10/12/15	10	8	20.00	290	220	24.14	19	13	31.58	58	52	10.34
Synthetic2	15/13/24	16	16	0.00	300	260	13.33	29	21	27.59	78	70	10.26
Synthetic3	20/18/28	18	15	16.67	460	320	30.43	35	23	34.29	92	80	13.04
Average	–	–		17.73	–		24.56	–		33.10	–		9.28

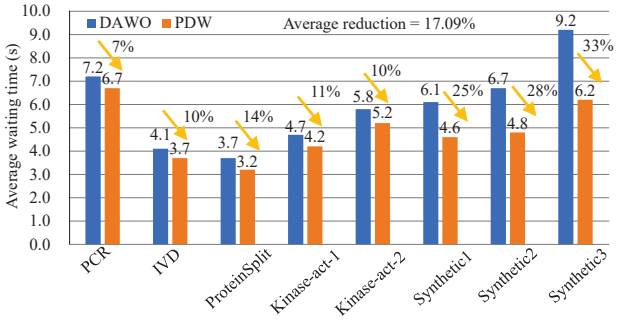

Fig. 4: Comparison results for the average waiting time.

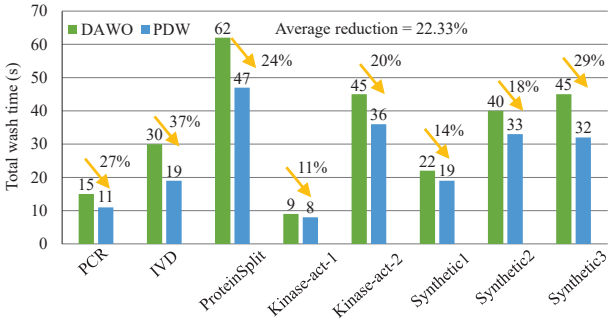

Fig. 5: Comparison results for the total wash time.

the chip. Moreover, a sweep-line method is used to assign wash operations to appropriate time intervals. To ensure a fair comparison, we implement both PDW and DAWO in Python and run them on the same PC for the aforementioned benchmarks.

Table II shows the comparison results, where columns N_{Wash}, L_{wash}, T_{delay}, and T_{assay} are the results of the number of wash operations, the total length of wash paths, the assay delay caused by wash operations, and the completion time of bioassays, respectively. Compared with DAWO, it can be seen that the number of wash operations is reduced by 17.73% on average across the benchmarks. Moreover, PDW achieves a 13.33%–30.43% reduction in the total length of wash paths, with an average reduction of 24.56%. We attribute these results to the following reasons: 1) Since PDW takes wash-necessity analysis into consideration, unnecessary wash operations are avoided completely and 2) since PDW formulates the wash optimization into an ILP problem, efficient wash paths can be established in a global manner compared with the heuristic adopted in DAWO.

Besides, since PDW assigns wash operations to optimized time windows, these operations can be executed concurrently with biochemical operations as long as there is no conflict among their wash paths, leading to shorter waiting time of biochemical operations as well as shorter total wash time as illustrated in Fig. 4 and Fig. 5. Correspondingly, it can be seen from Table II that the overall assay delay and the assay completion time are reduced by 33.10% and 9.28% on average, thus demonstrating the effectiveness of the proposed method.

V. Conclusion

We have investigated the wash optimization problem of continuous-flow LoC biochip systems and presented a systematic method called PathDriver-Wash to generate efficient wash schemes. By considering wash requirements, biochemical operations, fluid transportation/removal tasks simultaneously, unnecessary wash operations can be avoided and optimized wash paths/time windows can be computed efficiently. Moreover, by integrating wash operations with the regular removal of excess fluids, both extra resource occupation and assay execution efficiency can be optimized. Experimental results on multiple benchmarks have confirmed that the proposed method leads to fewer wash operations, more efficient wash paths, and shorter assay completion time.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China (No. 61960206008), the National Science Fund for Distinguished Young Scholars (No. 62025205), and the National Natural Science Fund of China for Excellent Young Scientists Fund Program (Overseas). K. Chakrabarty was not supported by any grant, either from the US or from China. Zhiwen Yu is the corresponding author.

References

[1] N. Convery and N. Gadegaard, "30 years of microfluidics," *Micro and Nano Engineering*, vol. 2, pp. 76–91, 2019.

[2] C. D. Chin, T. Laksanasopin, Y. K. Cheung, D. Steinmiller, V. Linder, H. Parsa, J. Wang, H. Moore, R. Rouse, G. Umviligihozo *et al.*, "Microfluidics-based diagnostics of infectious diseases in the developing world," *Nature medicine*, vol. 17, no. 8, pp. 1015–1019, 2011.

[3] J. W. Hong, Y. Chen, W. F. Anderson, and S. R. Quake, "Molecular biology on a microfluidic chip," *Journal of Physics: Condensed Matter*, vol. 18, no. 18, p. S691, 2006.

[4] X. Huang, Y. Pan, Z. Chen, W. Guo, R. Wille, T.-Y. Ho, and U. Schlichtmann, "BigIntegr: One-pass architectural synthesis for continuous-flow microfluidic lab-on-a-chip systems," in *Proc. Int. Conf. Comput.-Aided Des*, 2021, pp. 1–8.

[5] G. Liu, H. Huang, Z. Chen, H. Lin, H. Liu, X. Huang, and W. Guo, "Design automation for continuous-flow microfluidic biochips: A comprehensive review," *Integration*, vol. 82, pp. 48–66, 2022.

[6] T. H. Fereja, A. Hymete, and T. Gunasekaran, "A recent review on chemiluminescence reaction, principle and application on pharmaceutical analysis," *International Scholarly Research Notices*, vol. 2013, 2013.

[7] X. Huang, Y. Pan, G. L. Zhang, B. Li, W. Guo, T.-Y. Ho, and U. Schlichtmann, "PathDriver: A path-driven architectural synthesis flow for continuous-flow microfluidic biochips," in *Proc. Int. Conf. Comput.-Aided Des*, 2020, pp. 1–8.

[8] K. Hu, T.-Y. Ho, and K. Chakrabarty, "Wash optimization and analysis for cross-contamination removal under physical constraints in flow-based microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst*, vol. 35, no. 4, pp. 559–572, 2015.

[9] 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 Trans. Comput.-Aided Design Integr. Circuits Syst*, vol. 37, no. 3, pp. 615–628, 2018.

[10] X. Huang, W. Guo, Z. Chen, B. Li, T.-Y. Ho, and U. Schlichtmann, "Flow-based microfluidic biochips with distributed channel storage: Synthesis, physical design, and wash optimization," *IEEE Trans. on Computers*, vol. 71, no. 2, pp. 464–478, 2022.

[11] D. Brune and S. Kim, "Predicting protein diffusion coefficients." *Proceedings of the National Academy of Sciences*, vol. 90, no. 9, pp. 3835–3839, 1993.

[12] X. Huang, Y. Pan, G. L. Zhang, B. Li, W. Guo, T.-Y. Ho, and U. Schlichtmann, "PathDriver+: Enhanced path-driven architecture design for flow-based microfluidic biochips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst*, vol. 41, no. 7, pp. 2185–2198, 2021.

[13] Y. C. Lim, A. Z. Kouzani, and W. Duan, "Lab-on-a-chip: a component view," *Microsystem Technologies*, vol. 16, no. 12, pp. 1995–2015, 2010.

Authorized licensed use limited to: Fuzhou University. Downloaded on December 17,2024 at 03:26:50 UTC from IEEE Xplore. Restrictions apply.