Flow-based Augmented Droplet Routing Algorithm for MEDA-Based DMFB

Emuun Purevdagva Masayuki Shimoda Satoshi Tayu Atsushi Takahashi
Department of Information and Communications Engineering, School of Engineering
Institute of Science Tokyo, Japan
emuun@eda.ict.eng.isct.ac.jp, {shimoda, tayu, atsushi}@ict.eng.isct.ac.jp

Abstract— Micro Electrode Dot Array-based Digital Microfluidic Biochip (MEDA) is a promising platform in biological and medical applications, offering high flexibility in droplet-based operations such as disease diagnosis, DNA analysis, and PCR testing. To fully exploit its capabilities, fast and accurate droplet routing is essential. In this work, we formulate droplet transportation in MEDA as a flow network problem and propose a high-speed algorithm that efficiently finds a minimum-time routing with low computation cost. The algorithm first determines the shortest routing time using maximum flow search, and then refines the solution using minimum-cost flow optimization to reduce redundant movements. The proposed algorithm enables efficient handling of large-scale problems, contributing to improved experimental efficiency of MEDA biochips.

I. Introduction

Digital Micro Fluidic Biochip (DMFB) is a biochip used in biochemistry and medicine, and is utilized in various biochemical applications such as disease diagnosis, genetic analysis, PCR testing [1, 2]. DMFB consists of electrodes and a hydrophobic plane, and manipulates droplets by the principle of electrowetting-on-dielectric (EWOD). This enables operations such as droplet splitting, mixing, and parallel processing of multiple droplets [3]. However, conventional DMFBs have large limitations in droplet manipulation, and there have been issues of practicality.

Micro Electrode Dot Array-based biochip (MEDA biochip) is a biochip that has improved droplet manipulation by miniaturizing an electrode cell to about one-tenth the size of conventional ones [4, 5]. The MEDA biochip has microelectrodes of several μm in diameter arranged in a square grid. By individually controlling the on/off states of these microelectrodes, it is possible to deform droplets, move them diagonally, and perform cell-by-cell routing operations. Improved droplet manipulation enables flexible operations that are not feasible with conventional biochips. Such enhanced control is expected to increase experimental efficiency by enabling faster droplet routing and scaling up the experiment size.

This paper addresses a droplet routing problem in a MEDA biochip, where droplets are required to move from source cells to target cells as quickly as possible while passing through as few intermediate cells as possible. A droplet is assumed to consist of multiple unit droplets. These unit droplets are initially placed in the source cells and must be moved to the target cells without overlapping with each other at any time. The routing problem discussed in this paper is to find a minimum-time routing of unit droplets from source cells to target cells, using only available cells. Some cells in the MEDA biochip are unavailable for routing due to contamination from other droplets or hardware faults.

In recent years, various methods have been proposed to find better droplet routing for MEDA biochips [6, 7, 8, 9, 10, 11]. Methods for DMFB treat a droplet as a single large entity and search for a path for it to move as a whole. However, these methods are difficult to apply to MEDA biochips, where droplets can be divided and moved separately, and thus may fail to obtain an optimal routing. On the other hand, methods designed for MEDA biochips can find an optimal routing, but their large computation times make it difficult to apply to large-scale instances.

In this paper, we propose a method for MEDA biochips that finds a minimum-time droplet routing in a short computation time. In the proposed method, droplet routing is represented by a flow in a 3D flow graph. The (x,y)-plane of the 3D flow graph corresponds to a microelectrode array and the z-axis of it corresponds to a time step allowed in the routing. In the 3D flow graph, each unit flow from its primary source to its primary sink corresponds to the routing path of a unit droplet. In a case that a feasible droplet routing exists, a feasible routing is obtained if the number of layers of the 3D flow graph is set large enough. The proposed method initially sets the number of layers small, then iteratively increases it by one until the paths of all unit droplets are completed in the corresponding time steps.

II. Droplet Routing Problem

In biomedical tasks using a MEDA biochip, multiple droplets are routed simultaneously across its microelectrode cells. In a routing operation, each droplet placed at a source cell is required to move to a designated target cell. To improve task efficiency, droplet routing aims to minimize both the required time and the number of cells involved in the movement.

In a MEDA biochip, the microelectrode cells are arranged in a matrix of width w and height h. The cell map $M = \{c_{x,y} \mid 1 \leq x \leq w, 1 \leq y \leq h\}$ represents a microelectrode cell array where $c_{x,y}$ is the cell at coordinates (x, y). **Figure 1** shows an example of a MEDA cell map.

On a MEDA biochip, droplets that occupy adjacent cells physically form a single coalesced droplet. The required time

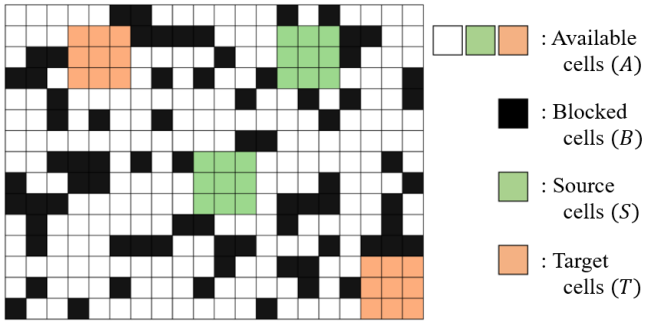


Fig. 1 : An example of a MEDA cell map.

to move a coalesced droplet to adjacent cells depends on the size and shape of the droplet, and a droplet split operation takes even longer [12, 13, 8, 11]. However, in this work, droplets are modeled by virtual unit droplets, each occupying a single cell without coalescence. Also, it is assumed that the time step for routing is set large enough to complete any single move operation in which a unit droplet is moved to an adjacent cell.

In cell map M, cells are categorized into a set of available cells $A(\subseteq M)$ and a set of blocked cells $B(= M \setminus A)$. In cell map M, some cells may become unusable due to contamination, damage, or other factors. Only the available cells in A are used for droplet routing. Let $D = \{d(1), d(2), \ldots, d(n)\}$ be the set of unit droplets to be routed. At the initial state of droplet routing, the unit droplets are placed in the source cells and are required to reach target cells. The set of source cells is given by $S = \{s(1), s(2), \ldots, s(n)\}$ $(\subset A)$ where unit droplet $d(i)$ is placed at $s(i)$ $(1 \le i \le n)$. The set of target cells is given by T $(\subset A)$ where $S \cap T = \emptyset$ and $|T| = |S|$.

A droplet routing request is specified by the cell map $M(A, S, T)$, where A, S, and T denote the sets of available cells, source cells, and target cells, respectively. In unit droplet routing, each droplet either stays in the same cell or moves to an adjacent cell in each step. The allowed movement directions are vertical, horizontal, and diagonal.

When a unit droplet moves from cell $c_{x,y}$ to cell $c_{x',y'}$ obliquely adjacent to $c_{x,y}$, that is, $x' = x \pm 1$ and $y' = y \pm 1$, then the droplet also passes cells $c_{x',y}$ and $c_{x,y'}$. An oblique movement of a unit droplet is allowed only if all these cells are available cells. A unit droplet movement from cell $c_{x,y}(\in A)$ to cell $c_{x',y'}(\in A)$ is allowed only if $|x - x'| \le 1$, $|y - y'| \le 1$, and $c_{x',y}, c_{x,y'} \in A$.

Let $P = \{p(1), p(2), \ldots, p(n)\}$ be the set of routing paths of unit droplets where $p(i)$ is the routing path of unit droplet $d(i)$ $(1 \le i \le n)$. At each step, each unit droplet exclusively occupies one cell. Let $c_t(i)(\in A)$ be the cell occupied by unit droplet $d(i)$ at time step t. The routing path of droplet $d(i)$ is represented by $p(i) = (c_0(i), c_1(i), \ldots)$ where $c_0(i) = s(i)$. In a feasible routing of droplets, distinct droplets $d(i)$ and $d(j)$ $(i \ne j)$ occupy different cells at any time step t, that is, $c_t(i) \ne c_t(j)$.

A droplet routing path P is evaluated by the time step $t(P) = \min\{t \mid c_t(i) \in T, \forall c_t(i) \in p(i) \in P\}$, which is the earliest time step at which all unit droplets are at the target cells. All unit droplets starting from the source cells S follow paths in P and arrive at cells in T at time step $t(P)$. The number of cells used in the routing P, which is given by $c(P) = |\{c_t(i) \mid c_t(i) \in p(i) \in P\}|$, is also used to evaluate P, serving as the secondary objective in this paper.

The droplet routing problem in this paper is defined as follows:

Droplet Routing Problem

Input: Cell map $M(A, S, T)$.

Output: Droplet routing paths $P = \{p(1), p(2), \ldots, p(n)\}$.

Objective:

- Minimize $t(P) = \min\{t \mid c_t(i) \in T, \forall c_t(i) \in p(i) \in P\}$,
- and then minimize $c(P) = |\{c_t(i) \mid c_t(i) \in p(i) \in P\}|$.

Constraints:

- $c_0(i) = s(i), \forall i (1 \le i \le n)$
- $c_t(i) \in A, \forall i (1 \le i \le n), \forall t (0 \le t)$
- $|x - x'| \le 1$, $|y - y'| \le 1$, and $c_{x',y}$, $c_{x,y'} \in A$ where $c_{x,y} = c_t(i), c_{x',y'} = c_{t+1}(i), \forall i (1 \le i \le n), \forall t (0 \le t)$.
- $c_t(i) \ne c_t(j), \forall i, j (i \ne j), \forall t (0 \le t)$

III. 3D Flow Graph $G(M, t_{\max})$

The 3D flow graph to find a unit droplet routing is defined in terms of cell-map $M(A, S, T)$ and time step $t_{\max}$ as $G(M, t_{\max}) = (V, E)$, where $V = \{v_S, v_T\} \cup V_{\mathrm{i}} \cup V_{\mathrm{o}}$ and $E = E_S \cup E_T \cup E_{\mathrm{io}} \cup E_{\mathrm{m}}$ [10].

In flow graph $G(M, t_{\max})$, two vertices v_S and v_T are the primary source and the primary sink, respectively, and an inflow vertex in V_{i} and an outflow vertex in V_{o} are defined for each available cell in A and time step t. That is,

$$\begin{aligned} V_{\mathrm{i}} &= \left\{ v^{\mathrm{i}}_{x,y,t} \mid c_{x,y} \in A, 0 \le t \le t_{\max} \right\}, \\ V_{\mathrm{o}} &= \left\{ v^{\mathrm{o}}_{x,y,t} \mid c_{x,y} \in A, 0 \le t \le t_{\max} \right\}. \end{aligned} \tag{1}$$

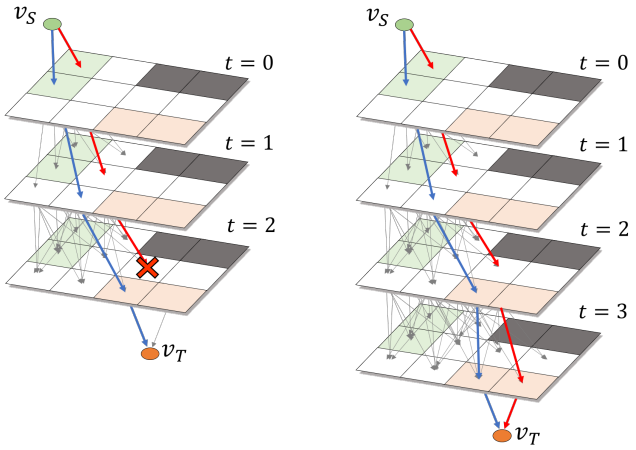
The primary source v_S and the primary sink v_T are connected to the vertices corresponding to the source and target cells, respectively.

$$E_S = \left\{ \left(v_S, v^{\mathrm{i}}_{x,y,0} \right) \mid c_{x,y} \in S \right\}$$

is the set of directed edges that connect v_S to the inflow vertex of a source cell in S at time step 0, while

$$E_T = \left\{ \left(v^{\mathrm{o}}_{x,y,t_{\max}}, v_T \right) \mid c_{x,y} \in T \right\}$$

is the set of directed edges that connect the outflow vertex of a target cell in T at time step $t_{\max}$ to v_T. The inflow vertex



(a) $G(M, 2)$, $|S| > |F| = 1$. (b) $G(M, 3)$, $|S| = |F| = 2$

Fig. 2 : Maximum flow of 3D flow graph $G(M, t_{\max})$.

$v^{\mathrm{i}}_{x,y,t}$ and the outflow vertex $v^{\mathrm{o}}_{x,y,t}$ of cell $c_{x,y}$ at time step t are connected by an internal directed edge in E_{io}. That is,

$$E_{\mathrm{io}} = \left\{ \left(v^{\mathrm{i}}_{x,y,t}, v^{\mathrm{o}}_{x,y,t} \right) \mid v^{\mathrm{i}}_{x,y,t} \in V_{\mathrm{i}}, v^{\mathrm{o}}_{x,y,t} \in V_{\mathrm{o}} \right\}. \tag{2}$$

E_{m} is the set of directed edges that correspond to feasible droplet movements in one time step. A feasible unit droplet movement from cell $c_{x,y} \in A$ to cell $c_{x',y'} \in A$ at time step t is represented by $\left(v^{\mathrm{o}}_{x,y,t}, v^{\mathrm{i}}_{x',y',t+1} \right) \in E_{\mathrm{m}}$ where $|x - x'| \leq 1$, $|y - y'| \leq 1$, $0 \leq t \leq t_{\max} - 1$, and $c_{x,y'}, c_{x',y} \in A$.

The capacity of each edge in E is set to 1. A cell is occupied by at most one unit droplet at each time step, which is guaranteed by setting the edge capacity of each internal edge to one.

Let F be a flow of $G(M, t_{\max})$ from v_S to v_T that consists of unit flows. Let

$$(v_S, v^{\mathrm{i}}_0(i), v^{\mathrm{o}}_0(i), v^{\mathrm{i}}_1(i), v^{\mathrm{o}}_1(i), \ldots, v^{\mathrm{i}}_{t_{\max}}(i), v^{\mathrm{o}}_{t_{\max}}(i), v_T)$$

be the path from v_S to v_T in $G(M, t_{\max})$ where unit flow $f(i)$ in F passes. The path of $f(i)$ corresponds to the routing path $p(i) = (c_0(i), c_1(i), \ldots, c_{t_{\max}}(i))$ of unit droplet $d(i)$ where inflow vertex $v^{\mathrm{i}}_0(i)$ and outflow vertex $v^{\mathrm{o}}_0(i)$ correspond to source cell $s(i)$ where unit droplet $d(i)$ is placed. Note that vertices $v^{\mathrm{i}}_t(i)$ and $v^{\mathrm{o}}_t(i)$ in the path are an inflow vertex and an outflow vertex, respectively, that correspond to $c_{x,y} \in A$ for some x and y for any t. Unit flows in F are internally disjoint, and they correspond to a feasible routing of unit droplets.

The amount of flow $|F|$ is at most $|S|$ since the set of internal edges that correspond to the set of source cells S forms a cut of size $|S|$. If the amount $|F|$ of any flow F is less than $|S|$, then there is no routing of unit droplets D within time step $t_{\max}$ (Fig. 2(a)). If the amount $|F|$ is equal to $|S|$, then F corresponds to a routing of unit droplets D within time step $t_{\max}$ (Fig. 2(b)).

Algorithm 1 Flow-based Augmented Droplet Routing (FADR)

Require: Cell map $M(A, S, T)$
1: $s \leftarrow s(1)(\in S)$
2: $t \leftarrow \min_{c \in T} \mathrm{Distance}_M(s, c)$
3: **while** True **do**
4: $F \leftarrow \mathrm{Maxflow}(G(M, t))$
5: **if** $|F| = |S|$ **then**
6: **break**
7: **end if**
8: $t \leftarrow t + 1$
9: **end while**
10: $F \leftarrow \mathrm{MincostMaxflow}(G(M, t))$
11: $P \leftarrow \mathrm{UnitDropletPaths}(F)$

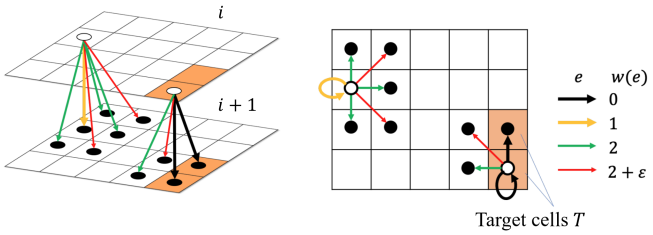
IV. Proposed Method: Flow-based Augmented Droplet Routing (FADR)

In this paper, we formulate the droplet routing problem as a flow network problem and propose the Flow-based Augmented Droplet Routing (FADR) method to find the path with the minimum routing time in a short computation time. This method repeatedly finds a maximum flow of 3D flow graphs. In the 3D flow graphs except the final one, the amount of the maximum flows is less than the number of unit droplets to be routed. This reduces the computation time per search, enabling faster discovery of a minimum-time routing path.

The pseudo-code of the proposed FADR is given in Algorithm 1. Given a cell map $M(A, S, T)$ as input, FADR first determines the number of layers t. Then, the 3D flow graph $G(M, t)$ is constructed, and a max-flow search $\mathrm{MaxFlow}(G(M, t))$ is performed on it. If the amount of flow F found is less than $|S|$, then t is incremented by one, and the process is repeated until a flow whose amount is equal to $|S|$ is found. Then, a minimum-cost max-flow search $\mathrm{MincostMaxflow}(G(M, t))$ is performed to obtain an optimized flow F. Finally, the flow F is converted into a set of unit droplet routing paths P using the function $\mathrm{UnitDropletPaths}(F)$.

The initial number of layers t should not be larger than the minimum time steps required in the routing. Otherwise, FADR fails to obtain a routing with the minimum time steps. If the number of layers t of a 3D flow graph is set to the maximum of the distances from sources to their nearest target cell, then it is always less than or equal to the required time steps and helps minimize the number of iterations. However, obtaining that value has a high computational cost. To simplify, the algorithm selects one source cell $s \in S$ arbitrarily and sets the initial t to the distance from s to its nearest target cell $c \in T$, defined as $\min_{c \in T} \mathrm{Distance}_M(s, c)$. In this work, the distance is obtained by using Dijkstra's shortest-path algorithm [14].

In our FADR, $\mathrm{MaxFlow}(G(M, t))$ obtains a maximum flow from v_S to v_T in $G(M, t)$. In our implementation, the Ford-Fulkerson method [15], where a unit-capacity augmenting path is repeatedly found by Breadth-First Search (BFS), is used.



(a) Weight of edges in E_{m}. (b) Projection onto 2D map.

Fig. 3 : Edge weight assignment.

Its computation time is roughly proportional to the number of unit flows found. Therefore, infeasible cases, as illustrated in Fig. 2(a), finish searches faster than feasible ones (Fig. 2(b)). Our FADR searches the infeasible region to reduce the total computation time.

In FADR, in order to reduce the number of cells used during the unit droplet routing, a minimum cost flow is searched by assigning the weight to each edge in the 3D flow graph $G(M,t)$ as assigned in [10].

For an edge not in E_{m}, the weight is set to 0. For an edge in E_{m}, the weight is assigned to encourage droplets to reach target cells earlier and to minimize the number of cells passed. The weight assigned to an edge in E_{m} is defined as follows:

$$W(v^{\mathrm{o}}_{x,y,t}, v^{\mathrm{i}}_{x',y',t+1}) = \begin{cases} 0 & \text{if } c_{x,y}, c_{x',y'} \in T, \\ 1 & \text{if } c_{x,y} \notin T \vee c_{x',y'} \notin T,\ x = x',\ y = y', \\ 2 & \text{if } c_{x,y} \notin T \vee c_{x',y'} \notin T,\ x = x' \oplus y = y', \\ 2+\epsilon & \text{otherwise,} \end{cases} \tag{3}$$

where $\oplus$ denotes the exclusive-OR, and ϵ is set to 0.001 in experiments. The edge weights assigned are illustrated in **Figure 3**.

The weight is set to 0 if the edge corresponds to a movement among target cells and set to positive otherwise. The earlier the droplet reaches the target cells, the smaller the total weight corresponding to the movement is expected to be. For an edge that corresponds to a movement that a droplet stays at a cell, a horizontal or vertical movement, and an oblique movement, the weight is set to 1, 2, and $2+\epsilon$, respectively. These weights prevent droplets from passing unnecessary cells. A stay is preferred to moving back and forth. A single oblique movement rather than a vertical movement followed by a horizontal movement is preferred. While one directional movement instead of two oblique movements is preferred.

Let F be a minimum cost flow of $G(M,t)$ where $|F| = |S|$. Let P be the routing path corresponding to F. P is a feasible routing path of unit droplets D. The time step $t(P)$, the number of time steps required in the routing corresponding to P, is expected to be small due to the weights assigned to edges, but may not be the minimum to complete the routing if the number of layers t is set larger. Also, the number of cells $c(P)$ used in the routing corresponding to P is not minimum in general.

In our FADR, MincostMaxflow$(G(M,t))$ finds a minimum cost flow. In our implementation, the maximum flow found is refined by the Bellman-Ford algorithm to minimize total edge weights. Although the routing time $t(P)$ remains unchanged in FADR, this step reduces redundant movement and the number of used cells $c(P)$.

V. Experiment

The performance of our proposed droplet routing FADR is evaluated by experiments. The experiments were conducted on a computer equipped with an Intel Core i7-8665U CPU and 16 GB of memory.

A. *Experimental Setup*

In this experiment, six types of benchmarks were used, combining three map sizes (10×15, 20×30, and 40×60) with two droplet sizes (1×1 and 3×3). For each case, 10 maps are randomly generated. In all cases, the number of droplets is set to 3. The number of obstacles $|B|$ is set to 30, 120, and 480 for smaller to larger map sizes.

Droplet size 3×3, the number of droplets 3 means that there are three clusters of droplets each occupying a 3×3 square region. Thus, for droplet size 3×3 and the number of droplets 3, the number of unit droplets $|D|$ is 27.

For each map, source and target cells of the specified cell matrix size are randomly generated without overlap, and random obstacles are designated from other cells. If there is no feasible droplet routing in a generated map, the map is regenerated to ensure only solvable maps are used in the benchmarks.

B. *Experimental Results*

The experimental results are summarized in **Table I**. For each benchmark, droplet routing algorithm (DRA) [10] and FADR were applied.

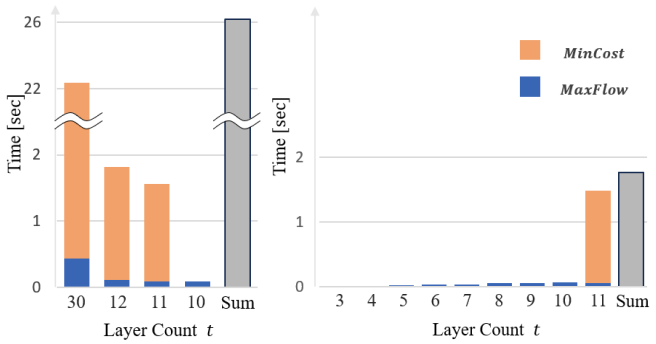
In DRA, minimum-cost maximum flows are repeatedly obtained by gradually reducing the number of layers of the 3D flow graph until no feasible droplet routing exists. The initial number of layers is set large enough to ensure feasible droplet routing. The number of layers in the next iteration is set one less than the time step when all unit droplets arrive at target cells. The initial number of layers was manually set to 15, 30, or 50 depending on the map sizes. DRA aims to reduce the number of iterations as small as possible by finding a minimum-cost maximum flow.

In the table, "#step" indicates the average of the minimum time steps of droplet routing, and "time" indicates the average computation time excluding the initialization, where "total", "maxF", and "minC" are the total computation time, the computation time for maximum flow searches, and the computation time for minimum-cost search, respectively. "#rep" is the average number of flow searches.

In experiments, it is confirmed that the computation time of FADR is 36% – 84% shorter than DRA. The computation

TABLE I: Average computation time and the number of repetitions of flow search.

Benchmark							DRA [10]		FADR(Ours)			
	Map		Droplet			#step	time	#rep	time			#rep
Type	size	$\|B\|$	size	#	$\|D\|$		total [s]		total [s] (%)	maxF [s]	minC [s]	
A-1	10×15	30	1×1	3	3	8.2	0.14	2.4	**0.09** (64)	0.05	0.03	6.8
A-2	10×15	30	3×3	3	27	7.4	1.77	2.4	**0.55** (31)	0.14	0.41	7.2
B-1	20×30	120	1×1	3	3	17.2	3.15	2.4	**1.38** (43)	0.76	0.62	12.8
B-2	20×30	120	3×3	3	27	16.1	39.94	3.6	**6.86** (17)	1.31	5.55	14.2
C-1	40×60	480	1×1	3	3	34.6	50.09	2.5	**27.48** (55)	13.25	14.24	24.4
C-2	40×60	480	3×3	3	27	28.1	545.07	5.3	**84.87** (16)	17.40	67.47	24.1



(a) DRA: Total time 26.02 sec (b) FADR: Total time 1.76 sec

Fig. 4 : Computation times for a map in B-2 where "#step" is 11.

time of FADR is much shorter than DRA, especially for 3×3 droplets. Although the number of repetitions of flow searches in FADR is larger than that of DRA, the total computation time of FADR is shorter since the computationally expensive minimum-cost maximum flow search is only executed once in FADR. This is also confirmed in **Figure 4**, which shows the computation time of each flow search and the total computation time of a map in benchmark B-2.

The total computation time of maximum flow searches in FADR depends on the map size and the number of unit droplets. However, in experiments, increasing the number of unit droplets by nine times only increases the computation time up to three times. The number of unit droplets does not significantly affect the computation time in FADR.

The total computation time of the minimum-cost maximum flow search in FADR also depends on the map size and the number of unit droplets. In experiments, roughly speaking, it grows proportionally with the number of unit droplets.

Examples of the routing paths obtained by FADR for the cell map shown in **Figure 1** are shown in **Figure 5**. The blue cells represent the cells the unit droplets pass through, and the yellow lines represent the routing paths of unit droplets. The width of a yellow line in a cell is drawn proportional to the number of unit droplets moving along the line. It is observed that two droplets are split into unit droplets and eventually combined at the target cell. **Figure 5(a)** and **Figure 5(b)** show the routing results before and after applying the minimum-cost flow search in FADR, respectively. Each corresponds to a minimum-time routing, but redundant movements are reduced in **Figure 5(b)**.

In FADR, the number of layers of the 3D flow graph is increased by one if a maximum flow search fails, and the number of repetitions of maximum flow searches increases according to the map size. A further reduction of the number of flow searches is a future challenge. The number of cells used in the routing obtained by FADR is not necessarily the minimum. An efficient reduction of the number of cells used is also among our future challenges.

VI. Conclusion

This paper proposed a method, called FADR, that efficiently finds a minimum time routing path of single type droplets. The proposed FADR reduces the number of flow explorations and finds a droplet routing path with the minimum routing time in a short computation time. The proposed method is expected to contribute to shortening the required time for experiments using MEDA, thereby advancing the medical and chemical fields.

Future challenges include establishing a general routing strategy that finds a routing of multiple types of droplets. To apply our FADR method to multiple droplet routing, the cells used in the routing of a unit droplet need to be further smaller. FADR does not necessarily well minimize the number of cells used in droplet routing.

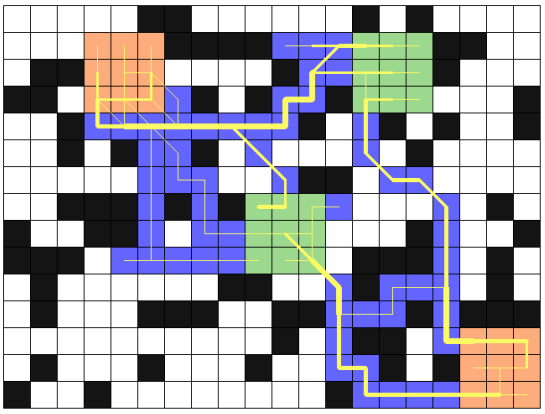
Acknowledgments

This work was partially supported by MEXT Initiative to Establish Next-generation Novel Integrated Circuits Centers (X-NICS) Grant Number JPJ011438.

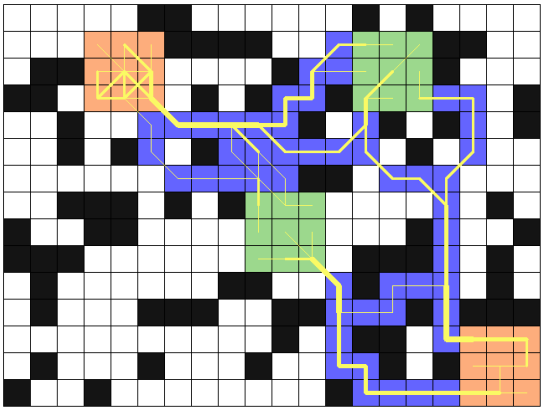
References

[1] Richard B. Fair. Digital microfluidics: Is a true lab-on-a-chip possible? *Microfluidics and Nanofluidics*, 3(3):245–281, 2007.

[2] Zhanwei Zhong, Zipeng Li, Krishnendu Chakrabarty, Tsung-Yi Ho, and Chen-Yi Lee. Micro-electrode-dot-array digital microfluidic biochips: Technology, design



(a) Routing path before minimum-cost flow search in FADR (#cell=94)



(b) Routing path obtained by FADR (#cell = 86)

Fig. 5 : Routing path obtained by FADR: #step=14 (Map size 15×20, $|B| = 75$, droplet size 3×3, the number of droplets 2).

automation, and test techniques. *IEEE Transactions on Biomedical Circuits and Systems*, 13(2):292–313, 2019.

[3] Sung Kwon Cho, Hyejin Moon, and Chang-Jin Kim. Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits. *Journal of Microelectromechanical Systems*, 12(1):70–80, 2003.

[4] Zhongkai Chen, Daniel Hsiang-Yung Teng, Gary Chung-Jhih Wang, and Shih-Kang Fan. Droplet routing in high-level synthesis of configurable digital microfluidic biochips based on microelectrode dot array architecture. *BioChip Journal*, 5:343–352, 2011.

[5] Zipeng Li, Kelvin Yi-Tse Lai, John McCrone, Po-Hsien Yu, Krishnendu Chakrabarty, Miroslav Pajic, Tsung-Yi Ho, and Chen-Yi Lee. Efficient and adaptive error recovery in a micro-electrode-dot-array digital microfluidic biochip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(3):601–614, 2018.

[6] Sarit Chakraborty and Susanta Chakraborty. Routing performance optimization for homogeneous droplets on MEDA-based digital microfluidic biochips. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 419–424, 2019.

[7] Ikuru Yoshida, Kota Asai, Tsung-Yi Ho, and Shigeru Yamashita. Droplet splitting routing for micro-electrode-dot-array digital microfluidic biochips. In *Synthesis And System Integration of Mixed Information technologies (SASIMI)*, pages 110–115, 2019.

[8] Chiharu Shiro, Hiroki Nishikawa, Xiangbo Kong, Hiroyuki Tomiyama, Shigeru Yamashita, and Sudip Roy. Shape-dependent velocity based droplet routing on MEDA biochips. *IEEE Access*, 10:122423–122430, 2022.

[9] Chiharu Shiro, Hiroki Nishikawa, Kong Xiangbo, Hiroyuki Tomiyama, and Shigeru Yamashita. Routing with washing droplets in MEDA biochips. In *IEICE Technical Report (VLD2022-84)*, volume 122, pages 67–72, 2023. (in Japanese).

[10] Katsuharu Yamamoto, Akira Jinguji, and Atsushi Takahashi. Droplet routing algorithm for MEDA-based DMFB. In *IPSJ DA Symposium*, pages 173–179, 2023. (in Japanese).

[11] Issei Nakamura, Shigeru Yamashita, Hiroyuki Tomiyama, and Ankur Gupta. Speeding up a routing method considering droplet division on MEDA biochips by Dijkstra's method. In *IEICE Technical Report (VLD2024-104)*, volume 124, pages 7–12, March 2025. (in Japanese).

[12] Biddut Bhattacharjee and Homayoun Najjaran. Size dependent droplet actuation in digital microfluidic systems. In *Proc. SPIE 7318, Micro- and Nanotechnology Sensors, Systems, and Applications*, number 73180H, May 2009.

[13] Pampa Howladar, Pranab Roy, and Hafizur Rahaman. Micro-electrode-dot array based biochips : Advantages of using different shaped CMAs. In *Proc. IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 296–301, July 2019.

[14] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[15] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. Computer science. McGraw-Hill, 2009.