

Towards droplet size-aware biochemical application compilation for AM-EWOD biochips

Mirela Alistar, Paul Pop

Technical Univ. of Denmark, DK-2800 Kgs. Lyngby

Phone: +45 4525 3732, Fax: +45 4593 3031, email: mali@dtu.dk

Abstract—Microfluidic-based biochips are replacing the conventional biochemical analyzers, and are able to integrate on-chip all the necessary functions for biochemical analysis using microfluidics. The digital microfluidic biochips are based on the manipulation of liquids not as a continuous flow, but as discrete droplets on an array of electrodes. Microfluidic operations, such as transport, mixing, split, are performed on this array by routing the corresponding droplets on a series of electrodes. Several approaches have been proposed for the compilation of digital microfluidic biochips, which, starting from a biochemical application and a given biochip architecture, determine the allocation, resource binding, scheduling, placement and routing of the operations in the application. To simplify the compilation problem, researchers have assumed an abstract droplet size of one electrode. However, the droplet size abstraction is not realistic and it impacts negatively the execution of the biochemical application, leading in most cases to its failure. Hence the existing compilation approaches have to be revisited to consider the size of the droplets. In this paper we take the first step towards a droplet size-aware compilation by proposing a routing algorithm that considers the droplet size. Our routing algorithm is developed for a novel digital microfluidic biochip architecture based on Active Matrix Electrowetting on Dielectric, which uses a thin film transistor array for the electrodes. We also implement a simulator that allows us to perform the needed adaptations and to validate the proposed routing algorithm.

I. INTRODUCTION

Microfluidic biochips have the potential to replace the conventional laboratory equipment as they integrate all the functions needed to complete a bioassay. Applications on biochips are considered in areas such as drug discovery, clinical diagnosis, DNA sequencing, protein analysis and immunoassays [1]–[3].

In this paper, we are interested in microfluidic platforms that manipulate the fluids as droplets, using electrowetting-on-dielectric (EWOD) [4]. We call such platforms *digital microfluidic biochips* (DMBs). DMBs are able to perform operations such as dispensing, transport, mixing, split, dilution and detection using droplets (discrete amount of fluid of nanoliters volume) [5]. A biochip is typically connected to a computer (or microcontroller) as shown in Fig. 1a. The biochip is controlled by the “electrode actuation sequence” that specifies for each time step which electrodes have to be turned on and off.

Researchers have presented approaches for the compilation of digital microfluidic biochips, which, starting from a biochemical application model and a given biochip architecture,

determine the electrode actuation sequence needed to run the application on the biochip. The compilation process consists of the following tasks: allocation, resource binding, scheduling, placement and routing of the operations in the application. So far, researchers have ignored the size of a droplet, and assumed that it covers one single electrode, even when droplets undergo merging and splitting. This was a useful simplifying assumption for the compilation work.

However, recently, a novel digital microfluidic biochip architecture has been developed based on Active Matrix Electrowetting on Dielectric (AM-EWOD), which uses a thin film transistor (TFT) array for the electrodes [6]. One of the advantages of the AM-EWOD technology is that it allows much larger arrays, with smaller electrode sizes, permitting precise control of the droplet size and shape. In addition, an AM-EWOD biochip can successfully actuate droplets that cover a large number of electrodes. For example, a colorimetric assay that measures glucose in human blood was executed on an AM-EWOD architecture using droplets of 100-electrode size. Hence, in the context of an AM-EWOD biochip, the size of the droplets can no longer be abstracted, i.e., there is an imperative need for droplet size-aware compilation methods.

There is a significant amount of work on biochemical compilation for conventional EWOD biochips [3], [7]–[10]. However, to the best of our knowledge, there is no previous work on compilation for AM-EWOD biochips. In this paper we take a first step in this direction by developing an approach to routing droplets for AM-EWOD devices, which takes the actual droplet sizes into account.

II. BIOCHIP ARCHITECTURE

A. EWOD-based digital microfluidic biochip

In an EWOD DMB, a droplet is sandwiched between a top ground-electrode and bottom control-electrodes, see Fig. 1b. Two glass plates, a top and a bottom one, protect the droplets

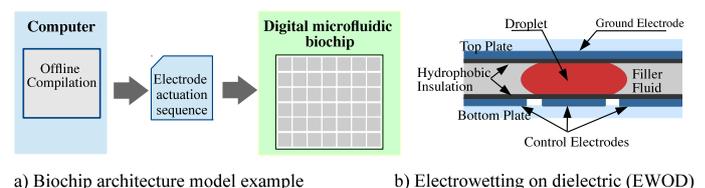


Fig. 1: Biochip architecture model example

from external factors. The droplets are manipulated using the EWOD principle [4]. For example, in Fig. 1b, if the control-electrode on which the droplet is resting is turned off, and the left control-electrode is activated by applying voltage, the droplet will move to the left. In this paper, we consider that an electrode is “on” (i.e., on-electrode) when it is activated and “off” (i.e., off-electrode) when it is not activated.

A DMB is modeled as a two-dimensional array of identical control-electrodes, see Fig. 3b, where each electrode can hold a droplet. There are two types of operations: reconfigurable (mixing, split, dilution, merge, transport), which can be executed on any electrode on the biochip, and non-reconfigurable (dispensing, detection), which are bound to a specific device such as a reservoir, a detector or a sensor. A mixing operation is executed when two droplets are moved to the same location and then transported together according to a specific pattern (see the mixing module M_1 in Fig. 3b). A segregation border is needed to prevent accidental droplet merging. A split operation is done by keeping the electrode on which the droplet is resting turned off, while applying concurrently the same voltage on two opposite neighboring electrodes.

B. AM-EWOD digital microfluidic biochip

Let us present the biochip architecture based on active matrix electrowetting on dielectric (AM-EWOD), proposed by [6]. An AM-EWOD architecture has a transparent TFT backplane which is responsible for applying voltage on the electrodes in order to actuate the droplets.

The AM-EWOD biochips have large arrays of electrodes compared to the conventional EWOD biochips. The AM-EWOD technology permits manipulation of droplets that occupy a large number of electrodes, i.e., in the range of tens of electrodes. In addition, the droplets can have various shapes. The shape of the droplets can be adjusted on demand for purposes such as faster routing and enhanced active mixing. AM-EWOD biochips can perform the same fluidic operations (e.g., transport, mixing, split, dispensing, etc) as the EWOD biochips at a much lower voltage (20 V).

The droplet actuation on AM-EWOD is performed differently than on EWOD architecture. There is ongoing research to develop algorithms that decide the electrode actuation sequence for each operation depending on the droplet viscosity, size and shape. As presented in Fig. 1b, on a EWOD architecture to move the droplet to the left, we need to turn on the left electrode and keep the middle electrode turned off.

On AM-EWOD, to move a large sized droplet, we have to turn on several columns of electrodes depending on the viscosity of the droplet. In Fig. 2 we show in the top, middle and bottom row, the droplet transport, splitting and morphing on a 120×120 AM-EWOD array. We group the columns of electrodes needed to be turned in order to move the droplet, in the “front line” of the droplet. In Fig. 2 we marked with green the front line of the droplet and with gray the off-electrodes. When actuated, the droplet does not keep its circle shape, but has a semi-elliptic shape. To split the droplet on AM-EWOD, we need to iteratively turn on the columns on

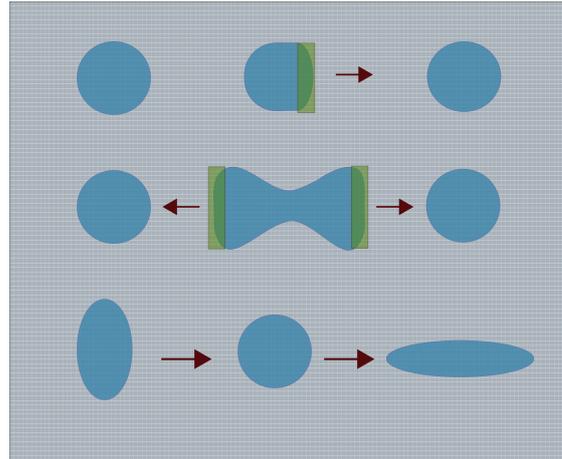


Fig. 2: Example droplet transport, splitting and morphing on AM-EWOD

the right and left sides of the droplet until the droplet snaps in two equal daughter droplets.

The droplet can also have a plug shape (see Fig. 2, bottom row) and morph into plugs of different widths. To morph, first all the electrodes underneath of the droplet are turned off and the droplet “relaxes” in a circle shape. Then, the electrodes for the desired shape are turned on and the droplet morphs.

III. BIOCHEMICAL APPLICATION COMPILATION

A biochemical application is modeled [2] using an acyclic directed graph, where the nodes represent the operations, and the edges represent the dependencies between them. Let us consider the biochemical application, depicted as a graph in Fig. 3a, which has fifteen operations. The directed edge between O_1 and O_3 signifies that operation O_1 has to finish before operation O_3 can start executing. Mixing operation O_3 uses the output droplets issued by operations O_1 and O_2 .

To be executed on a DMB, a biochemical application has to be compiled. The compilation typically consists of the following tasks, depicted schematically in Fig. 3: *modeling* of the

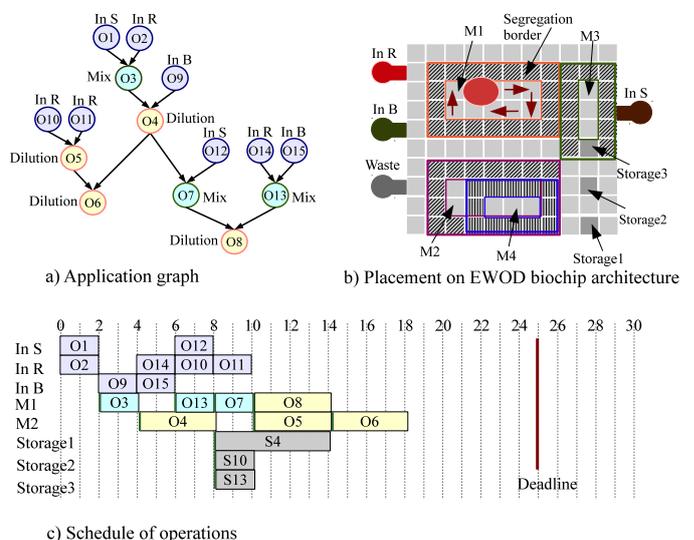


Fig. 3: Example compilation tasks

biochemical application functionality and biochip architecture, *allocation*, during which the needed modules are selected from a module library, *binding* the selected modules to the biochemical operations in the application, *placement*, during which the positions of the modules on the biochip are decided, *scheduling*, when the order of operations is determined and *routing* the droplets to the needed locations on the biochip.

A. Motivation

The compilation of a biochemical application on a biochip is an NP-complete problem [2]. Hence, to simplify the problem, the compilation implementations for EWOD biochips use the following assumption regarding the size of the droplets: all droplets are considered to have an “1x” abstract size, where “x” is the size of one electrode unit (see the “1x” mixing droplet in Fig. 3b). However, most biochemical applications use sequential mixing operations to produce intermediate droplets with a size that occupies several electrodes.

Next, we will discuss the impact of the droplet size abstraction on the execution of the biochemical application. Let us consider that at a moment t the running application has to perform a split operation. As depicted in Fig. 4a, the current compilations assume that a split operation is performed on a “1x” droplet resulting in two daughter droplets of “1x” size each. We denote with “ E_{1x} ” the electrode actuation sequence determined under the “1x” abstraction.

In Fig. 4 we show the impact of applying “ E_{1x} ” on droplets of a size larger than “1x”. Note that the depiction in Fig. 4 is schematic, i.e., the filler fluid and the hydrophobic layers are not presented. Also, in Fig. 4, we marked with green the on-electrodes and with blue the off-electrodes. When we apply E_{1x} on a real-life droplet of a “2x” size, the resulting forces split unequally the “2x” droplet (Fig. 4b). An unbalanced split is the most frequent cause of erroneous bioassays [11], [12].

Moreover, as depicted in Fig. 4c, E_{1x} does not work on a droplet with a size $\geq 3x$, i.e., the droplet will not split, but it will remain on the same position. The electrode actuation sequences that will successfully complete the split operation for the “2x” and “3x” droplets are presented in Fig. 4d and e, respectively. Hence, in order to ensure the correctness of a biochemical application we need a new electrode actuation sequence “ E_{Nx} ” determined for the actual size of the droplets.

The straightforward solution is to “scale” the droplet size from “1x” to “Nx” and to apply the transformation on “ E_{1x} ” to obtain “ E_{Nx} ”. Considering the traditional EWOD biochips, there are several problems with this approach:

(i) Scaling the droplet size will result in droplet congestion and accidental merging. To solve that, the biochip area has to be scaled resulting in an increased cost.

(i) In practice, engineers prefer to actuate the droplet in the shape of plug (Fig. 5b) instead of a circle (Fig. 5a). Actuation of a droplet as a circle [4] requires a pin connection for each electrode. Such a biochip architecture has as disadvantages complex wiring and increased cost. Because of that, real-life applications have been exclusively developed for pin-constrained biochips [13], [14] that have assigned multiple

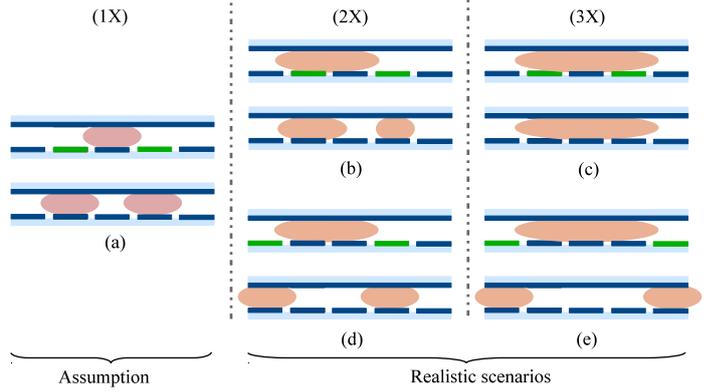


Fig. 4: Example wrong electrode actuation sequence

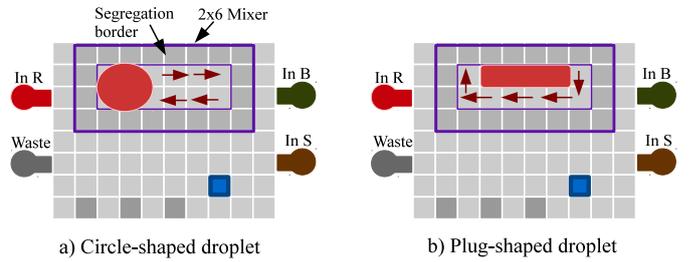


Fig. 5: Example circle and plug-shaped droplets

electrodes to the same pin. On such architectures, in most of the case the droplet can only be actuated as a plug, as depicted in Fig. 5b.

Hence, for a correct real-life use of EWOD and AM-EWOD biochips, we cannot use the electrode actuation sequence determined by existing compilation approaches. Hence, there is an imperative need for a new compilation approach, that is, a droplet size-aware compilation, which determines the electrode actuation sequence considering the actual size of the droplets.

IV. TOWARDS DROPLET SIZE-AWARE COMPILATION

In this paper we take steps towards implementing a droplet size-aware compilation for AM-EWOD biochips. The compilation uses a given library of algorithms \mathcal{L}_S for droplet actuation. The library \mathcal{L}_S contains for each operation the algorithm that determine the electrode sequence needed to actuate the droplets such that the operation is completed. The algorithms in \mathcal{L}_S have to consider that droplets can morph in various shapes depending on their viscosity, size and current shape.

The shape morphing property, not present in traditional EWOD biochips, offers new advantages to our droplet size-aware compilation. The placement and routing tasks can consider the droplet size and shape to minimize the application completion time. Hence the droplet size-aware compilation has more degrees of freedom since now the droplet size and shape can be controlled.

A. Droplet Routing

Due to the complexity of the compilation problem for AM-EWOD, we started with addressing first the droplet routing subproblem. Droplet routing is an NP-complete problem [15], with a lot of papers focused on the trade off between (i) finding a minimal route and (ii) avoiding droplet congestion. Most of the droplet routing approaches in the literature are based on maze routing and line searching algorithms used intensively for grid routing in Very-Large-Scale Integration (VLSI) systems [16] and for robot navigation among obstacles on a grid [17]–[19]. To avoid congestion, negotiation algorithms [15] and priority metrics [20], [21] were proposed. However for AM-EWOD biochips, we need routing strategies that consider the shape morphing property of the droplets.

We formulate the routing problem as follows: given as input an AM-EWOD architecture \mathcal{A} , the library \mathcal{L}_S and the current droplet to be scheduled $Drop$, we want to determine the route R_{Drop} between the source S and the target T , such that the application completion time is minimized. We consider that the binding and the placement of operations have been already decided. Hence, the route R_{Drop} has to be determined such that it avoids the “blocked” electrodes, i.e., the electrodes currently occupied by the operations currently executed. We group these electrodes in areas called “obstacles”. Hence, our problem reduces to finding a route R_{Drop} from the source S to the target T avoiding the obstacles on the grid \mathcal{A} . However, R_{Drop} has to be determined not for a cell of the grid, but for a droplet $Drop$ that is characterized by its shape, size and viscosity. In addition, $Drop$ can morph—property that allows a better use of the available area. In this paper we propose a routing approach that extends Soukup’s algorithm [22] to consider the droplet shape morphing and the fluidic constraints.

In Soukup’s algorithm a line segment is drawn starting from the source and moving towards the target. The segment is always extended towards the target, keeping the direction unchanged for as long as it moves towards the target. When the line segment meets an obstacle, an adapted version of Lee’s algorithm [23] is applied to the electrodes on the current line segment in order to search around the obstacle. Hence, the algorithm graphically “fills” the array of electrodes starting from the source and labeling with i the electrodes located at Manhattan-distance i from the source. The filling phase stops when an electrode in the direction of the target is reached. Another line segment is extended from that electrode. The search for a route continues in this manner until the target electrode is reached.

B. Proposed routing algorithm

We adapted Soukup’s algorithm to consider the size and shape of the droplet. Fig. 6 presents our routing algorithm that takes as input the droplet $Drop$, the source S , the target T , the AM-EWOD biochip \mathcal{A} and the library \mathcal{L}_S and outputs the route of the droplet.

The **MergeObst** function merges the obstacles that do not have enough space between them to allow the droplet to pass

```

FindRoute( $Drop, S, T, \mathcal{A}, \mathcal{L}_S$ )
1 MergeObst( $Drop, \mathcal{A}, \mathcal{L}_S$ )
2  $dir = \text{InitialDirection}(Drop, S, T, \mathcal{A})$ 
3 repeat
4    $R_{Drop} = \text{Move}(Drop, dir, T, \mathcal{A})$ 
5   if  $dir$  away from  $T$  then
6      $dir = \text{ChangeDirection}(Drop, T, \mathcal{A}, \mathcal{L}_S)$ 
7     if  $dir$  is null then
8        $ShapeQ = \text{SelectShape}(Drop, \mathcal{L}_S)$ 
9        $Sh = \text{FindFittingShape}(ShapeQ, dir, \mathcal{A})$ 
10       $Drop = \text{Morph}(Sh)$ 
11       $dir = \text{GetDirection}(Drop, T, \mathcal{A}, \mathcal{L}_S)$ 
12    end if
13     $R_{Drop} = \text{Move}(Drop, dir, T, \mathcal{A})$ 
14  end if
15  if  $Drop$  is blocked then
16     $R_{Drop} = \text{FindDetour}(Drop, T, \mathcal{A}, \mathcal{L}_S)$ 
17     $dir = \text{SetDirection}(Route_D, T)$ 
18  end if
19 until  $T$  is reached
20 return  $R_{Drop}$ 

```

Fig. 6: Routing algorithm

by. The challenge lays in finding an obstacle-merging strategy that speeds up the search by eliminating invalid routes, but it does not take away useful area. Since the droplet can morph into various shapes, **MergeObst** uses the shape library \mathcal{L}_S to calculate the minimum frontal line w that the current droplet can take. **MergeObst** selects the blocked electrodes that are located at a distance $\leq w$ and “merges” them, i.e., it marks as blocked the electrodes in between. For example, in Fig. 7, the obstacle O_2 and O_3 are located at a 2-electrode distance on the right and the left side, respectively. Assuming $w = 3$, obstacles O_2 and O_3 are merged by marking as blocked the electrodes in between, which are hashed in Fig. 7.

After the obstacles are merged, the algorithm decides the initial direction for the droplet movement such that the droplet is routed towards the target. In case two directions are possible, one of them is randomly selected. Similarly to Soukup’s algorithm, the current direction is changed only when the droplet is moving away from the target or when an obstacle is met. The algorithm stops when the target is reached.

Next, we will present in detail the two situations when the direction needs to be changed: (1) when the droplet movement heads away from the target (lines 5–14 in Fig. 6) and (2) when an obstacle is met (lines 15–18 in Fig. 6).

For the first case, we update the current direction such that the droplet moves towards the target. For example, considering the setup from Fig. 7, where the source and target locations are marked with S and T , respectively, the current direction needs to be changed towards the target, i.e., the droplet has to take a right turn. Once the new direction is decided, the algorithm checks if there is enough space to move the droplet keeping its current shape (line 6 in Fig. 6). In case there is not enough space, the algorithm tries to find Sh —the new shape that fits within the available area (lines 8–10). To find Sh , the

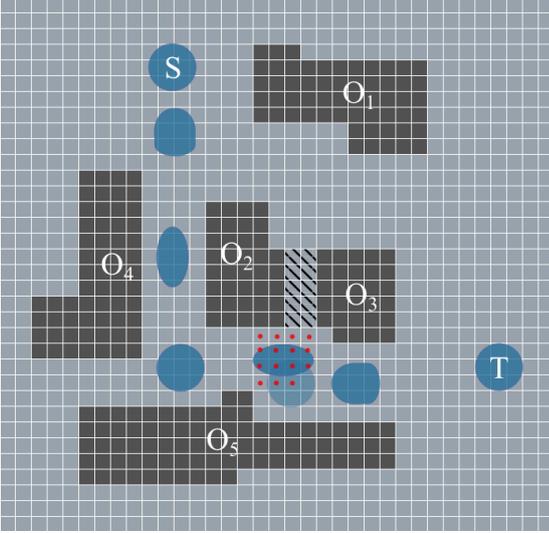


Fig. 7: Example routing

function **SelectShape** applies the corresponding algorithm in \mathcal{L}_S to determine $ShapeQ$ —the queue of possible new shapes for $Drop$. $ShapeQ$ is sorted such that the shape with the largest front line is first.

Next, the function **FindFittingShape** searches through $ShapeQ$ and returns the fitting shape with the largest front line. By doing so, we select the “fastest” shape, i.e., the shape that requires less electrode actuation steps to move the centroid (i.e., the geometrical center) of the droplet to a certain location. The droplet is then morphed into the new shape (line 10). As presented in Section III-A, the morphing step implies that the droplet first relaxes in a circle shape (i.e., the underneath electrodes are turned off) and the electrodes corresponding to the new shape are turned on. However, because of the obstacles reducing the available area, a fitting shape might not be found in \mathcal{L}_S . Also, even when moving towards the target, the droplet might encounter obstacles.

In case (2), when the droplet movement is blocked, **FindDetour** (line 16) searches for a detour around the obstacles. Our strategy to find a detour, presented in Fig. 8, takes as input the droplet $Drop$, the target T , the current electrode array \mathcal{A} , the library \mathcal{L}_S and outputs the detour route R_D . We start by applying the algorithms from the library \mathcal{L}_S to determine the queue $ShapeQ$ that contains all the shapes into which $Drop$ can morph. Next, we search for a shape Sh_i that can be routed towards the target. For that we apply Lee’s algorithm (line 5) on the centroid C_i of Sh_i and search the electrodes until we find an unblocking electrode (i.e., not occupied by obstacles or by the current droplet) in the direction of the target. In Fig. 7 the searched electrodes are marked with red. Lee’s algorithm returns R_C —the route for the centroid C_i .

Finally, we need to check if there is enough space along R_C for the whole droplet to fit. Previously, we have merged the obstacles such that there is a droplet shape to fit in any route between any two obstacles (see **MergeObst** function, line 1 in Fig. 6). Hence, we know that for any electrode $\in R_C$ there is a fitting shape $Sh_i \in \mathcal{L}_S$ for the droplet. For that reason,

```

FindDetour( $Drop, T, \mathcal{A}, \mathcal{L}_S$ )
1  $ShapeQ = \text{SelectShape}(Drop, \mathcal{L}_S)$ 
2 repeat
3    $Sh_i = \text{Remove}(ShapeQ)$ 
4    $C_i = \text{Centroid}(Sh_i)$ 
5    $R_C = \text{LeeRouting}(C_i, T, \mathcal{A})$ 
6    $R_D = \text{ValidateRoute}(R_C, Sh_i, \mathcal{L}_S)$ 
7 until ( $R_D \neq \text{null}$ ) or ( $ShapeQ = \emptyset$ )
8 if  $R_D$  is null then
9   Wait( $t$ )
10   $\mathcal{A} = \text{UpdateObstacleMap}()$ ;
11   $\text{FindRoute}(Drop, S, T, \mathcal{A}, \mathcal{L}_S)$ 
12 end if
13 return  $R_D$ 

```

Fig. 8: Algorithm for finding a detour around an obstacle

the function **ValidateRoute** (line 6) only checks the transition rules for morphing along $Route_C$. In case the algorithm cannot find any valid route for any of possible shapes $Sh_i \in ShapeQ$, the droplet is set to wait for a specific amount of time t (line 9). After t s, the obstacle map is updated (line 10) and the **FindRoute** algorithm is re-executed.

V. SIMULATOR FOR AM-EWOD BIOCHIPS

As mentioned, the approach for a droplet size-aware compilation should consider the droplet shape morphing property. By doing so, the routing of droplets and placement of operations can be improved to use better the available biochip area. Hence, the droplet size-aware compilation will obtain better results in terms of application completion time. However, to benefit of droplet shape morphing, the compilation has to rely on a library of algorithms \mathcal{L}_S . The problem is that the AM-EWOD technology is novel and currently there is little work on designing biochemical applications for AM-EWOD biochips.

A simulator will speed up the characterization of library \mathcal{L}_S by providing visual feedback and helping the designer to adjust and validate the algorithms. For that purpose we started to build a simulator that graphically depicts the execution of the biochemical protocol. We focused on the routing problem and used the simulator to help us understand how the droplet size and shape impact the routing task.

The simulator is implemented in C programming language and uses as input the route generated by our proposed **FindRoute** (presented in Fig. 6 and described in detail in Section IV-B). The simulator uses the Synchronized Multimedia Integration Language (SMIL) and Scalable Vector Graphics (SVG) technologies to build a motion picture (video) of the droplet routing. Hence, we can observe the droplet morphing along the route decided by our algorithm.

The visual feedback provided by the simulator showed us that we can further improve our routing algorithm to favor routes between obstacles rather than “freeways” in order to better exploit the area. An example is the case presented in Fig. 9, where our routing algorithm decides on the shortest route R_1 . However, due to the size of the droplets, route R_1 uses a lot of available space that can be used for other operations.

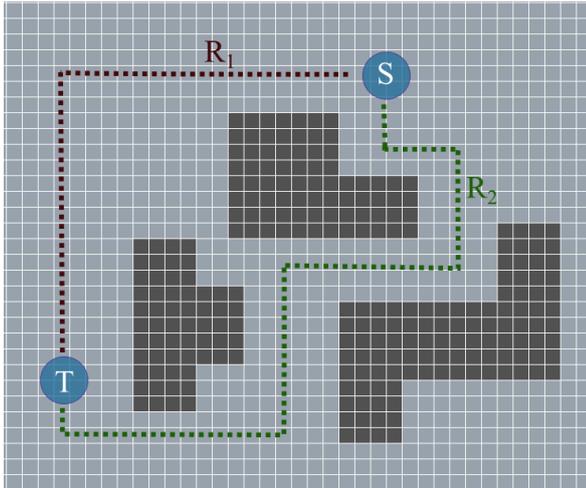


Fig. 9: Route example

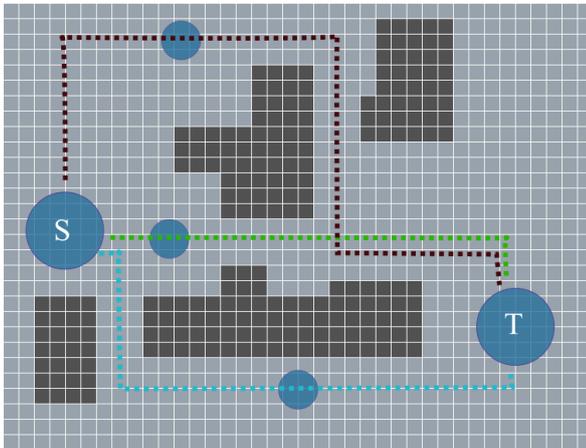


Fig. 10: Parallel routes

Hence, we can improve our routing algorithm to favor a longer route, such as route R_2 that uses the area between the obstacles leaving the rest of the area to be available for operations that have higher area requirements.

Another impact of the droplet size and shape on routing was observed for the case depicted in Fig. 10, where a valid route could not be found by our algorithm. For that case, we can adapt the routing algorithm to consider splitting the droplet in two or more daughter droplets that can be routed in the available area and finally merged back into the initial droplet.

VI. DISCUSSION AND FURTHER WORK

In this paper we have taken the first steps towards a droplet size-aware compilation by proposing a routing algorithm that considers the actual size and shape of a droplet. The routing algorithm was applied for the AM-EWOD biochips, which can manipulate significantly larger droplets with precise control of their size and shape. We also started to develop a simulator that gives visual feedback on the real-time execution of the biochemical application. Using the simulator, we studied the impact of droplet size and shape on the routing problem.

We concluded that the current routing algorithm can be further improved to favor routes that use the area between

obstacles and to consider splitting the droplet into daughter droplets that are routed separately and merged back when they all reach the target. For further work we will investigate the impact of droplet size and shape on the other compilation tasks, such as the placement of operations.

REFERENCES

- [1] D. Mark, S. Haeberle, G. Roth, F. von Stetten, and R. Zengerle, "Microfluidic lab-on-a-chip platforms: requirements, characteristics and applications," *Chemical Society Reviews*, vol. 39, no. 3, pp. 1153–1182, 2010.
- [2] K. Chakrabarty and F. Su, *Digital microfluidic biochips: synthesis, testing, and reconfiguration techniques*. CRC Press, 2006.
- [3] K. Chakrabarty, R. B. Fair, and J. Zeng, "Design tools for digital microfluidic biochips: toward functional diversification and more than Moore," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 7, pp. 1001–1017, 2010.
- [4] M. G. Pollack, "Electrowetting-based microactuation of droplets for digital microfluidics," Ph.D. dissertation, Duke University, Durham, NC, 2001.
- [5] R. B. Fair, "Digital microfluidics: is a true lab-on-a-chip possible?" *Microfluidics and Nanofluidics*, vol. 3, no. 3, pp. 245–281, 2007.
- [6] B. Hadwen, G. Broder, D. Morganti, A. Jacobs, C. Brown, J. Hector, Y. Kubota, and H. Morgan, "Programmable large area digital microfluidic array with integrated droplet sensing for bioassays," *Lab on a Chip*, vol. 12, no. 18, pp. 3305–3313, 2012.
- [7] D. Grissom and P. Brisk, "Fast online synthesis of generally programmable digital microfluidic biochips," pp. 413–422, 2012.
- [8] J.-D. Huang, C.-H. Liu, and T.-W. Chiang, "Reactant minimization during sample preparation on digital microfluidic biochips using skewed mixing trees," pp. 377–383, 2012.
- [9] E. Maftai, P. Pop, and J. Madsen, "Routing-based synthesis of digital microfluidic biochips," *Design Automation for Embedded Systems*, vol. 16, no. 1, pp. 19–44, 2012.
- [10] P. Roy, H. Rahaman, C. Giri, and P. Dasgupta, "Modelling, detection and diagnosis of multiple faults in cross referencing dmfb," pp. 1107–1112, 2012.
- [11] T. Xu and K. Chakrabarty, "Fault modeling and functional test methods for digital microfluidic biochips," *Trans. on Biomedical Circuits and Systems*, vol. 3, no. 4, pp. 241–253, 2009.
- [12] D. Rose, "Microdispensing technologies in drug discovery," *Drug discovery today*, vol. 4, no. 9, pp. 411–419, 1999.
- [13] R. Sista, Z. Hua, P. Thwar, A. Sudarsan, V. Srinivasan, A. Eckhardt, M. Pollack, and V. Pamula, "Development of a digital microfluidic platform for point of care testing," *Lab on a Chip*, vol. 8, no. 12, pp. 2091–2104, 2008.
- [14] "Advanced Liquid Logic," <http://www.liquid-logic.com>.
- [15] P.-H. Yuh, C.-L. Yang, and Y.-W. Chang, "Bioroute: A network-flow-based routing algorithm for the synthesis of digital microfluidic biochips," *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 11, pp. 1928–1941, 2008.
- [16] F. Su, W. Hwang, and K. Chakrabarty, "Droplet routing in the synthesis of digital microfluidic biochips," pp. 1–6, 2006.
- [17] T.-J. Pan and R. C. Luo, "Motion planning for mobile robots in a dynamic environment with moving obstacles," *IEEE*, pp. 578–583, 1990.
- [18] C.-P. Wu, T.-T. Lee, and C.-R. Tsai, "Obstacle avoidance motion planning for mobile robots in a dynamic environment with moving obstacles," *Robotica*, vol. 15, no. 05, pp. 493–510, 1997.
- [19] J.-C. Latombe, "Motion planning: A journey of robots, molecules, digital actors, and other artifacts," *The Int. Journal of Robotics Research*, vol. 18, no. 11, pp. 1119–1128, 1999.
- [20] T.-W. Huang and T.-Y. Ho, "A fast routability-and performance-driven droplet routing algorithm for digital microfluidic biochips," pp. 445–450, 2009.
- [21] M. Cho and D. Z. Pan, "A high-performance droplet routing algorithm for digital microfluidic biochips," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 10, pp. 1714–1724, 2008.
- [22] J. Soukup, "Fast maze router," pp. 100–102, 1978.
- [23] C. Y. Lee, "An algorithm for path connections and its applications," *IRE Trans. on Electronic Computers*, no. 3, pp. 346–365, 1961.