Exploring Speed and Energy Tradeoffs in Droplet Transport for Digital Microfluidic Biochips

Johnathan Fiske Daniel Grissom Philip Brisk
Department of Computer Science and Engineering
University of California, Riverside
Riverside, CA 92521, USA

Abstract - This paper transforms the problem of droplet routing for digital microfluidic biochips (DMFBs) from the discrete into the continuous domain, based on the observation that droplet transport velocity is a function of the actuation voltage applied to electrodes that control the devices. A new formulation of the DMFB droplet routing problem is introduced for the continuous domain, which attempts to minimize total energy consumption while meeting a timing constraint. Henceforth, DMFBs should be viewed as continuous, highly integrated cyber-physical systems that interact with and manipulate physical quantities, as opposed to inherently discrete and fully synchronized devices.

I. Introduction

Programmable *microfluidic* technology has the potential to revolutionize many subfields of (bio-)chemistry and bioengineering through miniaturization and automation [4, 5]. One such technology is the *Digital Microfluidic Biochip (DMFB)*, which manipulated discrete (hence the name "digital") droplets of liquid on a two-dimensional grid of electrodes [17], as shown in Fig. 1; Fig. 2 shows the basic operations supported: transport, merging and splitting, mixing, and storage. Over the past decade, there has been considerable interest in automating the process of compiling *assays* (biochemical protocols) into software programs that control the actuation of droplets on a DMFB [2, 10], shown in Fig. 3.

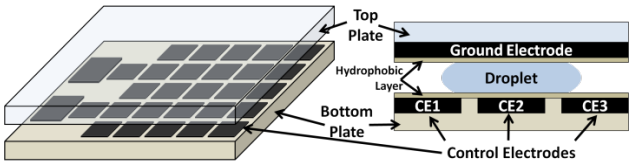


Fig. 1. A DMFB is a 2-dimensional electrode array, with I/O reservoirs on the periphery (a); a 1-dimensional cross-section; control electrodes (CEs) induce droplet motion according to the principles of electrowetting [17]; the hydrophobic layer provides insulation and prevents absorption (b).

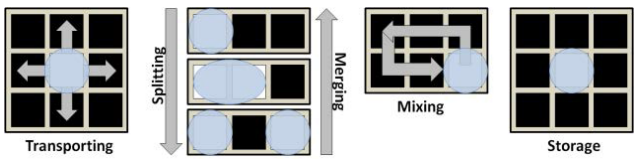


Fig. 2. The instruction set of a DMFB: droplet transport, splitting, merging, mixing, and storage. Each of these operations is achieved by activating a specific set of electrodes in sequence in the vicinity of one or more droplets.

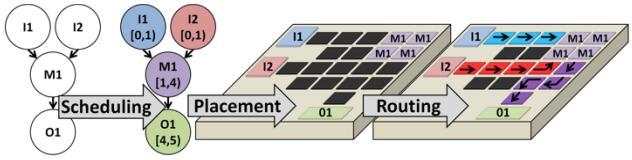


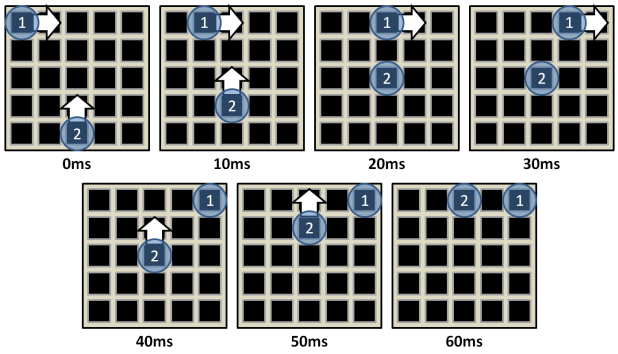
Fig. 3. The main steps of assay compilation; at present, assays are specified as directed acyclic graphs, without control flow. The focus of this paper is the routing stage.

A DMFB is a *cyber-physical system (CPS)*, meaning that it includes computational and physical components whose operation is tightly interleaved [12, 13]. One challenge in CPS design is that the computational domain is necessarily discrete and synchronized, i.e., actions occur on the granularity of *clock cycles*; in contrast, the physical domain is continuous. Prior work on DMFB compilation has imposed constraints on the system, which force it to operate in a discrete manner. One such example is the way that droplet transport, i.e., *routing* [1, 3, 11, 16, 18, 22, 25], is modeled.

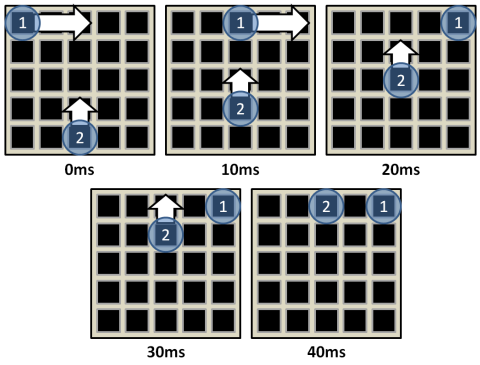
A DMFB is assumed to operate synchronously: a typical actuation frequency is 100 Hz, [25], i.e., it takes 10 ms to transport one droplet to a neighboring electrode. By considering the state of the system only at 10 ms intervals, the entire system can be discretized, as shown in Fig. 4(a): a droplet can either stay at rest (e.g., Droplet 2 at 20ms), or move to a neighboring electrode (e.g., Droplet 2 at 10ms); if both electrodes are activated, then the droplet will stretch into an ovular shape that covers both electrodes.

In actuality, the velocity of a droplet traveling across the surface of a DMFB is a quadratic function of the voltage applied to each electrode [15, 17]. Thus, discretizing droplet transport through synchronization assumes that the same voltage value is always applied to each electrode. In practice, this does not need to be the case; moreover, there is no reason to believe that synchronization yields optimal or near-optimal performance (i.e., minimum droplet transport time) or energy consumption; but, considering continuous voltages and velocities transforms the problem of droplet routing from the discrete to the continuous domain, which changes the nature of the problem and the algorithmic solutions that are required, as shown in Fig. 4(b).

This paper introduces *the first* droplet transport algorithm for DMFBs that considers continuous voltages applied to electrodes, and continuous, rather than discrete, droplet velocities, thereby dropping the implicit assumption of synchronized droplet transport in prior work. A continuous time, voltage-aware formulation of the droplet routing problem is introduced that tries to minimize energy consumption during droplet transport given a timing constraint. This approach yields significant reductions in energy consumption without adversely affecting droplet routing times. Even greater energy savings could potentially be achieved by relaxing the timing constraint as well.



(a) Constant Discrete Voltage/Velocities



(b) Continuous Voltages/Velocities

Fig. 4. Droplets 1 and 2 cross a DMFB from one side to another (white arrows indicate the relative velocity) for (a) a DMFB driven by a single voltage, where all droplets move at the same velocity; and (b) a DMFB driven by multiple continuous voltages where droplets are allowed to move at varying velocities.

Minimizing energy consumption during execution of an assay on DMFB is an important problem, especially for portable point-of-care applications. One representative example would be to perform diagnostics as part of a health care effort targeting rural areas in the third-world, where societal infrastructure is lacking, and battery lifetime is limited.

II. Related Work

The routing algorithm presented in this paper is compatible with *direct addressing* [17] and *active-matrix* [9, 15] DMFBs, which provide independent control over all electrodes; compilation flows targeting these devices are mature [1, 3, 11, 16, 18, 20-22, 25]. It is not compatible with *pin-constrained* [2] and *cross-referencing* [6, 24] DMFBs, which provide more restricted forms of control.

The proposed algorithm would work best with DMFBs that have integrated *capacitive-touch sensors* into the substrate, which provide precise information about the position of droplets during transport [14, 19]. Otherwise, it would be too difficult to determine when a droplet completes its route at a given voltage/velocity.

Droplet routing is NP-complete [1]. The vast majority of routers that have been published, to date, are polynomial-time heuristics which cannot guarantee optimality; exceptions include optimal algorithms based on *A* search* [1] and *integer linear programming* [24], and a recent *iterative improvement* algorithm based on *particle swarm optimization* [16].

Several heuristics divide droplet routing into two parts [3, 11, 18]: (1) *path planning*, which determines the route that each droplet takes from its source to its destination, assuming that droplets are routed one-by-one; and (2) *route compaction*, which converts a the path planning result into a set of concurrent routes while adhering to proper spacing rules and trying to minimize the route of the longest droplet; thus. The algorithms presented in this paper take a similar approach: path planning followed by route compaction; in fact, any path planning algorithm can be used. We modified the compactor to consider voltage assignment to electrodes and the resulting droplet velocity in the continuous domain.

One recent paper varies the voltages applied to electrodes in a DMFB to improve reliability [23]. Another throttles droplet velocity by varying the frequency of electrode actuation [12]. To the best of our knowledge, no other papers have looked into the issues of varying voltage assignment and/or droplet velocity. Our objective is to reduce energy consumption, which is synergistic with, but not directly related to reliability.

III. Continuous-Time Droplet Routing: Preliminaries

A. Context and Spacing Rules

We assume that the assay has already been scheduled and placed on the DMFB; in principle, any appropriate algorithm could be used to accomplish these two steps. Droplet routing is decomposed into a series of *sub-problems* that must be solved [22].

As shown in Fig. 5, droplets must obey spacing rules to prevent inadvertent merging [22], which were originally introduced for discrete droplet transport. The spacing rules define an *interference region (IR)* around each droplet; if one droplet enters the IR of another, then the two will merge; to ensure correctness, the router must prevent this from occurring. The IR for a droplet resting at position $p = (x, y)$ is the 3x3 sub-array surrounding p.

The interference region for a droplet that is moving from position $p_1 = (x, y)$ to an adjacent position, p_2, which is $(x \pm 1, y)$ or $(x, y \pm 1)$, is the 3x4 sub-array surrounding p_1 and p_2. The former is called the *static* constraint, and the latter is called the *dynamic* constraint, as shown in Fig. 5. Consider a droplet at position p_1 at time t in the discrete domain. If the droplet remains at position p_1 at time $t+1$, then the static constraint must be satisfied; if it moves to an adjacent position at time $t+1$, then the dynamic constraint must be satisfied. Given the respective positions of each droplet in a larger set as a function of time, it is straightforward to determine whether *any* pair of droplets violates the static or dynamic constraint, as the number of constraint checks is finite.

In the continuous domain, at time t, a droplet is either at rest in position p, or in transport from position p_1 to adjacent position p_2: the static constraint must be satisfied in the former case, while the dynamic constraint must be satisfied in the latter case.

Droplets undergoing transport must also maintain a 1-position margin of separation with all other ongoing assay operations, such as mixing and storage; the size and location of these respective operations are fixed for the duration of each routing sub-problem.

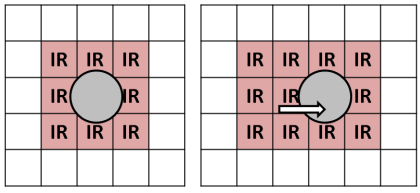


Fig. 5. The static interference region of a droplet at rest (left); the dynamic interference region of a droplet in motion (right).

B. Problem Statement

In a routing sub-problem, the input is a set of droplets to route, $D = \{d_1, d_2, \ldots, d_M\}$. The initial position of droplet d_i is called a *source*, denoted $src(d_i)$; d_i's final position is called a *sink*, denoted $sink(d_i)$. Each droplet must be routed from source to sink without violating static/dynamic interference constraints, while maintaining a 1-position separation with all ongoing assay operations. The objective is to minimize the completion time of all droplet routes.

C. Continuous-Time Routing Interference Constraints

The *path* taken by droplet d_i, is a sequence of positions taken by d_i along the route from $src(d_i)$ to $sink(d_i)$, and is denoted by the ordered sequence $P(d_i) = \langle p_{i,1}, p_{i,2}, \ldots, p_{i,N} \rangle$, where $p_{i,1} = src(d_i)$, $p_{i,N} = sink(d_i)$, and $p_{i,j+1}$ is adjacent to $p_{i,j}$ for $1 \leq j \leq N\text{-}1$.

In the discrete model, d_i spends $z_{i,j} \geq 1$ time-steps at position $p_{i,j}$ along the path, as it may be necessary to wait while other droplets pass by. The number of time-steps spent during routing is

$$Z(d_i) = \sum_{j=1}^{N} z_{i,j}. \quad (1)$$

In the continuous model, d_i will spend a continuous amount of time $t_{i,j} \geq 0$ at position $p_{i,j}$, during which the static interference constraint must be satisfied, followed by time $u_{i,(j,j+1)} > 0$ while moving from $p_{i,j}$ to adjacent position $p_{i,j+1}$, during which the dynamic interference constraint must be satisfied. The total amount of time spent during routing is

$$T(d_i) = \sum_{j=1}^{N} \left(t_{i,j} + u_{i,(j,j+1)}\right), \quad (2)$$

and the total time spent routing d_i from $src(d_i)$ to position $p_{i,j}$, is

$$U(d_i, p_{i,j}) = \sum_{k=0}^{j-1} \left(t_{i,k} + u_{i,(k,k+1)}\right), \quad (3)$$

where $t_{i,0} = u_{i,(0,1)} = 0$.

The continuous time interval at which d_i waits at position $p_{i,j}$ is therefore given by the pair

$$I(d_i, p_{i,j}) = \left[U(d_i, p_{i,j}), U(d_i, p_{i,j}) + t_{i,j}\right], \quad (4)$$

and the continuous time interval at which d_i is spent in transportation from position $p_{i,j}$ to $p_{i,j+1}$ is given by the pair

$$J(d_i, p_{i,j}, p_{i,j+1}) = \left(U(d_i, p_{i,j}) + t_{i,j}, U(d_i, p_{i,j+1})\right). \quad (5)$$

Thus, the timing information associated with each path is given by the ordered sequence of pairs

$$Q(d_i) = \langle I(d_i, p_{i,1}), J(d_i, p_{i,1}, p_{i,2}), \ldots, I(d_i, p_{i,N-1}), J(d_i, p_{i,N-1}, p_{i,N}) \rangle. \quad (6)$$

Fig. 6 shows an example in which a droplet moves continuously across six positions; it waits at the fourth position for six time-units, and then increases its velocity as it travels to the sixth.

If droplet d_i moves across the chip with positive velocity, then it remains at position $p_{i,j}$ for an instant, which is assumed to consume no time. Hence, in the continuous model, $I(d_i, p_{i,j}) = 0$ except when the droplet explicitly pauses at position $p_{i,j}$. A droplet may pause for two reasons: (1) to prevent the violation of an interference constraint; or (2) the droplet arrives at its respective sink and stops while waiting for other droplets to complete their routes.

Lemma 1. Any (static or dynamic) interference constraint between droplets d_i and d_j can be detected at continuous time $t \in \{t_1, t_2\}$ such that $(t_1, t_3) \in Q(d_i)$ and $(t_2, t_4) \in Q(d_j)$.

Proof: If d_i and d_j are initially placed at a location where a static constraint violation occurs (i.e., before droplets start to move), we are done, as $t = t_1 = t_2 = 0$. Now, assume that the constraint violation occurs at time $t > 0$. Either d_i or d_j must be in motion to cause the violation to occur; otherwise, it would have occurred at time $t' < t$. A constraint violation is an activation of an electrode at a position p^* inside the interference region of both d_i and d_j. The constraint violation occurs at the instant either droplet starts to move from an initial position p toward an adjacent position p^*. □

Fig. 7 illustrates the proof of Lemma 1.

Lemma 1 shows how to determine if constraint violations occur in continuous time. If there are K droplets in the system, traverse $Q(d_i)$ for $i = 1$ to K; for each interval $(t_k, t_l) \in Q(d_i)$, identify the exact position of each other droplet d_j at time t_k. If d_i is moving toward position p^* at time t_k, then a constraint violating occurs if p^* is in the static or dynamic interference region of d_j, depending on whether d_j is at rest or in motion.

The total number of interference checks that must be made to verify that the routes of all droplets are interference-free is

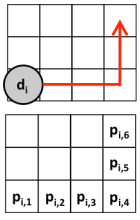
$$C = (K - 1) \sum_{i=1}^{K} |Q(d_i)|. \quad (7)$$

Since droplets move at different velocities, the constraint check is *antisymmetric*. For example, suppose that droplets d_i and d_j have a constraint violation at time t, such that $t_1 < t < t_4$ for $(t_1, t_4) \in Q(d_i)$ and $t_2 < t < t_3$ for $(t_2, t_3) \in Q(d_j)$, such that $t_1 < t_2 < t_3 < t_4$. The constraint violation is observable at time t_2, but not at time t_1, in accordance with Lemma 1. In contrast, interference constraint violations in the discretized, synchronized case are symmetric.

D. Simplifying Assumptions

In DMFB routing, droplets move horizontally or vertically, but not diagonally [22]; we make the same assumption here.

Given a droplet path $P(d_i)$, as defined in the preceding section, a *segment* is defined to be a maximum-length contiguous subsequence $\langle p_{i,j}, p_{i,j+1}, \ldots, p_{i,k} \rangle$ of $P(d_i)$ in which d_i travels in one direction (left, right, up, or down). Let $s_{i,j}$ denote the j^{th} segment along path $P(d_i)$, and let $S(d_i) = \langle s_{i,1}, s_{i,2}, \ldots, s_{i,n} \rangle$, $n \leq N$ (recall that N is the number of positions along the path) denote the decomposition of $P(d_i)$ into segments.



time

Transport/Wait Times	$t_{i,1}$	$u_{i,(1,2)}$	$t_{i,2}$	$u_{i,(2,3)}$	$t_{i,3}$	$u_{i,(3,4)}$	$t_{i,4}$	$u_{i,(4,5)}$	$t_{i,5}$	$u_{i,(5,6)}$	$t_{i,6}$
	0	3	0	3	0	3	6	2	0	2	11
Arrival Time at Each Position	$U(d_i, p_{1,2})$		$U(d_i, p_{1,3})$		$U(d_i, p_{1,4})$		$U(d_i, p_{1,5})$		$U(d_i, p_{1,6})$		$U(d_i, p_{1,6}) + t_{i,6}$
	3		6		9		17		19		30
	$I(d_i, p_{1,1})$	$J(d_i, p_{1,2})$	$I(d_i, p_{1,2})$	$J(d_i, p_{1,3})$	$I(d_i, p_{1,3})$	$J(d_i, p_{1,4})$	$I(d_i, p_{1,4})$	$J(d_i, p_{1,5})$	$I(d_i, p_{1,5})$	$J(d_i, p_{1,6})$	$I(d_i, p_{1,6})$
Q(di)	<[0, 0],	(0, 3),	[3, 3],	(3, 6),	[6, 6],	(6, 9)	[9, 15],	(15, 17),	[17, 17],	(17, 19),	[19, 30]>

Fig. 6. Example illustrating continuous time-domain droplet routing parameters: droplet d_i starts at position $p_{i,1}$ at time zero. It travels to $p_{i,4}$ at a velocity of 1/3 of a position per time-unit. It then waits at $p_{i,4}$ for 6 time-units, and travels to $p_{i,6}$ at a velocity of 1/2 of a position per time-unit. It then waits at $p_{i,6}$ for 11 additional time-units. The total time taken, including transport and waiting, is 30 time-units.

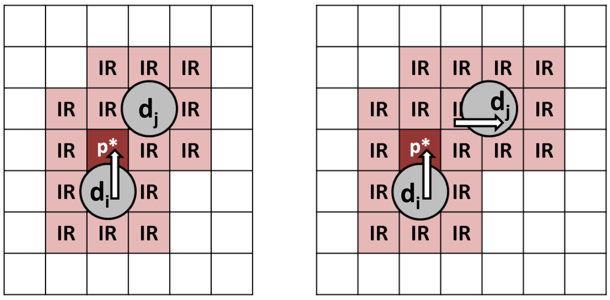


Fig. 7. Illustration of the proof of Lemma 1. Activating the electrode at position p^* transports droplet d_i into the interference region of droplet d_j; p^* is activated precisely at time instant t_k, which is the beginning of a time interval $(t_k, t_l) \in Q(d_i)$.

For example, suppose that droplet d_i travels from $src(d_i) = (1, 1)$ to $sink(d_i) = (4, 2)$ along path $P(d_i) = <(1, 1,), (2, 1), (3, 1), (4, 1), (4,2)>$; $P(d_i)$ can be decomposed into a horizontal segment $s_{i,1} = <(1, 1), (2,1), (3, 1), (4,1)>$ and a vertical segment $s_{i,2} = <(4,1), (4,2)>$, so $S(d_i) = <s_{i,1}, s_{i,2}>$. It is important to note that the subsequence $<(1, 1), (2,1), (3, 1)>$ is not considered to be a segment here, because it is *not* maximal length.

We assume that each droplet travels at a constant velocity along each segment, and that the droplet may only stop at the initial and final segment positions. This means that voltage selection is performed on the granularity of segments, but interference constraints must still be resolved on the granularity of positions in continuous time, as discussed in the preceding section.

E. Velocity and Energy Models

Continuous time-domain droplet routing introduces, a new degree of freedom: to select the voltage applied to each electrode from a continuous range, as opposed to turning the electrode on (at a constant voltage) or off; the router must still adhere to continuous time-domain interference constraints (Section III.C).

This subsection summarizes the equations used to estimate the droplet velocity and power consumption that results from applying a voltage to an electrode in an active-matrix DMFB [9, 15]. To avoid confusion with notation established in previous sections, we use words, rather than letters, to represent these quantities. Droplet velocity (mm/s) as a function of voltage is derived from existing data [15, Fig. 3(a)] using a least-squares quadratic approximation:

$$Velocity = 0.005 \times Voltage^2 + 0.0358 \times Voltage - 0.9103 \quad (8)$$

The energy consumption of each electrode depends on the voltage applied, the electrode's resistance, and the length of time that the voltage is applied:

$$Energy = Power \times Time = \frac{Voltage^2}{Resistance} \times Time \quad (9)$$

As noted previously, we assume that a droplet moves at a constant velocity along each segment; i.e., a constant voltage is applied to each electrode along the segment, and only one electrode (per droplet) is activated at any time.

F. Technology Parameters

The typical resistance of an electrode in an active-matrix DMFB is around $1G\Omega$, and the electrode pitch was reported to be *2.54mm* [15], which we take to be the length of each square in the DMFB array. We assume that there are minimum and maximum voltage levels (13 and 70 Volts respectively) for electrode actuation.

Voltages below the minimum cannot actuate droplet motion, and voltages above the maximum will physically degrade the chip. As an example, consider the voltage applied to transport droplet d_i along segment $s_{i,j}$; let $|s_{i,j}|$ denote the number of positions that comprise the segment, so the total distance to travel along this segment will be $2.54mm \times (|s_{i,j}| - 1)$. The droplet velocity (and thus, the total time traveled, which we will denote $\Delta T_{i,j}$) depends on the constant voltage level $V_{i,j}$ chosen for this segment.

The energy consumed to transport d_i along segment $s_{i,j}$, denoted by $E_{i,j}$, depends on $V_{i,j}$ and $\Delta T_{i,j}$ as well:

$$E_{i,j} = \frac{V_{i,j}^2}{1G\Omega} \times \Delta T_{i,j} \quad (10)$$

As per Eq. (8), $\Delta T_{i,j}$ also depends on $V_{i,j}$ (via the classic mechanical equation: *distance* = *velocity* × *time*). Thus, as long as droplet paths are determined a-priori, $E_{i,j}$ depends only on voltage.

We assume that 10 Volts are required to hold a droplet in-place; if no voltage is applied, the droplet will not be anchored and will drift aimlessly. Thus, a droplet at rest consumes energy regardless of whether a discrete or continuous routing model is used.

IV. Continuous-Time Droplet Routing Algorithms

Section III.A characterized the droplet routing problem; the objective was to finish all of the routes as quickly as possible, which is equivalent to minimizing the length (in time) of the longest route. This formulation of the droplet routing problem does not account for potential tradeoffs between droplet velocity and energy consumption of the device. Here, we assume that a time constraint is provided and the objective is to minimize energy consumption while meeting the timing constraint.

The algorithms that we introduce here employ the assumptions outlined in Section III.E. Our algorithm takes a two-part approach, in which droplet routing paths are computed a-priori, followed by a compaction step which enables concurrent droplet transport. Given the set of routing paths, compaction determines the sequence of timing intervals $Q(d_j)$ for each droplet d_j, as discussed in Section III.D. The compactor must prevent interference violations from occurring to ensure a legal routing solution. The voltage-aware formulation of the routing problem adds an additional constraint (timing) while defining a new optimization criterion (minimize energy) in addition to the basic criteria for an interference-free droplet route.

A. Compaction Strategy

Our implementation is based on an algorithm published by Roy et al. [18], which uses a Maze Router to compute droplet routes, and then applies a greedy heuristic to perform compaction. The heuristic sorts the droplets based on a priority function, and compact the routes one-by-one. A typical priority function, which we use in our implementation, is to prioritize the droplets by the length of their routes, such that droplets with the longest routes have the highest priority. A droplet d_i is *compacted* when the velocity along each segment in its route and the waiting time at each segment endpoint are known; i.e., once $Q(d_i)$ is computed.

In this algorithm, compacted routes are *never* revisited. When compacting droplet d_i, the compacted routes for all higher priority droplets impose constraints that must be satisfied to ensure a legal routing solution. If d_i's route crosses the compacted route of a higher priority droplet d_j, then the compactor must construct $Q(d_i)$ in a manner that prevents interference constraint violations involving any higher priority droplets from occurring; lower priority droplets are not given any consideration during compaction until they are routed themselves.

When compacting a droplet, the segments that compose the route are compacted one-by-one, in-order. To compact a segment along a route, the algorithm must choose a voltage (which, in turn, determines the droplet velocity) that leads to no interference constraint violations with any droplets whose routes are already compacted. The compaction process for the droplet completes when all segments along the route are successfully compacted.

B. Timing-Constrained Route Compaction Algorithm

This formulation of the routing problem includes a timing constraint, denoted T_C; all droplets must complete their routes in time no greater than T_C. The objective is to minimize the total energy consumption while doing so.

Droplets are processed for compaction, one-by-one, as described in the preceding subsection. The exact length $L(d_i)$ of the droplet path for droplet d_i is known, as it is the number of electrodes on the path multiplied by the electrode pitch (*2.54mm* in active-matrix technology [15]). The initial velocity for d_i is chosen to be $L(d_i)/T_C$, which is the minimum velocity required to meet the latency constraint, under the assumption that d_i does not pause along its path. The voltage level that produces this velocity can be determined using Eq. (8); if this voltage is lower than the minimum allowable voltage, then the minimum allowable voltage is chosen.

Starting at the first position in each segment the algorithm incrementally computes $Q(d_i)$ one electrode position at a time. As per the discussion in Section III.C, the compactor tests whether each advancement of $Q(d_i)$ at its chosen velocity violates an interference constraint each time that a new electrode is activated along the path. If a violation occurs, it is *not* possible to slow down d_i; otherwise, a timing violation would occur. Therefore, the only option is to increase d_i's velocity (by increasing the voltage) along the segment where the violation occurs. The increase should be the minimum necessary to suppress the constraint violation. Increasing the velocity along the segment will alter the timing intervals in $Q(d_i)$; therefore, the part of $Q(d_i)$ corresponding to the present segment being processed needs to be updated.

If a droplet is sped up along a segment, then it will arrive early at the destination if the initially computed velocity is used for the remainder of the route, and even earlier if its velocity increases along a future segment; in principle, the droplet could be slowed down along the remainder of the route in order to further reduce energy, as long as the timing constraint is not violated. Thus, whenever a droplet is sped up along a segment, the algorithm re-computes the minimum velocity required to complete the *remaining* route without violating the timing constraint. The remainder of the route is compacted using the voltage that corresponds to this newly computed velocity; if this voltage is lower than the minimum allowable voltage, then the minimum allowable voltage is used instead.

C. Limitations

In some cases, it may not be possible to find a compaction solution without exceeding the maximum allowable voltage. This is likely to occur if there exist a large number of droplets that need to be routed through a congested area of the DMFB. The only remedy for this situation is to re-schedule and re-place the assay, under the optimistic hope that the resulting routing sub-problems can be solved. The problem of unroutable routing sub-problems has been noted elsewhere [3, 18, 22, 25] and is not specific to the continuous time-domain formulation of the problem. We did not encounter any unroutable sub-problems in our experiments, although the theoretical possibility of their existence remains.

V. Simulation Results

A. Setup and Approach

We implemented the timing-constrained cyber-physical droplet routing algorithm in a publicly available open source framework for DMFB synthesis [8]. We modeled an active matrix DMFB [15], whose technology parameters are reported in Section III.F. We used a synthesis flow described in ref. [7] as our baseline; which includes Roy's algorithm for droplet routing [18]. We replaced the discrete compaction phase of Roy's algorithm with the continuous-time domain compactor described in Section IV.B, and compared the energy consumption with that of the baseline. We used a standard set of DMFB benchmarks that are publicly available and have been used in prior studies on DMFB synthesis and routing: polymerase chain reaction (PCR), multiplexed in-vitro diagnostics (In-vitro 1-5), and protein crystallization (ProteinSplit 1-4 and Protein).

Here, we focus on individual routing sub-problems in which multiple droplets are transported. For each routing sub-problem, we first route the droplets using the baseline routing algorithm using a constant voltage; the droplet transport velocity for a given voltage level is given by Eq. (8). We record the time t at which the last droplet finishes its route, along with the energy E_{Base} expended to transport all droplets, as modeled by Eq. (9). We then run the time-constrained droplet routing algorithm, described in Section IV, using t as a timing constraint; we record the energy expended to transport all droplets, denoted E_{TC}. We report the energy savings $E^* = (E_{Base} - E_{TC})/E_{Base}$ as a percentage.

Among the seven assays that we considered, PCR, Protein, and five variations of multiplexed in-vitro diagnostics, we identified 99 routing sub-problems of interest. We performed three comparisons per sub-problem, running the baseline routing algorithm with voltages of 30V, 50V, and 70V, yielding routing times denoted of t_{30V}, t_{50V}, *and* t_{70V}. We then ran the continuous time-domain routing algorithm using t_{30V}, t_{50V}, *and* t_{70V} as timing constraints, and report the energy savings E^*_{30V}, E^*_{50V}, and E^*_{70V}.

B. Results and Analysis

Fig. 8 reports the energy savings attained for all routing sub-problems at all three Voltage levels, and Fig. 9 concisely summarizes the results. The two key observations are: (1) for a given voltage level, the attainable energy savings varies significantly between routing sub-problems; and (2) greater energy savings are attainable at lower voltage levels.

The variation in energy savings across sub-problems is readily apparent in Fig. 9. The standard deviation in energy savings is equal to the average for the 30V experiments, and greater than the average for the 50V and 70V experiments. The reason for these high variations is not algorithmic inconsistency, but is actually an inherent property of the routing sub-problem instances. The experiment is set up so that the droplet that finishes last will take the same path and consume the same amount of energy in both the discrete and continuous-time models; whatever energy savings is accrued is due to other droplets slowing down by throttling the voltage. Droplets whose routes are significantly shorter than the longest route can travel at very low velocities without violating the timing constraint; thus, the greatest possible energy savings can be accrued for problem instances that have an inordinate number of droplets that fall into this category. In contrast, it is difficult to achieve significant energy savings for problem instances where all droplet routes have approximately equal length.

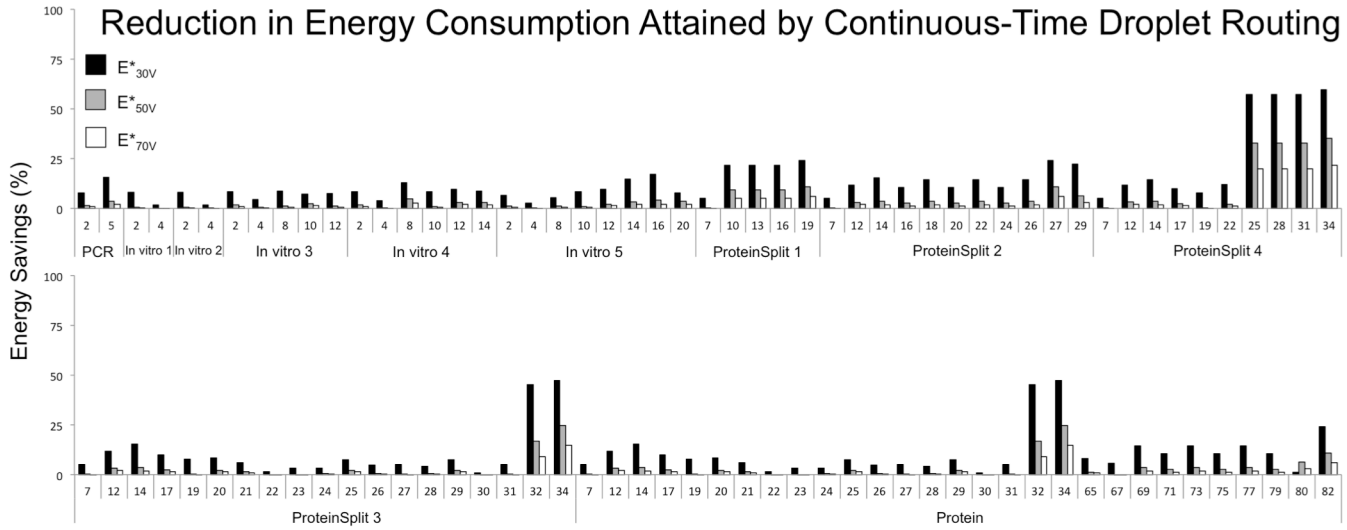


Fig. 8. Energy savings obtained by continuous-time droplet routing in comparison with discrete droplet routing at 30V, 50V, and 70V baselines. The experiments include 99 routing sub-problems; the ID number of each routing sub-problem is listed above each benchmark.

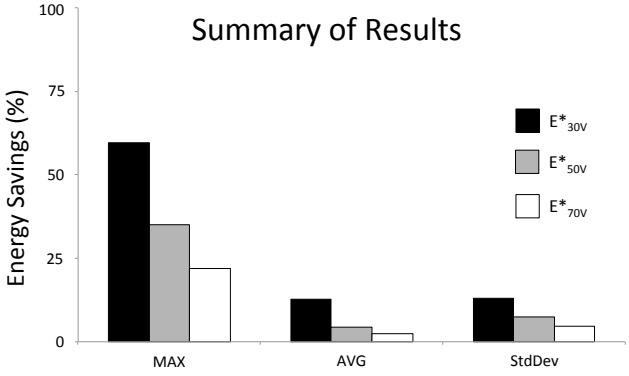


Fig. 9. Summary of the results reported in Fig. 8.

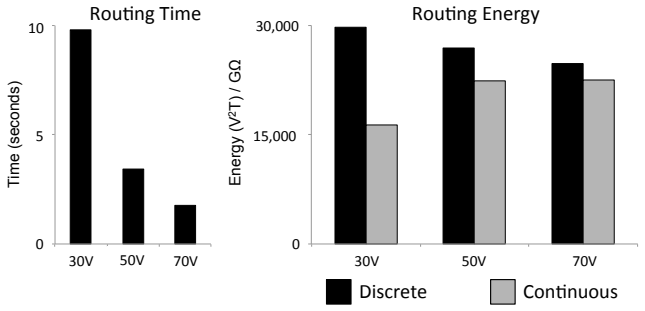


Fig. 10. Droplet routing times and energy consumption (as opposed to energy saving) for routing sub-problem 32 of ProteinSplit3.

Next, we address the issue of why the lower baseline voltage levels achieve greater energy savings than the higher voltage levels across the different routing sub-problems that were examined. As an example, Fig. 10 shows the droplet routing times and energy consumed for routing sub-problem 32 of ProteinSplit 3. The length of the longest droplet route at 30V is 2.9x longer than the length of the longest droplet route at 50V, and 5.6x longer than the length of the longest droplet route at 70V. As a consequence, there is a much greater window of time during which the compactor can slow down non-critical droplets to reduce energy at lower voltage levels.

Looking exclusively at the discrete routing model, increasing the baseline voltage yields a net energy savings because higher voltages yield higher droplet velocities, and shorter completion times. Thus, more energy is required to activate a sequence of electrodes at 30V for 9.8 seconds than to activate a sequence of electrodes at 50V for 3.4 seconds. Given the timing constraint provided by the discretized routing model at each voltage level, the continuous-time routing model reduces energy consumption; however, diminishing energy savings are observed at higher voltage levels due to shorter route completion times.

Altogether, these results demonstrate that varying the voltage applied to transport droplets at varying velocities in the continuous time-domain droplet routing model can lead to significant energy savings for favorable droplet routing sub-problems.

VI. Conclusion

This paper has developed *the first* algorithm by which synthesis tools can control droplet velocity and DMFB energy consumption during droplet transport by varying the voltage applied to electrodes. Both voltage and velocity are continuous quantities, so dealing with continuous, rather than synchronized and discrete, fluid flow creates new algorithmic challenges, especially to detect and prevent interference region constraint violations. Due to the discrete nature of the 2D electrode array, the interference regions themselves remain discrete, even though droplet transport is continuous in time. The proposed interference constraint detection algorithms leverages this observation to bound the number of distinct points in time at which constraints must be checked. This mechanism was integrated into a new continuous voltage-aware droplet routing algorithm, targeting activate-matrix actuation technology, which minimizes energy consumption while meeting timing constraints on droplet routes.

This initial foray into continuous droplet transport control is expected to launch future investigations on other aspects of DMFB synthesis that can be handled in the continuous domain. For example, it may be possible to leverage voltage assignment to explore trade-offs involving operation latency and energy consumption during scheduling.

The tradeoffs explored in this paper can be extended to account for different problem formulations such as minimizing droplet transport time given an energy budget, and routing wash droplets to reduce cross-contamination. This work promotes understanding of DMFBs as cyber-physical systems that control continuous physical quantities with a hybrid discrete-continuous architecture.

Acknowledgment

This work was supported in part by NSF Grant CNS-1035603. D. Grissom was supported by an NSF Graduate Research Fellowship. J. Fiske was supported by a UC LEADS summer internship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NSF.

References

[1] Böhringer, K. F. 2006. Modelling and controlling parallel tasks in droplet-based microfluidic systems. *IEEE Trans. CAD* 25, 2 (Feb. 2006), 329-339. DOI= http://dx.doi.org/10.1109/TCAD.2005.855958

[2] Chakrabarty, K. 2010. Design automation and test solutions for digital microfluidic biochips. *IEEE Trans. Circuits and Systems I: Regular Papers*, 57, 1 (Jan. 2010) 4-17. DOI= http://dx.doi.org/10.1109/TCSI.2009.2038976

[3] Cho, M., and Pan, D. Z. 2008. A high-performance droplet routing algorithm for digital microfluidic biochips. *IEEE Trans. CAD* 27, 10 (Oct. 2008) 1714-1724. DOI= http://dx.doi.org/10.1109/TCAD.2008.2003282

[4] Fair, R. B. 2007. Digital microfluidics: is a true lab-on-a-chip possible? *Microfluidics and Nanofluidics* 3, 3 (Jun. 2007), 245-281. DOI= http://dx.doi.org/10.1007/s10404-007-0161-8

[5] Fair, R. B., et al. 2007. Chemical and biological applications of digital-microfluidic devices. *IEEE Design and Test of Computers* 24, 1 (Jan.-Feb. 2007), 10-24. DOI= http://dx.doi.org/10.1109/MDT.2007.8

[6] Fan, S-K., Hashi, C., and Kim C-J. 2003. Manipulation of multiple droplets on N×M grid by cross-reference EWOD driving scheme and pressure-contact packaging. In *Proceedings of the 16th IEEE International Conference on Micro Electro Mechanical Systems* (Kyoto, Japan, January 13 - 23, 2003). MEMS '03, 694-697. DOI= http://dx.doi.org/10.1109/MEMSYS.2003.1189844

[7] Grissom, D., and Brisk, P. Fast online synthesis of generally programmable digital microfluidic biochips. In *Proceedings of the ACM/IEEE International Conference on Hardware Software Codesign and System Synthesis* (Tampere, Finland, October 07 - 12, 2012). CODES-ISSS '12, 413-422. DOI= http://dx.doi.org/10.1145/2380445.2380510

[8] Grissom, D., O'Neal, K., Preciado, B., Patel, H., Doherty, R., Liao, N., and Brisk, P. A digital microfluidic biochip synthesis framework. In *Proceedings of the IEEE/IFIP International Conference on VLSI and System-on-a-Chip* (Santa Cruz, CA, USA, October 07 - 10, 2012). VLSI-SOC '12, 177-182, DOI= http://dx.doi.org/10.1109/VLSI-SoC.2012.6379026

[9] Hadwen, B., et al. 2012. Programmable large area digital microfluidic array with integrated droplet sensing for bioassays. *Lab-on-a-Chip* 12, 18 (May. 2012), 3305-3313. DOI= http://dx.doi.org/10.1039/c2lc40273d

[10] Ho, T-Y., Chakrabarty, K., and Pop, P. 2011. Digital microfluidic biochips: recent research and emerging challenges. In *Proceedings of the ACM/IEEE International Conference on Hardware Software Codesign and System Synthesis* (Taipei, Taiwan, October 09 - 14, 2011). CODES-ISSS '11, 335-343. DOI= http://dx.doi.org/10.1145/2039370.2039422

[11] Huang, T-W., and Ho, T-Y. 2009. A fast routability- and performance-driven droplet routing algorithm for digital microfluidic biochips. In *Proceedings of the International Conference on Computer Design* (Lake Tahoe, CA, USA, October 04 – 07, 2009) ICCD '09, 445-450. DOI= http://dx.doi.org/10.1109/ICCD.2009.5413119

[12] Luo, Y., Chakrabarty, K., and Ho, T-Y. 2013. Design of cyberphysical digital microfluidic biochips under completion-time uncertainties in fluidic operations. In *Proceedings of the Design Conference* (Austin, TX, USA, June 02-06, 2013) DAC '13, article #44, DOI= http://dx.doi.org/10.1145/2463209.2488788

[13] Luo, Y., Chakrabarty, K., and Ho, T-Y. 2013. Error recovery in cyberphysical digital microfluidic biochips. *IEEE Trans. CAD* 32, 1 (Jan. 2013) 59-72. DOI= http://dx.doi.org/10.1109/TCAD.2012.2211104

[14] Murran, M. A., and Najjaran, H. 2012. Capacitance-based droplet position estimator for digital microfluidic devices. *Lab-on-a-Chip* 12, 11 (Mar. 2012) 2053-2059. DOI= http://dx.doi.org/10.1039/c2lc21241b

[15] Noh, J. H., Noh, J., Kreit, E., Heikenfeld, J., and Rack, P. D. 2012. Toward active-matrix lab-on-a-chip: programmable electrofluidic control enabled by arrayed oxide thin film transistors. *Lab-on-a-Chip* 12, 2 (Jan. 2012), 353-360. DOI= http://dx.doi.org/10.1039/c1lc20851a

[16] Pan, I., and Samanta, T. 2013. Efficient droplet router for digital microfluidic biochip using particle swarm optimizer. In *Proceedings of the SPIE 8760, International Conference on Communication and Electronics System Design 87601Z* (Jaipur, India, January 28, 2013) 87601Z-1 – 8760Z-10. DOI= http://dx.doi.org/10.1117/12.2012352

[17] Pollack, M. G., Shenderov, A. D., and Fair, R. B. 2002. Electrowetting-based actuation of droplets for integrated microfluidics. *Lab-on-a-Chip* 2, 2 (Mar. 2002), 96-101. DOI=http://dx.doi.org/10.1039/b110474h

[18] Roy, P., Rahaman, H., and Dasgupta, P. 2010. A novel droplet routing algorithm for digital microfluidic biochips. In *Proceedings of the 20th Great Lakes Symposium on VLSI* (Providence, RI, USA, May 16 - 18, 2010) GLSVLSI '10, 441-446. DOI= http://dx.doi.org/10.1145/1785481.1785583

[19] Shih, S. C. C., Fobel, R., Kumar, P., and Wheeler, A. R. 2011. A feedback control system for high-fidelity digital microfluidics. *Lab-on-a-Chip* 11, 3 (Feb. 2011) 535-540. DOI= http://dx.doi.org/10.1039/C0LC00223B

[20] Su, F., and Chakrabarty, K. 2008. High-level synthesis of digital microfluidic biochips. *ACM Journal on Emerging Technologies in Computing Systems* 3, 4 (Jan. 2008), article #16. DOI= http://dx.doi.org/10.1145/1324177.1324178

[21] Su, F., and Chakrabarty, K. 2006. Module placement for fault-tolerant microfluidics-based biochips. *ACM Trans. Design Automation of Electronic Systems* 11, 3 (Jul. 2006), 682-710. DOI= http://dx.doi.org/10.1145/1142980.1142987

[22] Su, F., Hwang, W., and Chakrabarty, K. 2006. Droplet routing in the synthesis of digital microfluidic biochips. In *Proceedings of Design Automation and Test in Europe* (Munich, Germany, March 06-10, 2006) DATE '06, 1-6. DOI= http://dx.doi.org/10.1109/DATE.2006.244177

[23] Yeh, S-H., Chang, J-W., Huang, T-W., and Ho, T-Y. 2012. Voltage-aware chip-level design for reliability-driven pin-constrained EWOD chips. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design* (San Jose, CA, USA, November 05-08, 2012). ICCAD '12, 353-360. DOI= http://dx.doi.org/10.1145/2429384.2429461

[24] Yuh, P-H., Sapatnekar, S. S., Yang, C-L., and Chang, Y-W. 2009. A progressive-ILP-based routing algorithm for the synthesis of cross-referencing biochips. *IEEE Trans. CAD* 28, 9 (Sep. 2009), 1295-1306. DOI= http://dx.doi.org/10.1109/TCAD.2009.2023196

[25] Yuh, P-H., Yang, C-L., and Chang, Y-W. 2008. BioRoute: a network-flow-based routing algorithm for the synthesis of digital microfluidic biochips. *IEEE Trans CAD* 27, 11 (Nov. 2008), 1928-1943. DOI= http://dx.doi.org/10.1109/TCAD.2008.2006140