2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

A Capacity-Aware Wash Optimization for Contamination Removal in Programmable Microfluidic Biochip Devices

Piyali Datta[$], Arpan Chakraborty, and Rajat Kumar Pal
Department of Computer Science and Engineering, University of Calcutta, Kolkata – 700 106, West Bengal, India.
piyalidatta150888@gmail.com[$]

***Abstract*—Flow-based microfluidic biochips have emerged as a potential lab-on-chip platform for numerous biochemistry operations. Among various flow-based biochips, Programmable Microfluidic Devices (PMDs) receive much attention due to its capability of performing functionalities on a single platform with no hardware modifications. High precision control and production of correct outcomes are the urgent needs in the PMDs. While sharing the micro-channels, a fluid-flow may be contaminated by the residues stuck on the channel. Washing of the microchannels with buffer fluid is an immediate solution for safe execution of biochemistry. However, each wash fluid has a finite washing capacity and can wash only a limited number of contaminated spots. In this paper, we propose a wash optimization model that aims to remove all the contaminations while minimizing washing time and total capacity wastage. The effectiveness of the proposed approach has been evaluated considering a number of baseline methods and the previous works.**

***Index Terms*—automation, contamination, design optimization, microfluidics, washing**

I. Introduction

Microfluidic biochips (MFBs) have revolutionized the point-of-care diagnoses with rapid integration of different biochemical research functionalities [1]. In recent times, along with various other microfluidic devices such as Digital Microfluidic Biochips (DMFBs), Micro-Electrode-Dot-Array (MEDA), etc. flow-based microfluidic biochips have achieved increased popularity [1]. Among the existing architectures of flow-based platforms, Programmable Microfluidic Device (PMD) is quite popular due to enhanced flexibility with software programmed control mechanism [2].

To realize a bioprotocol, a sequence graph (called bioassay; depicted in Fig. 1(a)) modelling is first carried out while the node and edge represent respectively the operations (i.e. mixing of samples and reagents) and the operational dependencies. Then it is passed through the *Design Synthesis* to decide the placement of mixing modules, the scheduling time stamps of each sample and/or reagent fluid route, etc. [3], [4], [5], [6].

A basic flow-based microfluidic device comprises of two elastomer layers: flow layer and control layer (Fig. 1 (b)). The flow layer is connected to a fluid reservoir through a pump that generates the fluid flow whereas the control layer is connected to an external air pressure source. Figure 1(c) shows the central channel network of PMDs and a schematic view of a node in PMDs surrounded by four valves. Fluid transportation is controlled by synchronous valve opening and closing [2], [7].

High precision control and correct outcomes are the fundamental needs in any microfluidic functionality. Since the diameters of micro-channels are reduced to scale down feature sizes as well as to minimize valuable sample and reagent

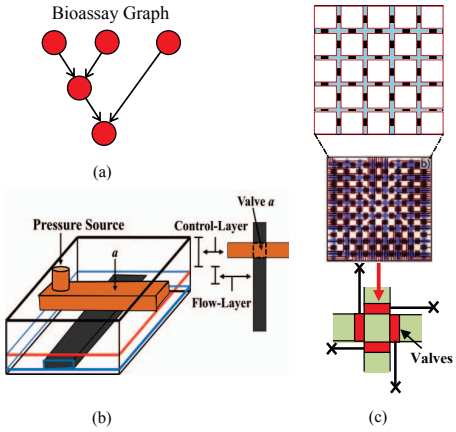


Fig. 1. (a) A bioassay graph, (b) Schematic view of valves [7], (c) Central channel networks in PMD and a crossing with four surrounded valves [7].

consumption, the higher perimeter-to-area ratio of micro-channels increases the probability of leaving residues of fluid in the microchannel walls [9]. When two contiguous fluidic flows pass through the same micro-channel, the latter has a higher chance to be contaminated by the residue from the first flow leading to erroneous assay outcomes. To maintain the effectiveness of biochip systems, a design must ensure that an execution profile of bio-protocol is free from any cross contamination. For contamination removal, washing with buffer fluid is the most common remedy that arranges a flow through the microchannels such that the residues in the flow path can be flushed. As wash fluid routing should be scheduled with the regular sample-reagent fluid routing and washing increases an overhead delaying the final outcome, wash time minimization becomes a crucial need [8], [9].

In [9], a wash optimization method for cross contamination removal has been proposed that deals with several washing targets while considering already occupied micro-channels as obstacles. A set of wash routes is produced to flush all the contaminated spots within minimum wash time while searching the paths that are implementable on a chip layout. Depending on this path-dictionary, the wash optimization problem has been formulated as a variant of hitting-set problem. However, the *washing capacity* of a buffer fluid has been considered here to be *unlimited*. The washing capacity of a 'metered' buffer fluid definitely diminishes when a larger degree of residues are washed away. Thus, the buffer fluid reaching a contaminated location is of no use if its washing capacity is

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00089

already exhausted [10]. In DMFB, wash scheme considering capacity constraints has been developed [10] while no such attempt has been taken for flow-based platforms.

Another pro-creative way for dealing with contamination problem is to arrange mutually independent paths for all the regular fluids. Since the size restriction of the chip strongly recommends re-usability of the routing resources [7], the scope to incorporate mutually independent paths is a challenging task [9]. Contamination thus becomes unavoidable on PMDs that mandates washing a channel before reusing it. The situation becomes worse, since due to channel constraints a wash fluid may have to travel through a path that is not a targeted contamination but is used by a regular fluid. The washing capacity decreases due to this unavoidable visit and in the worst case results in wash failure. Iteration in the washing process is continued till the goal is met. Obviously, iteration of washing tasks increases total assay completion time.

This paper presents a novel capacity-wastage-aware wash optimization scheme for the PMDs that ensures removal of all the contaminations while minimizing the wash time. Once the regular fluid routing is decided (i.e. the occupied micro-channels are defined by the user), the proposed strategy determines an optimized path set with the least wash completion time. At the same time, it considers the capacity constraints of the wash fluid such that washing failure does not occur in any case. Since the capacity of the wash fluid is not exhausted after washing a set of micro-channels, its re-usability is thereby ensured by reducing the overhead of rip-up and re-routing in the washing process.

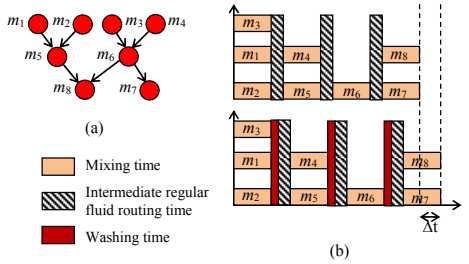
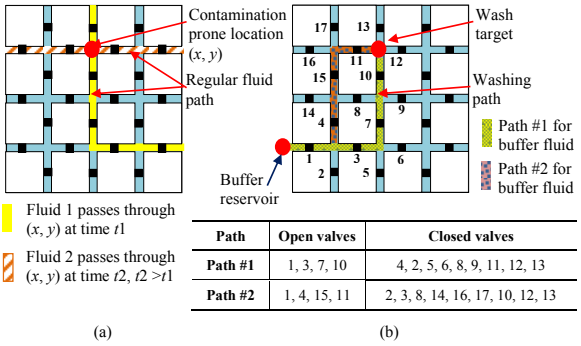


Fig. 2. (a) An example bioassay, (b) Scheduling without and with washing.

II. Problem Formulation

While integrating a wash scheme together with ongoing regular tasks (mixing/routing), two cases may arise; (i) wash-fluid route length becomes longer consuming more washing capacity and (ii) no feasible route remains there for washing within the deadline. In both the cases, contamination removal process becomes inefficient. To deal with it, we incorporate washing as a revival phase after each level of regular fluidic (mixing/routing) operations, ensuring contamination-free situation for the next level regular fluidic operations. As washing is performed in an interleaved manner with regular fluidic operations, wash time affects the assay completion time directly that makes efficient washing scheme a critical need. Figure 2(a) shows an example of bioassay for which two schedules are shown in Fig. 2(b). Observably, the schedule including washing takes some additional time Δt; hence, for an efficient washing Δt must be minimized.



Path	Open valves	Closed valves
Path #1	1, 3, 7, 10	4, 2, 5, 6, 8, 9, 11, 12, 13
Path #2	1, 4, 15, 11	2, 3, 8, 14, 16, 17, 10, 12, 13

(a) (b)

Fig. 3. (a) A regular fluid routing schedule leading to contamination, (b) Two alternative paths for washing with different micro-channels allocation.

Since washing is an interleaved task with regular fluidic operations (i.e. mixing, sample-reagent routing, and detection), each washing phase should take the previous fluidic status of the chip. We explain this in Fig. 3(a), where two different fluid paths pass through a 'cross-site' (x, y) at time instant t_1 and t_2 ($t_2 \geq t_1$), respectively, leading (x, y) to be a contamination-prone location. Hence, an intermediate wash fluid flow is to be planned through this location, and a path needs to be formed from a wash port to (x, y), accordingly. Two alternative paths (path #1 and path #2) are shown in Fig. 3(b) occupying different micro-channels. Now, if a wash fluid moves through a path that has already been used by some regular fluid, the wash fluid loses some units of its washing capacity absorbing residues from that path. Thus, fluid flow through path #1 consumes some washing capacity of wash fluid whereas no additional capacity is consumed on the path #2 as the micro-channels on path #2 have not been contaminated by any regular fluid. Thus path selection for wash fluid is a crucial issue in washing to minimize wash time as well as to reduce capacity wastage.

Based on the foregoing observations, the considered design optimization problem is as follows.

Input: A bioassay graph; Intermediate synthesis results, i.e. scheduling decisions, module placements, fluid routes; Design specifications, i.e. size of the grid, the allowable capacity of each wash fluid, maximum allowable wash time, and the assay deadline.

(a) Static Fluidic Constraint: Any two fluidic flows must maintain a *safe separation* at any time instant to avoid unintended mixing.

(b) Dynamic Fluidic Constraint: No two fluidic movements are allowed simultaneously if any *valve hazard* occurs. Valve hazard refers to the situation where a valve is required to be 'on' for one fluid while another fluid demands the valve to be in 'off' state.

(c) Valve Constraints: A node on a PMD-grid cannot input and output fluid simultaneously.

(d) Capacity Constraints: The capacity of each wash fluid visiting a wash target must be greater than or equal to the capacity required to wash that location.

Objective: To prepare a capacity-wastage-aware washing

scheme for contamination removal, by

(a) deriving the wash routes targeting the contaminations,
(b) minimizing the overall capacity consumption, and
(c) minimizing the total wash time and thereby minimizing overall assay completion time.

III. Proposed Strategy

To ensure safe bioassay operation, each *metered* wash fluid (W_f) must wash a micro-channel in between every two consecutive passages of two different sample or reagent fluids. To determine the scheduling decisions, i.e. *start* and *end* time-stamps of each W_f, we require the already generated schedule of regular fluid and the chip utilization profile (i.e. locations of executing modules and routes of each regular fluid). If P^i_{wash} denotes an *i*-th washing sub-problem, considering these two information, the targeted contaminated spots for each such P^i_{wash} can be identified.

Now, the capacity consumption for any two contaminated spots may be different. For example, if a contaminated mixer is to be washed, the capacity consumption will be higher than that of a routing node. Thus, in the next step, we retrieve the capacity consumption detail of the chip from its current status. After obtaining the capacity consumption detail of the targets for any intermediate P^i_{wash}, efficient wash fluid scheduling can be achieved. The proposed strategy first obtains a *loose route* for P^i_{wash} resulting in a traversal ordering of the contaminated spots minimizing total time of travel as well as total capacity consumption by a W_f. It is worthwhile to mention that due to *fluidic turns* in the flow path flow speed may be reduced [7]; hence, the best path for any fluidic movement is a straight line. However, obtaining a straight path for all routing may not be possible. Hence, in the next phase a *detailed route*, is found out that is aware of the *path quality* in an objective to reduce the number of turns minimizing the total capacity consumption. Subsequently, the steps have been discussed in greater detail.

A. Wash Target Identification

To keep channel utilization information, we maintain a structure $channel\_matrix$ which is a 2D array (i.e. $m \times n$ PMD grid) comprising boolean values initially holding all '0's. After completion of each regular fluidic (mixing and/or routing) sub-problem, say P^i_{reg}, the $channel\_matrix$ is updated and the corresponding positions of the used nodes in the PMD grid (during each intermediate P^i_{reg}) are made '1'. A '1' at any position signifies that the associated node is contaminated and hence is a candidate for washing. Now, if P^i_{wash} is carried out, the nodes through which washing is done will be updated as '0' referring that they are now free from contamination and further usable by any regular sample / reagent. Now, to find the wash targets we deploy another structure $next\_matrix$ holding the information for next use, i.e. the information about the nodes of the PMD grid to be used for the next regular fluidic sub-problem, i.e. P^{i+1}_{reg}. It is also an $m \times n$ 2D array, where the nodes to be used next are '1's and the remaining nodes are '0's.

Analysing the $channel\_matrix$ and $next\_matrix$ after any P^i_{reg}, the set of contaminated spots, i.e. the wash targets can be identified. The nodes that are '1' in both matrices are considered to be the target for P^i_{wash}. This can be directly obtained by a boolean AND operation between $channel\_matrix$ and $next\_matrix$ producing another 2D array $target\_matrix$ of the same dimension. Figure 4(a) explains this with an example.

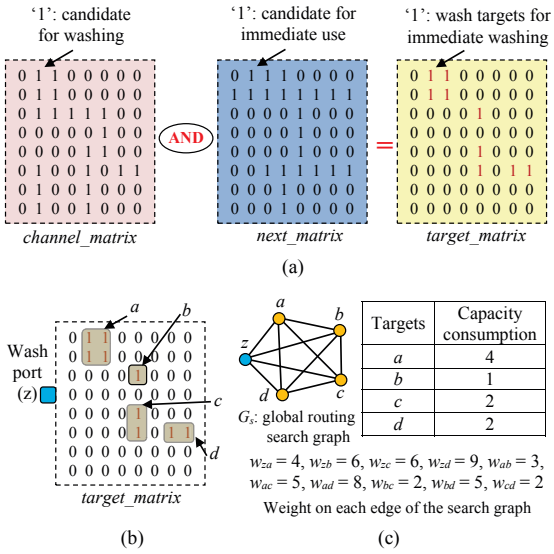


Fig. 4. (a) Generation of *target_matrix*, (b) Capacity consumption information reclaimed from *target_matrix*, (c) Search graph during global routing.

B. Capacity Consumption Information Reclaim

Once the targets are identified a set of wash paths is to be arranged for each W_f between a buffer source port outside the working chip area and the target locations. Since the capacity of each W_f is finite, the shortest path covering all the targets would not suffice for wash optimization. To deal with this scenario, from $target\_matrix$, on-chip *capacity consumption* information must be reclaimed (as shown in Fig. 4(b)).

As mentioned earlier, all the contaminated spots do not demand identical capacity consumption. As washing off a contaminated mixer demands a high degree of capacity than that of a single contaminated node, instead of referring a single node of a contaminated mixer as individual wash target with capacity consumption one unit, we consider all the *m* nodes of the mixer as a single chunk of contaminated location with capacity demand *m* units. Now, if a W_f with existing wash capacity less than *m* reaches a target of capacity consumption *m*, washing failure occurs. In such scenarios, a new W_f has to be traversed again to that location for accomplishing its task. Clearly, it increases the wash time and washing overhead delaying the assay completion time. Figure 5 exemplifies the fact. Hence, before setting up the routing of each W_f, the information regarding the capacity consumption of each node is necessary. Identifying the contiguous wash targets from $target\_matrix$, it can be directly reclaimed.

C. Capacity-Wastage-aware Global Route of Buffer Fluid

Let us assume that each wash sub-problem P^i_{wash} can deploy multiple W_f to deal with n targets. A search graph (as

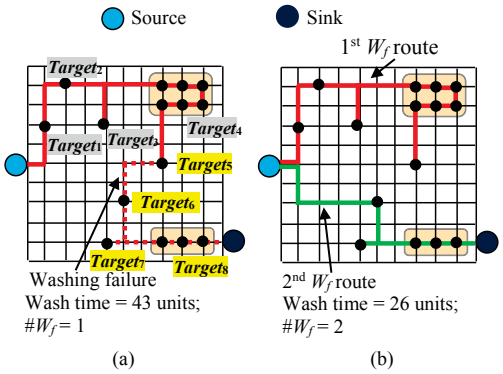


Fig. 5. Washing scheme with on-demand wash fluid injection: (a) Washing failure starting from fifth contamination, (b) Success with wash capacity.

shown in Fig 4(c)) can be visualized by taking the source port and all the wash targets belonging to P^i_{wash}. It is a complete graph of $n+1$ vertices including a source port and n targets where the connectivities are represented by edges. A weight on each edge is assigned by following the Manhattan distance between the vertices to represent the travel cost (i.e. travel time) of W_f between the nodes. The capacity consumption is also associated with each vertex (other than the source port).

Problem 1: *Capacity-Wastage-aware Route Order (CWRO).*

Instance: A search graph G_s with vertex set $V = \{u_0, u_1, u_2, ..., u_n\}$, and edge set E; u_0 is the source port; $V \setminus \{u_0\}$ represents the wash targets; Each target needs a capacity consumption demand d_i, where $d_i \neq d_j$ and $i, j = 1, 2, ..., n$; Each wash fluid W_f has identical washing capacity C; Each edge (u_i, u_j), $i \neq j$, is associated with an edge cost w_{ij}; A positive integer K.

Question: Is there a set of paths in G_s for a number of W_f that start from u_0 and visit each target in $V \setminus \{u_0\}$ exactly once such that the total capacity consumption in each route does not exceed the total capacity, C and the total traversal cost is less than K?

Problem 1 is equivalent to the well-known Capacitated Vehicle Routing Problem (CVRP) [11]. To optimally solve *CWRO* for minimum capacity consumption and minimum traversal cost, here we present a Mixed Integer Linear Programming (MILP) formulation. Let the decision variable is x_{ij}. If x_{ij} = 1, then the edge connecting vertices u_i and u_j is included in a route; else x_{ij} = 0. Followings are the objective functions and constraints for Problem 1.

$$\min \sum_{u_i, u_j \in E} w_{ij} x_{ij} \tag{1}$$

subject to

$$\forall u_i \in V, \sum_{u_j \in V, u_i \neq u_j} x_{ij} = 1 \tag{2}$$

$$\forall u_j \in V, \sum_{u_i \in V, u_i \neq u_j} x_{ij} = 1 \tag{3}$$

$$\forall u_i, u_j \in V \text{ and } u_i \neq u_j, \text{if } x_{ij} = 1, \text{ then } p_j = p_i + d_i \tag{4}$$

$$\forall u_i \in V, p_i \geq d_i \text{ and } p_i \leq C \tag{5}$$

$$x_{ij} \geq 0 \tag{6}$$

Here, (1) denotes the objective function minimizing total traversal cost; (2) states that from each vertex u_i exactly a single vertex can be visited; (3) states that if a vertex u_j is visited then it must come from exactly a single vertex u_i; (4) is a subtour elimination constraint in the form of an indicator constraint. It states that if p_i and p_j denote the cumulative demands of vertices u_i and u_j, respectively, then going from u_i to u_j it must be satisfied, where d_i is the demand at u_i; (5) states that at any time the cumulative demand must not exceed the total capacity C; and (6) is implied.

D. Turning-Aware Detailed Routing Phase

The MILP in the previous section produces a set of loose routes that may contain some locations that are not immediate wash targets in P^i_{wash} but may have contamination. If a W_f passes through these locations certain amount of capacity is consumed affecting the wash routing. In the worst case, it may lose its capacity fully leading to washing failure. Hence at the detailed routing phase, this issue must be taken care of. Another imperative issue to be dealt with is on-chip fluidic turns. As mentioned earlier, to maintain path quality the number of fluidic turns must be reduced.

In the detailed routing phase, each produced route from *CWRO* is observed to obtain a route dictionary. It actually holds multiple feasible routes corresponding to each loose route. The objective behind this is to avoid additional capacity consuming nodes on wash path. Also, since fluidic turns are unavoidable in a route, a number of 'L' shaped sub-paths can be incorporated in the original loose route. Though 'L' shaping idea does not avoid fluidic turns fully, it helps to reduce their number and improves the path quality.

Figure 6(a) depicts a set of 'L' sub-paths reorganizing the loose routes. To reduce additional capacity consumption on a path, a set of alternate paths between each target pair with a different number of fluidic turns is considered. From this, the minimum capacity consumable path for each wash fluid is

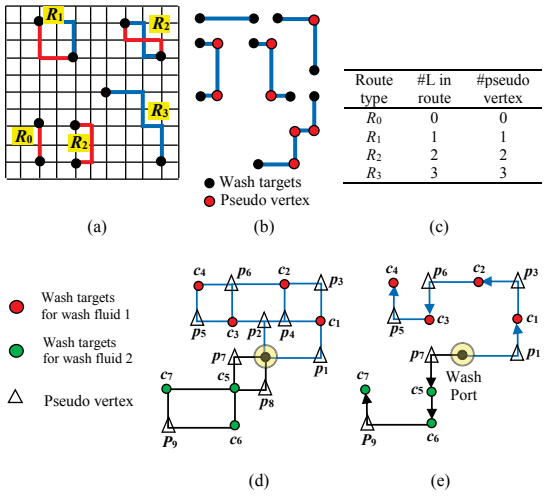


Fig. 6. (a) Various 'L'-shaped routes, (b)-(c) Introduced pseudo vertices for different 'L'-shaped routes, (d) Connections through one-L routes, (e) Paths of two wash fluids with minimum additional capacity consumption.

established. Depending on the fluidic turns, i.e. zero or more turns, an 'L' shaping can be further categorized into *zero-L*, *one-L*, *two-L*, ... paths, occupying different micro-channels. However, as the micro-channels are either horizontal or vertical, on each such path we have to add a set of pseudo vertices to implement the rectilinear paths with different number of 'L's. These pseudo vertices are basically junction nodes on a PMD chip as exemplified in Fig. 6(b). Figure 6(c) depicts the number of pseudo vertices for each 'L'-shaped path.

Let us denote all the wash targets for a P^i_{wash} as vertex set V_1 and the set of necessary pseudo vertices as vertex set V_2. A weighted graph $G_d(V, E)$ at the detailed routing phase can be immediately obtained, where $V = V_1 \cup V_2$ and E is the set of connections between any two vertices during the 'L' shaping process. Observing the *channel_matrix*, a weight w on each edge can be assigned as follows:

w = Number of micro-channel units through an edge that is a candidate for washing.

Once we obtain the weighted graph G_d associated with each P^i_{wash}, we find the minimum cost path for all the wash fluids covering the contaminated spots. Floyd-Warshal's shortest path algorithm can be tailored to find optimized cost path between each wash target pair that collectively form the wash fluid paths. Figure 6(d) shows an example of G_d taking only *one-L* paths while Fig. 6(e) shows the final paths for W_f.

IV. Experimental Results

To evaluate the effectiveness of the proposed method, here we compare it with three baseline algorithms and a number of related works in existing literature.

Baseline Approaches:

1) *Horizontal Scan* (*H-Scan*): After identifying the wash targets, these are sorted in non-decreasing order of their x-coordinates. Depending on the location of the wash port (i.e., whether it is nearer to the leftmost or the rightmost periphery), wash fluid (W_f) is scheduled to traverse sequentially starting from the left most or the right most contaminated location.
2) *Vertical Scan* (*V-Scan*): Unlike H-Scan, it sorts the wash targets according to non-decreasing y-coordinates and then are traversed starting from the lowest (or highest) y-coordinate depending on the location of the wash port.
3) *Nearest Target First* (*NTF*): A W_f visits each target according to the nearest location first rule.

In the three baseline approaches, the washing capacity is not considered apriori. If a W_f already loses capacity on its way, a new wash fluid has to be injected on-demand and is routed to the current wash target.

We have implemented the algorithms in Python and simulated the MILP using CPLEX for Python on a 3.20 GHz Core i5 machine with 8 GB RAM. To evaluate the proposed wash strategy, we have considered five chip dimensions (10×10, 20×20, 30×30, 40×40, and 50×50) for PMDs and two test cases for each chip capacity. Half-perimeter length has been considered to be the maximum inter-target distance for each case. Taking each test case, we have randomly generated ten instances for each specification. Table I shows the averaged quantitative values of each metric (i.e. wash time, number of deployed wash fluids flow, and CPU time) for the proposed approach along with the aforementioned baseline schemes.

From Table I, it is observed that the proposed approach outperforms the baseline approaches regarding wash time and the number of requisite wash fluids. Here, wash time is measured considering the time required to traverse in between two consecutive micro-channel junction as *one unit*. It completes contamination removal from all the wash targets taking 75%, 71.8%, and 43.2% less time (on an average) than *H-Scan*, *V-Scan*, and *NTF*, respectively. Figure 7(a) shows the comparison of wash time between our approach and the baseline algorithms for all the test cases. The number of wash fluids required is another crucial metric to evaluate washing scheme. The requisite number of wash fluids flow in our approach is 42.1%, 41.4%, and 47.3% less (on an average) than the aforementioned baseline approaches, respectively. Figure 7(b) depicts the variation of requisite wash time with varying number of on-chip wash targets for our approach as well as the baseline approaches. The horizontal axis represents the ratio of wash targets to the total number of micro-channels on a chip (in percentage), whereas requisite wash time is depicted along vertical axis. Noticeably, rate of increment of wash time with the number of wash target is minimum in our approach.

In [7], a routability driven flow routing algorithm for a PMD has been introduced that minimizes assay completion time while satisfying fluidic and valve constraints. The approach focuses into regular fluid routing without considering contamination problem. As our approach to washing has been reduced to wash fluid routing, we compare the results with [7]. In [7], two baseline approaches have been introduced (sequential

TABLE I
COMPARATIVE STUDY OF WASH TIME, #WASH FLUIDS FLOW, AND CPU TIME AMONG THE BASELINE METHODS AND OURS

Chip Size	#Wash targets	Maximum inter-target distance	H-Scan			V-Scan			Nearest Target First			Ours		
			Wash Time (unit)	#Wash Fluids	CPU Time (s)	Wash Time (unit)	#Wash Fluids	CPU Time (s)	Wash Time (unit)	#Wash Fluids	CPU Time (s)	Wash Time (unit)	#Wash Fluids	CPU Time (min)
10×10	10	20	84	2	1.043	96	2	1.123	50	3	1.067	32	2	<1.0
	20		156	4	2.151	119	4	2.043	70	6	1.172	56	4	<1.0
20×20	10	40	220	4	1.030	208	3	1.017	128	4	1.112	84	3	<1.0
	20		560	7	2.053	500	7	2.150	213	7	1.090	140	4	<2.0
30×30	20	60	780	8	2.320	790	8	2.048	343	10	2.118	170	5	<2.5
	30		1158	11	2.674	1068	11	2.199	467	12	2.157	277	7	<3.5
40×40	20	80	992	9	2.835	980	9	2.899	387	10	2.165	218	4	<4.0
	30		1722	14	3.102	1342	13	3.136	665	13	3.271	371	7	<4.5
50×50	20	100	1149	10	3.172	1040	10	3.403	541	10	3.100	292	5	<5.0
	30		1870	15	3.158	1660	15	3.167	998	16	3.199	555	7	<6.0

TABLE II
COMPARATIVE STUDY OF WASH TIME AMONG THE PREVIOUS WORKS AND OURS

Comparison of wash completion times considering [7]						Comparison of completion times with [9]					
Test cases [7]	Sequential [7]	MIS [7]	Flow-routing [7]	Ours w/o-capacity	Ours w-capacity	Test cases [9]	M1 [9]	M2 [9]	M3 [9]	M4 [9]	Ours
AI	82.1	45	26.4	30.2	42.2	BI	15.6	10.0	8.0	7.0	2.6
AII	162.1	62.2	36.6	40.0	46.3						
AIII	537.9	145.8	58.4	59.6	146.2	BII	15.6	11.4	10.0	8.4	4.2
AIV	1150.5	231.8	83.0	90.3	174.3						
AV	1973.3	302.4	106.0	194.0	360.2	BIII	15.6	14	11.4	8.3	5.6
AVI	3006.5	423.2	131.2	372.2	880.3						

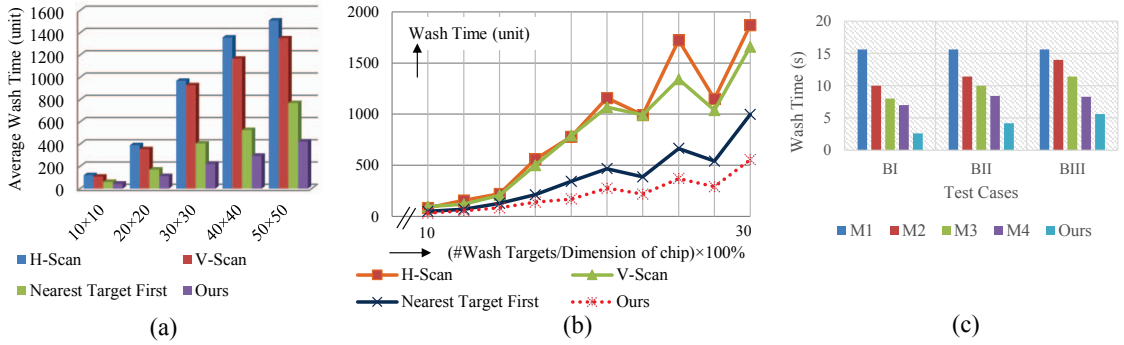


Fig. 7. (a) Comparative study of average wash times among the baseline approaches and ours, (b) Variation of wash times with varying number of wash targets, (c) Comparative study of wash times among previous works and ours.

and Maximum Independent Set (MIS)), and six test cases have been defined. We have assumed these same specifications while generating the test cases to apply our approach (without and with capacity consideration). The first six columns of Table II depict the comparative study. Observably, our approach takes less time than the sequential one, irrespective of w-capacity or w/o-capacity consideration. However, the proposed method w/o-capacity consideration outperforms MIS while w-capacity takes more time than MIS. In addition, though the flow routing algorithm takes less time than the proposed one, it does not consider contamination removal. In [9], a wash optimization method for cross-contamination removal in flow-based microfluidic biochips has been developed. We have considered all the four methods and the same test case specifications stated in [9]. The last six columns of Table II and Fig. 7(c) show the comparative study. In all the cases, the proposed method minimizes wash time, reduces capacity wastages, and produces a set of *good* wash paths.

V. CONCLUSION

Since cross contamination is unavoidable in PMDs and correct outcomes are mandatory, this paper presents a wash optimization strategy which not only minimizes the overall wash time but effectively reduces the number of required buffer fluids and capacity wastage for each of them. Another important issue of concern is that during a fluidic flow an increasing number of on-chip fluidic turns reduces the flow speed and hence it may incorporate synchronization hazards. Thus, to improve path quality, the proposed strategy minimizes the number of flow turns by deploying 'L' shaped paths. A comprehensive comparative study of the previous works and that of ours have been conducted. However, a more complex scenario would be performing a complete design synthesis in PMDs where the regular and wash fluid movements are jointly considered. This is left as future work.

REFERENCES

[1] T.Y. Ho, K. Chakrabarty, and P. Pop, "Digital Microfluidic Biochips: Recent Research and Emerging Challenges," IEEE CODES + ISSS, 2011.

[2] L.M Fidalgo, and S.J. Maerkl, "A software-programmable microfluidic device for automated biology," Lab on Chip, vol. 9. pp. 1612–1619, 2011.

[3] W.H. Minhass, P. Pop, J. Madsen, and F.S. Blaga, "Architectural synthesis of flow-based microfluidic large-scale integration biochips," in Proc. Compilers, architectures and synthesis for embedded systems, pp. 181–190, 2012.

[4] H. Yao, Q. Wang, Y. Ru, Y. Cai, and T.Y. Ho, "Integrated Flow-Control Codesign Methodology for Flow-Based Microfluidic Biochips," IEEE Design and Test, vol. 32, pp. 60–68, 2015.

[5] C. Liu, B. Li, H. Yao, P. Pop, T.Y. Ho, and U. Schlichtmann, "Transport or Store?: Synthesizing Flow-based Microfluidic Biochips using Distributed Channel Storage," in Proc. Design Automation Conference (DAC), p. 49, 2017.

[6] K.Hu, T.A. Dinh, T.Y. Ho, and K. Chakrabarty, "Control-layer routing and control-pin minimization for flow-based microfluidic biochips," IEEE Trans. Comput.-Aided Design of Integr. Circuits and Syst., vol. 36, pp. 55–68, 2017.

[7] Y.S. Su, T.Y. Ho, and D.T. Lee, "A routability-driven flow routing algorithm for programmable microfluidic devices," Design Automation Conference (ASP-DAC), pp. 605–610, 2016.

[8] T.W. Huang, C.H. Lin, and T.Y. Ho, "A contamination aware droplet routing algorithm for digital microfluidic biochips," in Proc. IEEE/ACM ICCAD, pp. 151–156, 2009.

[9] 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 of Integr. Circuits and Syst., vol. 35, pp. 559–572, 2016.

[10] Q. Wang, Y. Shen, H. Yao, T.Y. Ho, and Y. Cai, "Practical functional and washing droplet routing for cross-contamination avoidance in digital microfluidic biochips," in Proc. Design Automation Conference (DAC), ACM/EDAC/IEEE, ICCAD, pp. 1–6, 2014.

[11] P. Toth, and D. Vigo, "Vehicle routing: problems, methods, and applications," SIAM, 2014.