2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Coordination of Droplets on Light-Actuated Digital Microfluidic Systems

Zhiqiang Ma and Srinivas Akella

***Abstract*— In this paper we explore the problem of coordinating multiple droplets in light-actuated digital microfluidic systems intended for use as lab-on-a-chip systems. In a light-actuated digital microfluidic system, droplets of chemicals are actuated on a photosensitive chip by moving projected light patterns. Our goal is to perform automated manipulation of multiple droplets in parallel on a microfluidic platform. To achieve collision-free droplet coordination while optimizing completion times, we apply multiple robot coordination techniques. We present a mixed integer linear programming formulation for coordinating droplets given their paths. This approach permits arbitrary droplet formations, and coordination of both individual droplets and batches of droplets. We then present a linear time stepwise approach for batch coordination of droplet matrix layouts.**

I. Introduction

Coordinating multiple robots in a shared workspace has attracted much attention in the robotics community [9], [16], [30]. A typical goal of coordination is to achieve collision-free and time-optimal robot motions. We focus here on application of robot coordination techniques to droplet manipulation on light-actuated digital microfluidic devices. Digital microfluidic systems (DMFS) are a class of lab-on-a-chip systems where discrete droplets are manipulated on an array of electrodes [7], [11], [27]. Droplet actuation occurs by electrowetting, where electrode activation increases droplet wetting of the surface. In addition to transport, mixing and splitting of droplets can be performed on DMFS [11]. This enables complex biochemical analyses to be performed in such lab-on-a-chip systems.

Recent research has led to light-actuated digital microfluidic (LADM) systems [3], [6], [22]–[24]. These devices are digital microfluidic systems where the lower substrate is a continuous photoconductive surface. Projection of light on the lower substrate effectively creates virtual electrodes in the illuminated regions. By moving the illumination regions, droplets can be moved anywhere on the microfluidic chips (as depicted in Figure 1) to perform multiple chemical or biological reactions in parallel.

Since droplets in LADM devices are not restricted to moving on a fixed set of electrodes as in traditional DMFS, LADM devices provide greater droplet motion freedom, the ability to variably change droplet sizes, and eliminate issues of wiring large numbers of electrodes. Droplet transport, generation, mixing, and separation operations can be performed with projected light patterns, and a large number of droplets can be manipulated in parallel. Hence proper droplet coordination is extremely important for LADM devices. For instance, droplet collisions can contaminate droplets and should be avoided except when mixing is intended. Our goal is to move droplets as quickly as possible to destinations without collisions. By modeling droplets as robots, we can achieve collision-free motions optimized to reduce completion time.

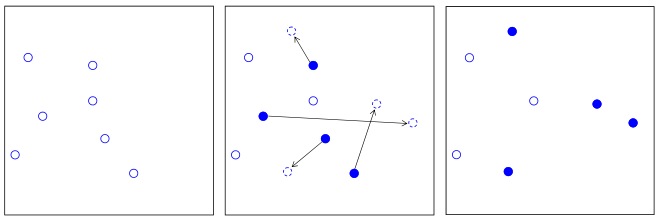


Fig. 1. Schematic snapshots of droplets in a light-actuated digital microfluidic (LADM) system. (a) Initial state. (b) Droplets to be moved are drawn shaded to represent the light source; arrows indicate the paths to their goal locations (dotted). (c) Goal state. Based on [24].

In this paper we focus primarily on creating matrix formations of droplets, similar to microwell layouts, for biological applications. To avoid collisions and minimize completion time, we use a mixed integer linear programming (MILP) approach [1] to schedule droplet starting times. We then present a linear complexity stepwise coordination algorithm for droplet matrix layouts.

II. Related Work

"Lab-on-a-chip" systems enable low cost, rapid, and portable diagnostic and biochemical systems. Digital microfluidic systems (DMFS) are a class of lab-on-a-chip systems where discrete droplets are manipulated on an array of electrodes [7], [11], [27]. In addition to transport, mixing and splitting of droplets can be performed on DMFS [11]. Droplet coordination and scheduling on DMFS has been previously explored [2], [10], [12], [13], [18].

An exciting new development is the actuation of droplets using light sources, such as lasers and more recently, lower-intensity light sources. Chiou et al. [4] first described the use of lasers to achieve optical actuation of droplets on DMFS electrodes, and demonstrated a full set of droplet functions: transport, generation, separation, and multidroplet manipulation [3]. Chiou et al. [5] introduced an optical image-driven dielectrophoresis technology that could manipulate

This work was partially supported by National Science Foundation Award IIS-1019160.

Z. Ma and S. Akella are with the Department of Computer Science, University of North Carolina at Charlotte, Charlotte, North Carolina 28223. zma5@uncc.edu, sakella@uncc.edu

cells and microparticles on a continuous photoconductive surface. Choi et al. [8] demonstrated a microparticle manipulation platform that uses an LCD screen. Park et al. [20] developed floating electrode optoelectronic tweezers using dielectrophoresis. They subsequently [21] achieved parallel droplet motion while lowering the intensity of the laser beam. Chiou et al. [6] developed an optoelectrowetting technique that used a continuous photosensitive layer, with no patterned electrodes, to achieve droplet actuation with lasers. Pei et al. [24] demonstrated light-actuated digital microfluidics for large-scale parallel manipulation of droplets using a projector light source of much lower optical intensity. They achieved this by using a high dielectric constant, low thickness dielectric material. They subsequently [23] demonstrated rapid droplet mixing using LADM. Park et al. [22] used a lateral field design to demonstrate droplet motion, mixing, and splitting using projectors and using LCDs.

There has been extensive research on motion planning for multiple robots; see [9], [15], [16] for overviews and [17], [26], [28], [30] for example approaches related to our work. However robot motion planning techniques have not been previously applied to droplet coordination on LADMs.

III. Background: Coordination of Multiple Robots with Specified Paths

Since our application involves multiple droplets moving in a shared workspace on a microfluidic device, we summarize our previous work [1] on coordinating multiple robots with specified paths and trajectories. Given a set of robots with specified paths and constant velocities, we can find the starting times for the robots such that the completion time for the set of robots is minimized and no collisions occur. We denote the ith robot by $\mathcal{A}_i$, and the time when robot $\mathcal{A}_i$ begins to move by t_i^{start}; this is to be computed.

A. Collision Zones

Assume robots $\mathcal{A}_i$ and $\mathcal{A}_j$ can collide. We define $\mathcal{A}_i(\gamma_i(\zeta_i))$ as the workspace that $\mathcal{A}_i$ occupies at path parameter value ζ_i along its path γ_i. The geometric characterization of this collision is

$$\mathcal{A}_i(\gamma_i(\zeta_i)) \cap \mathcal{A}_j(\gamma_j(\zeta_j)) \neq \emptyset.$$

$\mathcal{PB}_{ij}$ is the set of all points on the path of robot $\mathcal{A}_i$ at which $\mathcal{A}_i$ could collide with $\mathcal{A}_j$, and can be represented as a set of intervals

$$\mathcal{PB}_{ij} = \{[\zeta_{is}^k, \zeta_{if}^k]\} \tag{1}$$

where each interval is a *collision segment*, and s and f refer to the start and finish of the kth collision segment. We refer to the corresponding pairs of collision segments of the two robots as *collision zones*, denoted by $\mathcal{PI}_{ij}$. The set of collision zones, which describe the geometry of possible collisions, can be represented as a set of ordered pairs of intervals:

$$\mathcal{PI}_{ij} = \{< [\zeta_{is}^k, \zeta_{if}^k], [\zeta_{js}^k, \zeta_{jf}^k] >\}. \tag{2}$$

For scheduling the robots, we must describe the timing of the collisions. Given the speed of the robots, the set of times at which it is possible that robot $\mathcal{A}_i$ could collide with robot $\mathcal{A}_j$ can be easily computed.

We refer to each interval as a *collision-time interval*. Let T_{is}^k (respectively T_{if}^k) denote the time at which $\mathcal{A}_i$ starts (resp. finishes) traversing its k^{th} collision segment if $t_i^{start} = 0$. For the two robots $\mathcal{A}_i$ and $\mathcal{A}_j$, we denote the set of all collision-time interval pairs by $\mathcal{CI}_{ij}$, and represent it as a set of ordered pairs of intervals

$$\mathcal{CI}_{ij} \quad = \quad \{< [T_{is}^k, T_{if}^k], [T_{js}^k, T_{jf}^k] >\} \tag{3}$$

If $[T_{is}^k, T_{if}^k]$ and $[T_{js}^k, T_{jf}^k]$ do not overlap, then the two robots cannot be in the k^{th} collision zone simultaneously, and therefore no collision will occur in this collision zone.

B. Sufficient Conditions for Collision-free Scheduling

Therefore the sufficient condition for collision avoidance amounts to ensuring that there is no overlap between the two intervals of any collision-time interval pair for the two robots. If $[T_{is}^k + t_i^{start}, T_{if}^k + t_i^{start}] \cap [T_{js}^k + t_j^{start}, T_{jf}^k + t_j^{start}] = \emptyset$ for every collision-time interval pair, then no collision can occur (Figure 2). This sufficient condition leads to an optimization problem: *Given a set of robots with specified trajectories, find the starting times for the robots such that the completion time for the set of robots is minimized and no two intervals of any collision-time interval pair overlap.*

C. Collision-free Coordination of Multiple Robots

We developed a mixed integer linear programming (MILP) formulation for coordinating the motions of multiple robots with specified trajectories, where only the start times can be modified [1]. Let T_i be the time required for robot $\mathcal{A}_i$ to traverse its entire trajectory when starting at time $t_i^{start} = 0$. The maximum time for robot $\mathcal{A}_i$ to complete its motion, $t_i^{start} + T_i$, is its *completion time*. The completion time for the set of robots, $t_{complete}$, is the time when the last robot completes its task.

Consider coordination of a pair of robots $\mathcal{A}_i$ and $\mathcal{A}_j$ with specified trajectories. Ensuring the robots are not in their kth collision zone at the same time yields a disjunctive "or" constraint that can be converted to an equivalent pair of constraints using an integer zero-one variable δ_{ijk} and M, a large positive number [29]. When robot $\mathcal{A}_i$ enters the collision zone first, the constraint $t_i^{start} + T_{if}^k < t_j^{start} + T_{js}^k$ holds and $\delta_{ijk} = 0$, and when robot $\mathcal{A}_j$ enters the collision

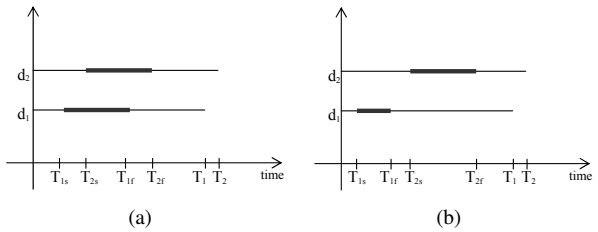


Fig. 2. Timelines for two droplets. The bold lines correspond to the collision-time intervals. (a) Collision can occur. (b) Collision will not occur.

zone first, the constraint $t_j^{start} + T_{jf}^k < t_i^{start} + T_{is}^k$ holds and $\delta_{ijk} = 1$.

Let N be the number of robots. Let N_{ij} denote the number of collision-time interval pairs for robots $\mathcal{A}_i$ and $\mathcal{A}_j$, i.e., $N_{ij} = |\mathcal{CI}_{ij}|$. We wish to minimize the completion time while ensuring the robots are not in their shared collision zones at the same time. A collision-free solution for this coordination task is given by the MILP formulation [1]:

$$
\begin{aligned}
&\text{Minimize } t_{complete} \\
&\text{subject to} \\
&t_{complete} - t_i^{start} - T_i \geq 0,\ 1 \leq i \leq N \\
&t_i^{start} + T_{if}^k - t_j^{start} - T_{js}^k - M\delta_{ijk} \leq 0 \\
&t_j^{start} + T_{jf}^k - t_i^{start} - T_{is}^k - M(1-\delta_{ijk}) \leq 0 \\
&\quad \text{for all } < [T_{is}^k, T_{if}^k], [T_{js}^k, T_{jf}^k] > \in \mathcal{CI}_{ij} \\
&\quad \text{for } 1 \leq i < j \leq N \\
&t_i^{start} \geq 0,\ 1 \leq i \leq N \\
&\delta_{ijk} \in \{0,1\},\ 1 \leq i < j \leq N,\ 1 \leq k \leq N_{ij}.
\end{aligned}
\tag{4}
$$

This MILP formulation can be directly applied to the coordination of droplets moving on known paths at constant speeds to achieve arbitrary layouts.

IV. Coordinating Droplets for Matrix Layouts

A. Droplet Matrix Layouts

Biochemists often need to perform a large number of tests in parallel (e.g., using microwell plates) so the conditions for each test can be varied. For example, they may want to quantify the effect of differing reagent concentrations on the outcome of a reaction. A matrix layout of droplets created by mixing droplets obtained from a set of column dispense stations and row dispense stations, each of which contains a particular chemical of a specified concentration, is suitable for such testing (Figure 3). Such experiments are well suited for execution on LADM devices [25].

Assume there are m row dispense stations on the left and n column dispense stations on the top to create an $m \times n$ matrix. Each entry (i, j) in the droplet matrix includes two droplets, each extracted from the left (ith row) and the top (jth column) dispense stations. A sketch of a 2×3 matrix is shown in Figure 3. The matrix entry locations are implicitly defined by the dispenser locations. We select the paths for the droplets to be the grid lines of the matrix, as in Figure 3. Each grid line starts from the edge of the corresponding dispense station and extends perpendicular to the dispense station. There is a region of feasible locations for each entry, which depends on the grid line locations. We select the grid lines to start from the center point of the edges. The subsequent step is to merge and mix the two droplets at each entry. Since a mixing operation can be performed in fixed time [23], we do not consider it while solving the coordination problem.

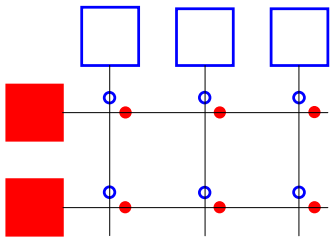


Fig. 3. An example 2×3 droplet matrix. Blue and red squares are column and row droplet dispense stations, and circles are droplets. Droplet paths are indicated by thin lines. A droplet's appearance indicates its source. Paired droplets at each grid entry will be merged for mixing.

We analyze two types of droplet matrices: *uniform grid matrices*, where the distance intervals between two adjacent entries along any row or column are the same, and *non-uniform grid matrices*, where the distance between two adjacent rows or columns can be arbitrary. See example uniform and non-uniform grid matrices in Figure 4 and Figure 5 respectively.

B. Coordination on Droplet Matrix

The objective is to form the droplet matrix as soon as possible while avoiding collisions. We now analyze the parallel motion of droplets and introduce multiple approaches to achieve this objective. We first state the *droplet matrix coordination* problem: *Given m dispense stations on the left and n dispense stations on the top, create a droplet matrix with $m \times n$ entries, and minimize the completion time while avoiding droplet collisions.* A matrix entry (i, j) consists of a droplet from the ith row dispense station and a droplet from the jth column dispense station. We assume all droplets move at the same constant velocity. One solution is to coordinate individual droplets using the MILP formulation when building the matrix. In addition, we propose two batch coordination strategies. We first discuss the batch coordination strategies.

C. Batch Coordination

Here we propose to move droplets in batches, filling one whole column or one whole row simultaneously. Each batch consists of one row or column of droplets extracted from the dispense stations at the same time. Temporary stations (the dotted circles in Figure 4) are an extra column or row of stations we specify next to the dispense stations. Each newly extracted batch moves simultaneously to the temporary stations. We assume that once a batch of droplets leaves its temporary station, it will continue moving without stopping until it reaches its destination row or column. A new batch is generated as soon as the current batch leaves the temporary stations. Droplet matrices can be classified into two types, uniform grid and non-uniform grid, based on column and row spacing. We now analyze them separately.

1) Uniform Grid: Here the distance intervals between two adjacent entries along any row or column are the same, as in Figure 4. We assume the speed of all droplets is fixed and equal, and therefore travel time intervals are identical.

Our solution moves batches of droplets to populate the farthest entries first. To avoid collisions, assume it is allowed to have a slight lag time T_l at the temporary stations on the side with more dispense stations, e.g., if $m < n$, let the lag be on the top, otherwise let the lag be on the left. To be safe, T_l can be defined to equal twice the diameter of the droplet

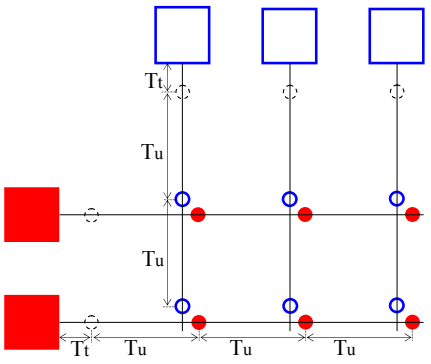


Fig. 4. An example 2×3 uniform grid droplet matrix. Dotted circles indicate temporary stations.

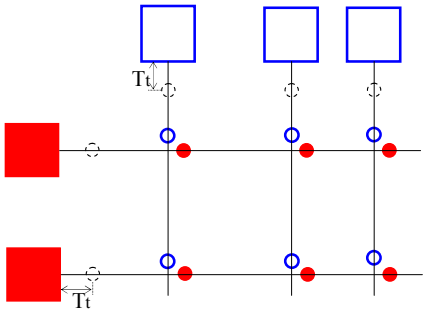


Fig. 5. An example 2×3 non-uniform grid droplet matrix.

divided by its speed. Each matrix entry contains two stations, one for the droplet from the top and one for the droplet from the left. We select the entry station locations to be vertically and horizontally offset to avoid a droplet at an entry station from blocking the motion of other droplets through the entry. Figure 4 shows an example with 2×3 dispense stations. A collision will occur at entry $(1, 1)$ if the first batch from the top and first batch from the left start to move at the same time. The lag time mentioned above avoids such collisions.

We compute the completion time for the above motion strategy. Let the time taken for extracting one droplet from a dispense station be T_e and the travel time from a dispense station to its corresponding temporary station be T_t. Assume the time interval from the temporary station to the first entry is the same as the interval between two adjacent entries T_u. Since different batches could move simultaneously and assuming $m \leq n$, the completion time $t_{complete}$ is

$$\begin{cases} T_e + T_t + \max\{mT_u + T_l, nT_u\}, & \text{if } T_u > T_e + T_t \\ \max\{m(T_e + T_t) + T_l, n(T_e + T_t)\} + T_u, & \text{otherwise.} \end{cases} \tag{5}$$

If $T_u > T_e + T_t$, the droplet batch from the top reservoirs to the farthest columns will take the longest time, $mT_u + T_e + T_t + T_l$, among all batches from the top. Similarly, the longest movement time from the left will be $nT_u + T_e + T_t$. When $T_u \leq T_e + T_t$, a similar analysis applies.

The completion time in Equation 5 can be computed in constant time. This eliminates the need for the MILP formulation for batch coordination on uniform grids.

2) Non-uniform Grid: Here the distance between two adjacent rows or columns can be arbitrary, as in the example grid of Figure 5. The batch movement strategy is similar to the uniform case. Start to generate another batch, as soon as one batch leaves the temporary stations. To avoid collisions, we permit a start time delay (computed from the MILP formulation discussed below) at temporary stations for corresponding batches.

Let b_{ir} be the droplet batch extracted from the top dispense stations for the ith row and b_{jc} be the droplet batch extracted from the left dispense stations for the jth column. Let T_{ir} be the travel time of b_{ir} from the temporary stations to its goal row. Similarly define T_{jc} for b_{jc}. If there is no collision, different batches can move simultaneously and the completion time $t_{complete}$ is

$$\begin{aligned} &\max_{i,j}\{T_e + T_t + \max\{T_{ir}, T_{jc}\}, \\ &\quad \max_{i,j}\{i(T_e + T_t) + T_{ir}, j(T_e + T_t) + T_{jc}\}\}, \\ &\quad \text{where } i \in \{1, 2 \ldots, m\} \text{ and } j \in \{1, 2 \ldots, n\}. \end{aligned} \tag{6}$$

Equation 6 computes the largest completion time of the droplets from the left and top dispense stations in different situations.

More typically, collisions can occur and so we formulate the problem as an MILP coordination problem that minimizes the completion time while ensuring collision-free motion. Since all droplets in a batch move simultaneously, the coordination objects are now the $m + n$ batches (rather than $2mn$ droplets).

Let t_{ir}^{start} be the start time of batch b_{ir}, and similarly, t_{jc}^{start} for b_{jc}. Given a pair of batches, the number of collisions k depends on the possible collisions caused by the droplets in each batch. For an $m \times n$ matrix, any pair b_{jc} and b_{ir} has $j(i-1)$ potential collision zones (b_{1r} does not cross any other column batches.). So the matrix has a total of $\sum_{i=1}^{m}\sum_{j=1}^{n} j(i-1) = \frac{mn(m-1)(n+1)}{4}$ potential collision zones. The MILP formulation for batch coordination is:

$$\begin{aligned} &\text{Minimize } t_{complete} \\ &\text{subject to} \\ &t_{complete} - T_e - T_t - t_{ir}^{start} - T_{ir} \geq 0,\ 1 \leq i \leq m \\ &t_{complete} - T_e - T_t - t_{jc}^{start} - T_{jc} \geq 0,\ 1 \leq j \leq n \\ &t_{ir}^{start} - t_{(i+1)r}^{start} \geq T_e + T_t, 1 \leq i \leq m-1 \\ &t_{jc}^{start} - t_{(j+1)c}^{start} \geq T_e + T_t, 1 \leq j \leq n-1 \\ &t_{ir}^{start} + T_{ir}^{kf} - t_{jc}^{start} - T_{jc}^{ks} - M\delta_{irjc}^{k} \leq 0 \\ &t_{jc}^{start} + T_{jc}^{kf} - t_{ir}^{start} - T_{ir}^{ks} - M(1 - \delta_{irjc}^{k}) \leq 0 \\ &\quad \text{for all } < [T_{ir}^{ks}, T_{ir}^{kf}], [T_{jc}^{ks}, T_{jc}^{kf}] > \in \mathcal{CI}_{irjc} \\ &\quad \text{for } 1 \leq i \leq m \text{ and } 1 \leq j \leq n \\ &\delta_{irjc}^{k} \in \{0, 1\},\ t_{ir}^{start} \geq 0 \text{ and } t_{jc}^{start} \geq 0 \\ &1 \leq i \leq m \text{ and } 1 \leq j \leq n. \end{aligned} \tag{7}$$

δ_{irjc}^{k} is a binary zero-one variable and M is a large positive constant. The third and fourth inequalities represent the filling-farther-entries-first constraint. These two inequalities mean batches going to farther entries are extracted at least

$T_e + T_t$ prior to batches for their nearer neighbors. In computing the collision interval, we define the collision interval as $[t - t_{safety}, t + t_{safety}]$, where t_{safety} is a predefined safety time that ensures that one droplet leaves the collision zone before another one starts to enter.

D. Stepwise Coordination

Since the MILP formulation is NP-hard and has worst-case exponential computational complexity [19], [29], we have developed a stepwise coordination method with a substantially lower computational complexity. This batch approach is most suitable for non-uniform grids with a large number of rows and/or columns; while it is applicable to uniform grids also, optimal solutions for them can be obtained as discussed in Section IV-C.1.

The move procedure is divided into *steps*. The number of steps for a general case is $\max\{m, n\}$. For a 2×3 matrix example, the total number of steps is 3 (Figure 6). The basic rule is still to fill farthest entries first and move droplets in batches. In each step, each movable batch moves from its current location to its next destination (i.e., the next entry location on its motion path). The following step begins only after all moving batches have reached their next destinations. If some batches arrive at their next destinations earlier than others, they have to wait until all batches complete motion for the current step.

Stepwise coordination avoids collisions due to the horizontal and vertical location differences of the stations at each entry and the safety zone we design to avoid collisions (Figure 7). There is at most one pair of droplets, one from the top and the other from the left, present in the safety zone at the same time. The distance between consecutive entries must be larger than the corresponding width of the safety zone, or the matrix formulation is invalid. Figure 7 depicts one matrix entry, its safety zone (drawn dotted), and its corresponding dispense stations. When blue and red droplets

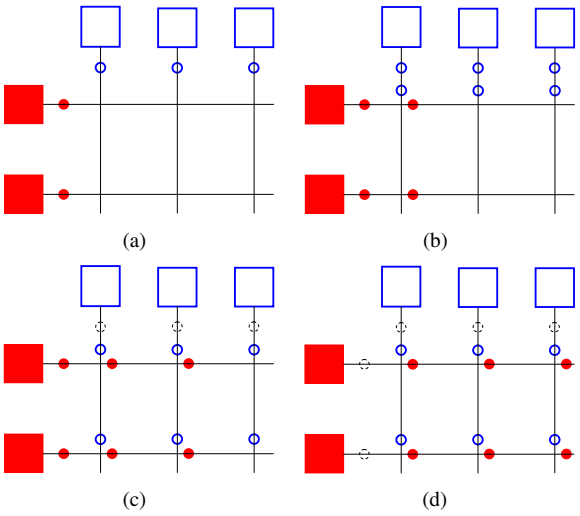


Fig. 6. Stepwise coordination for the 2×3 matrix example. Snapshots (a), (b), (c), and (d) are of the initial state, and after the first, second, and third steps respectively.

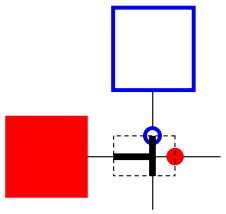


Fig. 7. Safety zone and entry stations for stepwise coordination.

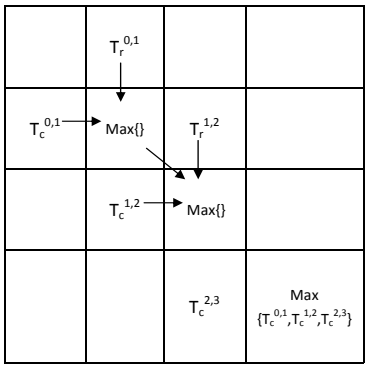


Fig. 8. Table for computing completion time using stepwise coordination. In the diagonal entries, Max{} returns the maximum value of input passed from the tails of the arrows.

move to their stations, no collision can occur since their paths do not cross. The vertical dimension of the safety zone is at least $2\sqrt{2}D$, where D is the droplet diameter, and is equal to the bold black horizontal segment. Thus when droplets leave the stations, the blue droplet cannot collide with an incoming red droplet from the left. If a collision occurred, the incoming red droplet must have been in the safety zone before the previous red droplet left the safety zone, which violates the one-pair-of-droplets rule.

We now analyze the movement steps and completion time. Let b_{ir} be the batch starting from top temporary stations heading to the ith row entries and b_{jc} be the batch from the left temporary stations to the jth column entries. Let $T_r^{p,q}$ represent the travel time from row p to row q for b_{ir}, and $T_c^{p,q}$ be the time for b_{jc} from column p to column q; temporary stations have an index of 0. In Figure 6(a), b_{2r} and b_{3c} are extracted. In the first step, the next destinations of b_{2r} and b_{3c} are row 1 and column 1 respectively. Therefore, the first step takes $\max\{T_r^{0,1}, T_c^{0,1}\}$ to complete. The second step illustrated in Figure 6(b) is a little more complex. It includes the movement of b_{1r} to row 1, b_{2r} to row 2, b_{2c} to column 1, and b_{3c} to column 2. The travel time is $\max\{T_r^{1,2}, T_c^{1,2}, \max\{T_r^{0,1}, T_c^{0,1}\}\}$. In step 3, only batches b_{1c}, b_{2c}, and b_{3c} from the left move, with a maximum travel time of $\max\{T_c^{0,1}, T_c^{1,2}, T_c^{2,3}\}$. The total completion time is the sum of T_e, T_t, and the travel times for the three steps.

Building a table to record the costs of the steps helps us work out the completion time. Figure 8 shows the tridiagonal matrix table for the above example. The lower band records $T_c^{p,q}$, the travel time between columns; the upper band records the travel time between rows $T_r^{p,q}$. The travel time

of each step is computed along the diagonal. For an $m \times n$ matrix, the computational complexity of filling out the table is $\mathcal{O}(m+n)+\mathcal{O}(\max(m,n))$, far less than the exponential complexity of MILP coordination. We introduce a general formulation to represent the algorithm to calculate the step times. For a matrix of dimension $m \times n$, assuming $m < n$, the sth step time t_s is

$$t_s = \begin{cases} \max\{T_r^{0,1}, T_c^{0,1}\} & s = 1, \\ \max\{T_r^{p,q}, T_c^{p,q}, t_{s-1}\} & 2 \le s \le m, \\ \max\{T_c^{0,1}, \ldots, T_c^{s-1,s}\} & m < s < n. \end{cases} \quad (8)$$

Conversely, if $m > n$, the third equation of Equation 8 becomes $\max\{T_r^{0,1}, \ldots, T_r^{s-1,s}\}, n < s < m$. The total completion time, therefore, equals $T_e + T_t + \sum_s t_s$.

All the movement strategies discussed above focus on batch movement. In Section IV-E below, we remove the batch constraint and coordinate droplets individually.

E. Individual Droplet Coordination

Individual droplet coordination is a direct application of the MILP formulation of Section III-C. Assume that once a droplet leaves its temporary station, it does not stop until the goal row or column is reached. We define the droplet going to the (i, j) entry from the left dispense station as d_{jcir}, and the droplet going to the same entry from the top dispense station as d_{irjc}. The droplet d_{jcir} could collide with d_{qrpc}, where $q > i$ and $p \le j$, so the total number of collision zones d_{jcir} has is $j(n-i)$. Therefore the total number of collision zones (and the number of binary variables) is $\sum_{i=1}^{m}\sum_{j=1}^{n} j(n-i) = \frac{nm(m-1)(n+1)}{4}$. We solve the MILP of Equation 4, with a slight modification to ensure successive droplets from a dispenser do not collide.

F. Implementation

We have implemented our coordination strategies on several examples. IBM ILOG CPLEX Optimizer [14] was used to solve the MILP problems. Consider the 5×5 droplet matrix shown in Figure 9. Let the diameter of the droplets be 0.5 mm. The maximum speed achieved on an LADM device is 2 cm/s [24]; we assume the speed of droplets is fixed at 1 cm/s. The intervals between entries are indicated in Figure 9. We draw the timelines in Figure 10(a). The bold lines are possible collision time intervals ($2t_{safety}$); their length is 0.1s. We formulate the MILP problem for this matrix based on Equation 7. Let $T_e + T_t$ equal 0.5s. The coordination result is demonstrated in Figure 10(b). CPLEX takes 0.038s to solve the problem on a 2.53 GHz Intel Xeon E5540 CPU with 12 GB of RAM. The completion time is 9.5s, which is the lower bound for this specific problem and implies the optimum result was obtained. Coordination results for several non-uniform droplet matrices are shown in Table I.

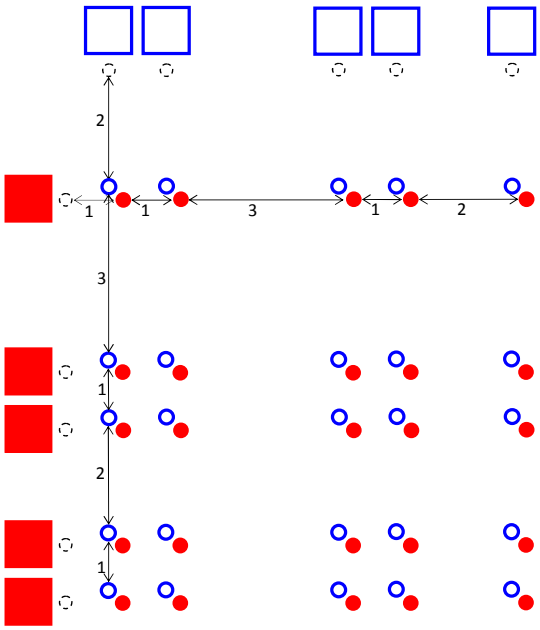
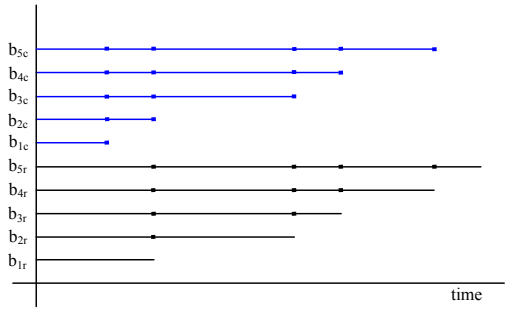
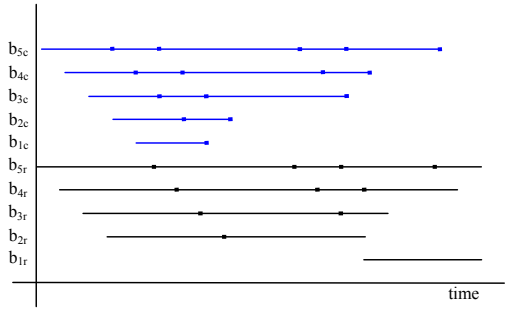


Fig. 9. A 5×5 droplet matrix layout. Numbers on the first row and column are the time intervals (in seconds) for a speed of 1 cm/s.



(a)



(b)

Fig. 10. Timelines for each batch of droplets for the 5×5 example. Bold lines are possible collision time intervals. (a) Timelines before coordination. (b) Timelines after coordination.

V. Conclusion

We presented algorithms for droplet coordination on light-actuated digital microfluidic devices. Mixed integer linear programming (MILP) formulations can provide collision-free coordination for a broad class of droplet coordination problems. We focused on droplet matrix layouts and presented batch and individual motion strategies for them. We also presented a stepwise batch coordination strategy with complexity linear in the number of batches. While it does not guarantee optimal solutions, it provides a low computational complexity algorithm that can solve large problem sizes.

Our work divided the droplet coordination problem into

TABLE I

COMPLETION TIMES FOR INDIVIDUAL COORDINATION, BATCH COORDINATION, AND STEPWISE COORDINATION ON MULTIPLE DROPLET MATRICES.

Matrix size	Individual			Batch			Stepwise
	Completion Time (sec)	Execution Time (sec)	No. of Variables	Completion Time (sec)	Execution Time (sec)	No. of Variables	Completion Time (sec)
2×3	5.5	0.014	6	5.5	0.012	6	7.5
4×6	9.5	0.021	126	9.5	0.023	126	17.5
8×12	18.5	0.18	2184	18.5	0.20	2184	35.5
5×5	9.5	0.03	150	9.5	0.038	150	14.5
10×10	18.5	0.37	2475	18.5	0.43	2475	29.5
15×15	29.5	14.48	11025	29.5	29.22	11025	44.5

two sub-problems, path planning and scheduling. Finding the optimal solutions to these subproblems does not guarantee the combination of these two solutions is the optimal solution to the original problem. Therefore generating better solutions when they exist is one direction for future research. In addition to interfacing our algorithms with LADM devices, exploring a broader set of droplet movement patterns, permitting wait times and varying droplet speeds, and handling cases when the number of dispense stations does not match the number of rows and columns of the droplet matrix are directions for future work.

ACKNOWLEDGMENTS

Thanks to Ming Wu and Shao Ning Pei for introducing us to this problem, and Vasanth Shekar for helpful discussions.

REFERENCES

[1] S. Akella and S. Hutchinson. Coordinating the motions of multiple robots with specified trajectories. In *IEEE International Conference on Robotics and Automation*, pages 624–631, Washington, DC, May 2002.

[2] K. F. Böhringer. Modeling and controlling parallel tasks in droplet-based microfluidic systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(2):334–344, Feb. 2006.

[3] P. Y. Chiou, Z. Chang, and M. C. Wu. Droplet manipulation with light on optoelectrowetting device. *Journal of Microelectromechanical Systems*, 17(1):133–138, 2008.

[4] P. Y. Chiou, H. Moon, H. Toshiyoshi, C. J. Kim, and M. C. Wu. Light actuation of liquid by optoelectrowetting. *Sensors and Actuators A: Physical*, 104(3):222–228, 2003.

[5] P. Y. Chiou, A. T. Ohta, and M. C. Wu. Massively parallel manipulation of single cells and microparticles using optical images. *Nature*, 436(7049):370–372, 2005.

[6] P. Y. Chiou, S. Y. Park, and M. C. Wu. Continuous optoelectrowetting for picoliter droplet manipulation. *Applied Physics Letters*, 93(22):221110–221110–3, 2008.

[7] S. K. Cho, H. Moon, and C.-J. Kim. Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits. *Journal of Microelectromechanical Systems*, 12(1):70–80, Feb. 2003.

[8] W. Choi, S.-H. Kim, J. Jang, and J.-K. Park. Lab-on-a-display: a new microparticle manipulation platform using a liquid crystal display (LCD). *Microfluidics and Nanofluidics*, 3:217–225, 2007.

[9] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[10] J. Ding, K. Chakrabarty, and R. B. Fair. Scheduling of microfluidic operations for reconfigurable two-dimensional electrowetting arrays. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(12):1463–1468, Dec. 2001.

[11] R. B. Fair, V. Srinivasan, H. Ren, P. Paik, V. K. Pamula, and M. G. Pollack. Electrowetting-based on-chip sample processing for integrated microfluidics. In *IEEE International Electron Devices Meeting*, pages 32.5.1–32.5.4, Washington, D.C., USA, Dec 2003.

[12] E. J. Griffith and S. Akella. Coordinating multiple droplets in planar array digital microfluidic systems. *International Journal of Robotics Research*, 24(11):933–949, Nov. 2005.

[13] M. Gupta and S. Akella. A scheduling and routing algorithm for digital microfluidic ring layouts with bus-phase addressing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3144–3150, San Diego, CA, Oct. 2007.

[14] IBM. IBM ILOG CPLEX Optimizer version 12.2.

[15] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, 1991.

[16] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[17] S. M. LaValle and S. A. Hutchinson. Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 14(6):912–925, Dec. 1998.

[18] L. Luo and S. Akella. Optimal scheduling for biochemical analyses on digital microfluidic systems. *IEEE Transactions on Automation Science and Engineering*, 8(1):216–227, Mar. 2011.

[19] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[20] S. Park, C. Pan, T.-H. Wu, C. Kloss, S. Kalim, C. E. Callahan, M. Teitell, and E. P. Y. Chiou. Floating electrode optoelectronic tweezers: Light-driven dielectrophoretic droplet manipulation in electrically insulating oil medium. *Applied Physics Letters*, 92:151101–1–151101–3, 2008.

[21] S.-Y. Park, S. Kalim, C. Callahan, M. A. Teitell, and E. P. Y. Chiou. A light-induced dielectrophoretic droplet manipulation platform. *Lab Chip*, 9:3228–3235, 2009.

[22] S.-Y. Park, M. A. Teitell, and E. P. Y. Chiou. Single-sided continuous optoelectrowetting (SCOEW) for droplet manipulation with light patterns. *Lab Chip*, 10:1655–1661, 2010.

[23] S. N. Pei, J. K. Valley, S. L. Neale, H. Hsu, A. Jamshidi, and M. C. Wu. Rapid droplet mixing using light-actuated digital microfluidics. In *Conference on Lasers and Electro-Optics*, Optical Society of America Technical Digest (CD), San Jose, CA, USA, May 2010.

[24] S. N. Pei, J. K. Valley, S. L. Neale, A. Jamshidi, H. Hsu, and M. C. Wu. Light-actuated digital microfluidics for large-scale, parallel manipulation of arbitrarily sized droplets. In *23rd IEEE International Conference on Micro Electro Mechanical Systems*, pages 252–255, Wanchai, Hong Kong, Jan 2010.

[25] S. N. Pei and M. C. Wu. Personal communication, 2010.

[26] J. Peng and S. Akella. Coordinating multiple robots with kinodynamic constraints along specified paths. *International Journal of Robotics Research*, 24(4):295–310, Apr. 2005.

[27] M. G. Pollack and R. B. Fair. Electrowetting-based actuation of liquid droplets for microfluidic applications. *Applied Physics Letters*, 77:1725–1726, 2000.

[28] T. Simeon, S. Leroy, and J.-P. Laumond. Path coordination for multiple mobile robots: A resolution-complete algorithm. *IEEE Transactions on Robotics and Automation*, 18(1):42–49, Feb. 2002.

[29] L. A. Wolsey. *Integer Programming*. John Wiley and Sons, New York, 1998.

[30] P. Wurman, R. D'Andrea, and M. Mountz. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, 29(1):9–19, 2008.