

Towards Optimal Strategies for Moving Droplets in Digital Microfluidic Systems

Karl F. Böhringer

*Department of Electrical Engineering
University of Washington, Seattle
Seattle, WA 98195, USA
karl@ee.washington.edu*

Abstract - In digital microfluidic systems, analyte droplets (volume typically less than 1 μ l) are transported across a planar electrode array by dielectrophoretic or electrowetting effects. This paper outlines a high-level approach to optimally control digital microfluidic systems, i.e., to develop efficient algorithms that generate a sequence of control signals for moving one or many droplets from start to goal positions in the shortest number of steps, subject to constraints such as minimum required separation between droplets, obstacles on the array surface, and limitations in the control circuitry. However, optimality may be prohibitive for large-scale configurations because of the high asymptotic complexity. Alternative solutions include (1) an investigation of still useful but more limited system configurations; and (2) approximation algorithms that trade off optimality of the control sequences with higher efficiency of the algorithms that generate these control sequences.

Keywords - *digital microfluidics; droplet manipulation; control strategy; lab on a chip.*

I. INTRODUCTION

Advances in microfabrication and microelectromechanical systems (MEMS) over the past decades have led to a rapidly expanding collection of techniques to build systems for the handling and analyzing of very small quantities of liquids (see, e.g., [1]). These microfluidic systems typically consist of sub-millimeter scale components such as channels, valves, pumps, and reservoirs, as well as application-specific sensors and actuators. Microfluidic devices hold great promise, for example for novel fast, low-cost, portable, and disposable diagnostic tools. Applications include the massively parallel testing of new drugs, the on-site, real-time detection of toxins and pathogens, and PCR (polymerase chain reaction) for DNA sequence analysis. They usually operate with continuous flows of liquids, in analogy to traditional macro-scale laboratory setups, and integrate all functionality into complete “bio-systems-on-a-chip (bioSOCs)”.

A. Digital Microfluidic Systems

More recently, there has been increased interest in microfluidic devices that handle *discrete* droplets, with volumes usually in the sub-microliter range. In these “digital microfluidic systems” (DMFS), droplets are generated, transported, merged, analyzed, and disposed on planar arrays of

addressable cells. This architecture for microfluidic systems is attractive because of (1) greater flexibility – analyte handling may be reconfigured simply by re-programming rather than by changing the physical layout of the microfluidic components; (2) high droplet speeds – reportedly up to 25cm/s [2]; (3) no dilution and cross-contamination due to diffusion and shear-flow; and (4) the possibility for massively parallel microfluidic circuits.

B. Droplet Transport

Small droplets can be moved across a planar surface effectively with a variety of techniques, for example with electric fields (e.g., [2-6]), the thermocapillary effect (e.g., [7]), electrochemical surface modulation (e.g., [8]), or conformational changes in molecular surface layers (e.g., [9]). For the work in this paper, droplet transport with electric fields is most suitable; hence we briefly discuss the two main techniques in this realm.

1) Dielectrophoresis

In dielectrophoresis (DEP), neutrally charged objects are first polarized by an electric field, and then experience a net force due to the field. This force can only be non-zero if a field gradient exists, i.e., the positively and negatively polarized regions of the object occupy areas of different field strengths. If the object has stronger polarization than the surrounding medium then it is pulled towards the areas of higher field strength (this is called positive DEP), but if the surrounding medium has higher polarization, then the object is pushed towards areas of lower field strength (negative DEP). DEP can be considered the electrostatic analogy of induced magnetism. Common examples for DEP are charged clothes that attract (neutral) lint particles. More information on dielectrophoresis can be found, e.g., at [10].

2) Electrowetting

Electrowetting on dielectric (EWOD) exploits the decrease of contact angle that an aqueous droplet on a dielectric surface experiences when exposed to an electric field. If the field is localized at only one side of the droplet, then the difference in contact angle causes a pressure differential in the droplet, which drives it towards the region of higher field strength. Electrowetting and its applications in microfluidics have been investigated by several groups, including [2-4, 6].

II. RELATED WORK

Finding the optimal plan to generate, store, move, merge, split, and dispose multiple droplets on a digital microfluidic array is a complex problem, which combines general path planning and scheduling tasks with the more application-specific tasks of droplet generation, merging, and splitting. Various researchers have studied parts of the overall problem and have shown important results on algorithmic solutions and their computational complexity.

Each droplet can be interpreted as a point robot moving in a discrete two-dimensional configuration space. Under this assumption, path planning of the droplets becomes a robot motion planning problem with multiple moving robots. Erdmann and Lozano-Pérez showed in 1987 that this problem is NP-hard, but presented an algorithm that may find a good solution in polynomial time [12]. This approach assigns priorities to each robot (droplet) and generates paths successively, starting with the highest priority robot. Lower priority robots consider higher priority robots as time-varying obstacles that must be avoided. The algorithm is not complete, and generated solutions depend on the priority ranking of the robots and may not be optimal.

A rather different approach to this problem can be taken when the paths of the droplets are considered given a priori. Under this assumption, we obtain a scheduling problem, where the array cells en route are the limited resource that must be shared among different droplets. Recently, Akella et al. attacked this problem, again from the point of view of multiple coordinated robots. The problem is formulated as an integer programming problem, which can be solved with standard optimization tools [13, 14].

A similar technique was used by Ding, Zhang, et al. [15-17] who attack the problem from the VLSI design perspective. Again, the problem leads to an integer programming formulation, which is essentially equivalent to Akella's approach. Both groups show NP-hardness of the scheduling problem even for fixed robot (droplet) routes.

VLSI circuit routing techniques could also be employed, which address the path planning problem but do not apply directly to the inherently two-dimensional layout of the digital microfluidic platform.

In [18], this author described the problem as a graph search, and suggested search techniques such as A*. Even though this brute-force approach, unlike the other work mentioned above, guarantees optimality and completeness, it is not practical for larger scale problems because of its computational complexity, which is exponential in the number of moving droplets.

While it is not within the scope of this paper to develop a comprehensive algorithmic solution for the general problem of droplet manipulation on massively parallel microfluidic systems, we will attempt to present a formal problem definition and algorithms for partial solutions, and point in the direction of more general solutions for future work.

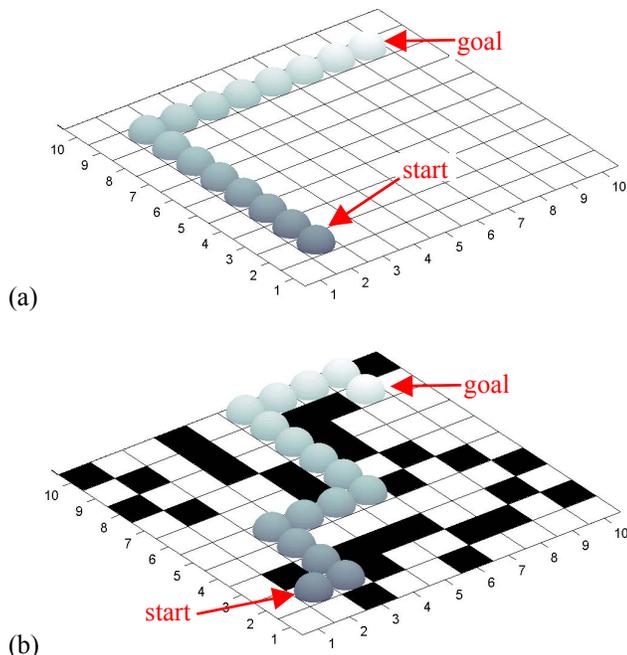


Figure 1. (a) Droplet moving on a 10×10 array from cell (2,2) to cell (9,9). The trace of the droplet is shown, with darker color indicating earlier steps. (b) Droplet moving from cell (2,2) to (9,9) while avoiding obstacles (“forbidden” cells shown in black). Here, an optimal strategy requires 16 steps, two more than in (a).

C. Examples

Figure 1 presents examples of control strategies for a simple digital microfluidic system. On a 10×10 array, a single droplet must be moved from cell (2,2) to cell (9,9). Figure 1a shows an optimal strategy consisting of 14 steps. In the system in Figure 1b, “forbidden” cells marked as black squares must be circumnavigated, resulting in a slightly longer solution sequence. In this paper, we describe how these solutions can be generated automatically, and generalize the approach to more complex scenarios with multiple moving droplets and additional constraints stemming from the specific physical implementation of the DMFS.

D. Paper Overview

The goal of this paper is to generate optimal sequences of control signals to move droplets from start to goal positions in the shortest number of steps. With growing array size and number of droplets, this becomes increasingly challenging: closely related optimizations are the traveling salesman problem, VLSI circuit routing, factory floor plan layout, resource scheduling, and motion planning with multiple moving robots, which are known to be computationally expensive (i.e., NP-hard [11]). Section II summarizes related work. Section III gives a more formal problem definition. Algorithms to control DMFS are discussed in Section IV, and examples applicable to different DMFS hardware configurations are presented in Section V. Section VI summarizes the paper, and gives conclusions and an outlook on future work.

III. DMFS HARDWARE SPECIFICATION AND FORMAL PROBLEM DEFINITION

Let us first specify the important physical properties and design parameters of a digital microfluidic system. Then we can move on to a more abstract DMFS model that is independent of specific implementation details.

A. DMFS Design Specifications

- **Layout:** Typically, a DMFS consists of a rectangular array A with $m \times n$ cells (but, e.g., an arrangement of hexagonal cells is also possible).
- **Control circuitry:** Various addressing schemes are possible to activate individual cells in a DMFS. We can distinguish, e.g., individually addressable electrodes for each cell, or simpler row/column addressing. For the latter, entire rows and columns are activated, and the droplet is attracted to a neighboring cell $A(x,y)$ only if it lies at the intersection of active column x and row y .
- **Parallelism:** Does the DMFS controller allow simultaneous activation of more than one cell, and is the total number of active cells limited by a number significantly smaller than $m \times n$?
- **Location of cells with special functions:** Droplet generators, reservoirs, cells for merging and splitting of droplets, sensors, waste, etc. may require dedicated cells with special embedded hardware.

These specifications provide a physical framework within which a DMFS can operate. Based on this framework, we can establish a formal description of the problem of controlling droplets in a DMFS. Once a sufficiently general DMFS model exists, we can investigate algorithmic solutions at an abstract level, without worrying about the specific details of varying hardware implementations.

B. Problem Definition

A digital microfluidic system is given by an array A with d droplets, their start locations, and their goal locations. Our aim is to automatically generate a strategy to move the droplets from start to goal (as shown, e.g., in Figure 1). More specifically, droplets can be of several types T_i (e.g., T_1 = “DI water”, T_2 = “buffer solution”, etc.), with $i \in \{1 \dots t\}$, and t the total number of different droplet types.

Each cell $A(x,y)$ in the array can be either occupied by a droplet (denoted as “ T_i ”), empty (“ \emptyset ”), or blocked by an obstacle (“ X ”). Thus, at any given time the system can be described by $A(x,y) = c_{xy}$ for $(x,y) \in \{1 \dots m\} \times \{1 \dots n\}$ and $c_{xy} \in C = \{T_1, \dots, T_t, \emptyset, X\}$. In particular, given a start placement $A_s \in C^{m \times n}$ and a goal placement $A_g \in C^{m \times n}$, we need to find a sequence of valid transitions that results in the desired droplet motion from A_s to A_g .

Various kinds of transitions exist, including droplet generation, moving, disposing, merging, and splitting.

- **Droplet generation:** For $(x,y) \in \{1 \dots m\} \times \{1 \dots n\}$ and some $i \in \{1 \dots t\}$, a droplet is generated at coordinate (x,y) if

$A(x,y) = \emptyset$ at time t and $A(x,y) = T_i$ at time $t+1$.

- **Moving:** Let (x,y) and $(x',y') \in \{1 \dots m\} \times \{1 \dots n\}$ and $|x-x'| + |y-y'| = 1$ (i.e., $A(x,y)$ and $A(x',y')$ are directly adjacent). At time t , $A(x,y) = T_i$ and $A(x',y') = \emptyset$ and at time $t+1$, $A(x,y) = \emptyset$ and $A(x',y') = T_i$.
- **Merging:** Let again (x,y) and $(x',y') \in \{1 \dots m\} \times \{1 \dots n\}$ and $|x-x'| + |y-y'| = 1$. At time t , $A(x,y) = T_i$ and $A(x',y') = T_j$, and at time $t+1$, $A(x,y) = T_k$ and $A(x',y') = \emptyset$, where T_k is the droplet type that results in merging types T_i and T_j .
- **Splitting:** Definition similar to merging.
- **Disposing:** Definition similar to droplet generation.

In addition, to avoid accidental merging of droplets, at least one empty cell is required between two occupied cells at all times. Transitions are further restricted by the addressing circuitry and cells with specialized functions.

IV. DMFS CONTROL STRATEGIES

This section focuses on a limited but important subproblem in the control of DMFS: generating efficient paths for multiple droplets that move from a given start configuration A_s to a desired goal configuration A_g . We will first give a simple, complete algorithm based on A* search, but find that its computational complexity is very high (exponential in number of droplets). We then present a more efficient algorithm that trades off completeness for faster execution times.

A. Basic Algorithm Outline

This algorithm maintains a graph data structure to represent the array (inclusive special cells and obstacles) and to keep track of droplet locations. At any given time t_i , the state of the DMFS is described by $A_i \in C^{m \times n}$, representing a node in the graph. Transitions between states define edges in this graph, and finding an optimal control strategy to transform start state A_s into goal state A_g becomes a standard graph search problem, which can be solved, for example, using an A* algorithm known from artificial intelligence programming [19]: A* graph search employs a metric that estimates the expected cost of a

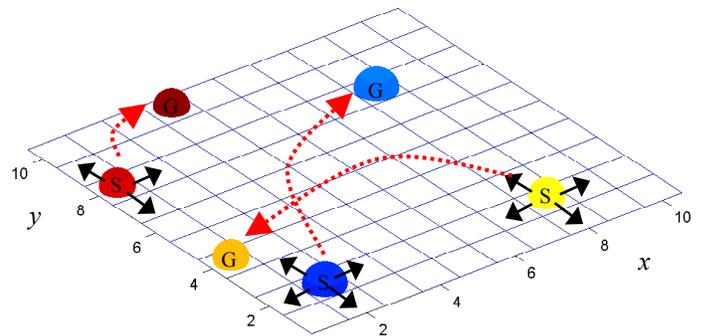


Figure 2. Three droplets with respective start and goal positions (indicated by S and G). The number of choices grows exponentially with the number of droplets. At any time there are up to 4^3 choices for the next step, and at least 12 steps are required to move all droplets simultaneously from start to goal. Hence, straightforward programming could produce software attempting to explore $(4^3)^{12} > 10^{28}$ choices.

partial solution path in the directed graph. This estimate provides a heuristic that gives preference to the more promising paths. It can be shown that if certain “admissible” metrics are used, then A* is guaranteed to find an optimal solution if one exists, and indicates failure otherwise.

The downside of this approach is its high asymptotic complexity. Suppose the number of droplets is d . In the simplest case, all are of the same type T_0 . Then the number of different placements of droplets on the array is $\binom{mn}{d}$, which for modest numbers $m=n=10$ and $d=10$ yields more than 1.7×10^{13} possibilities. If all droplets are of distinct type $T_1 \dots T_d$, this number increases by $d!$ (to $\approx 3 \times 10^{19}$). One might hope that in practice, most of these choices need not be explored. However, at each step, d droplets offer up to $4d$ choices to be moved, assuming 4 neighbor cells per droplet. Thus, finding a strategy with s steps could mean checking up to $(4d)^s$ choices or risk missing the solution, resulting again in astronomical numbers even for $s < 10$. This is illustrated in Figure 2.

We conclude that the search graph explored with the A* algorithm has $O((mn)!)$ nodes and a branching factor of $O(4s)$, leading to prohibitive complexity for any non-trivial array size with more than a few droplets.

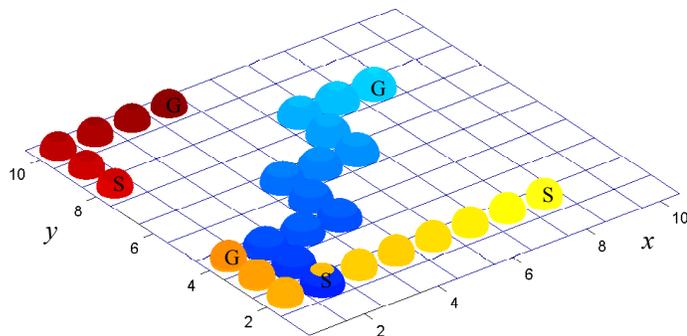


Figure 3. Optimal solution to the setup in Figure 2 by the prioritizing algorithm. The blue droplet was assigned highest priority and an optimal motion (12 steps) was generated. The yellow droplet requires 9 steps and moves over cell (2,2) previously occupied by the blue droplet. The red droplet does not interfere with the other droplets in this case.

B. Prioritized Droplet Control

The discussion above has shown that droplet motion planning for DMFS has two main aspects: generating efficient droplet motion plans, and finding efficient algorithms to generate these plans. Because of the inherent complexity of the problem, compromises need to be made to obtain practical solutions, and completeness or optimality in motion plans has to be traded off with efficiency in plan generation.

This section applies ideas from Erdmann and Lozano-Pérez [12] to DMFS control. The algorithm proceeds as follows:

- (1) Assign priorities to each droplet in the DMFS. This can be done at random, or based on application-specific guidelines (e.g., water may have lower priority than droplets containing expensive or volatile compounds).

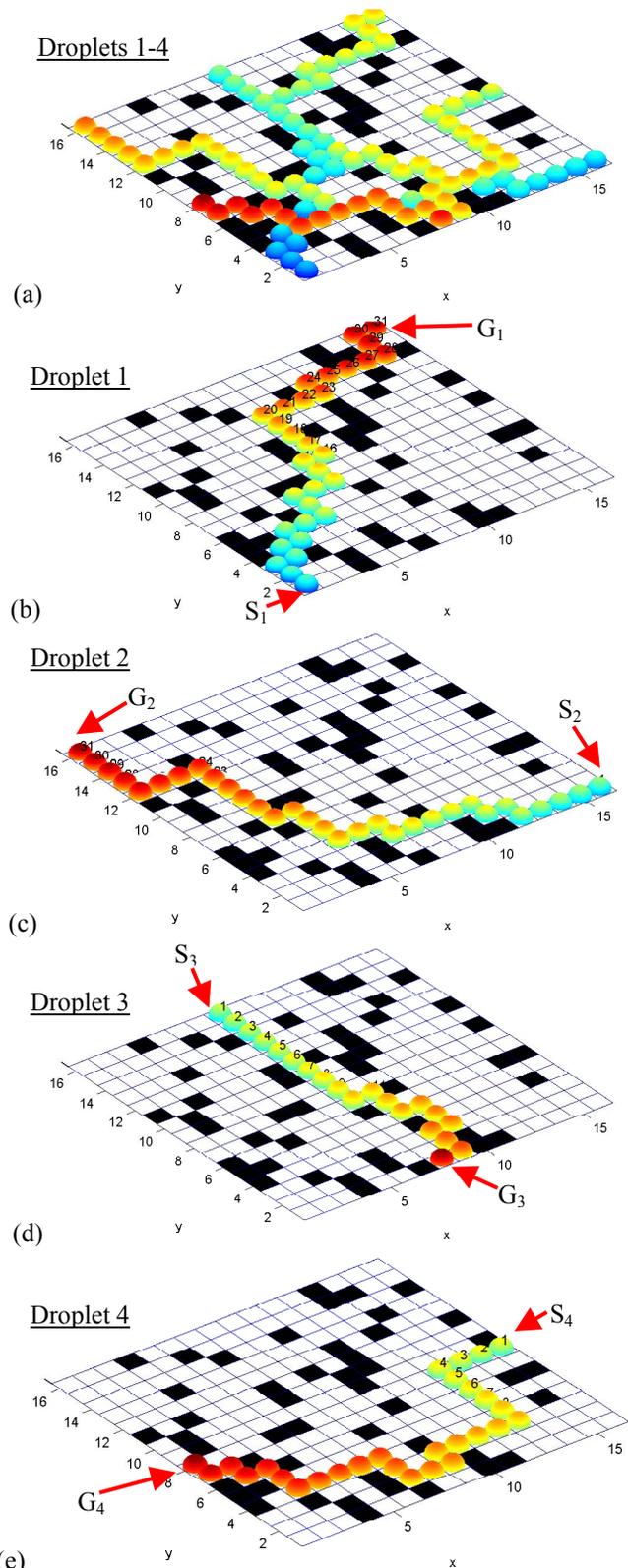


Figure 4. (a) Four droplets moving simultaneously from start $S_1=(1,1)$, $S_2=(16,1)$, $S_3=(8,16)$, $S_4=(16,8)$ to goal $G_1=(16,16)$, $G_2=(1,16)$, $G_3=(8,1)$, $G_4=(1,8)$. (b-e) Individual paths (with time stamps) for droplets 1 through 4 in decreasing order of priority. Solutions generated with sequential prioritized A* algorithm.

- (2) For each droplet, starting with the highest priority, generate an optimal motion plan. Droplets with higher priorities are considered time-dependent obstacles. Droplets with lower priorities are ignored.

This algorithm eliminates the exponential complexity in d , where d is the number of droplets in the DMFS. Instead, as the complexity of the A* algorithm for path planning of a single droplet is $O(nm \log(nm))$, the complexity to determine d droplet paths with this sequential prioritized approach is simply $O(dnm \log(nm))$. However, as stated above, this algorithm is neither complete, nor are the generated paths necessarily optimal. Figure 3 gives an example of this algorithm for the start and goal configurations of Figure 2.

Figure 4 shows a more extensive example of this algorithm. On a 16×16 array with randomly distributed obstacles, four droplets are initially placed at (1,1), (16,1), (8,16), and (16,8). Their respective goals are at (16,16), (1,16), (8,1), and (1,8). Figure 4a shows the simultaneous trace of all droplets. Figures 4b-e depict the individual traces for each of the four droplets. We can observe that the two droplets with the highest priorities (Figures 4b and 4c) achieve an optimal path with 31 steps each. Droplet 3 (Figure 4d) has to evade droplets 1 and 2 and therefore turns left in steps 10 and 13, instead of choosing the shorter path towards the right. Similarly, droplet 4 (Figure 4e) would interfere with higher priority droplets, were it to travel on a more direct path towards its goal.

The solution in Figure 4 was generated in a few seconds by a simple MATLAB implementation of this algorithm.

In the following section we show more examples performed with variations of this algorithm. They include multiple droplets, obstacles, and constraints on the control circuitry. Even though rather simple, these examples should summarize the basic principles of DMFS control strategies, and motivate ideas for improved algorithms, which will be summarized in Section VI.

V. OTHER SAMPLE DROPLET MANIPULATION STRATEGIES

In this section we show two additional examples of optimal control strategies. In Figure 5 two droplets of different types require 8 steps to switch their positions while circumnavigating an obstacle.

This strategy assumes that the electrode in each cell can be activated independently from all other cells. The two droplets are always separated by at least one empty cell, such that accidental merging is avoided. Note that the

darker droplet moves more than necessary (gratuitous steps 4 and 5), but this does not affect the overall number of 8 steps in the control strategy. Future software improvements will eliminate this programming artifact.

A. Limited Row-Column Addressing

The previous examples (Figure 1-5) assumed that each cell in the array is individually addressable. However, [20] introduced a simpler addressing scheme for DMFS based on a top layer of row electrodes and a bottom layer of column electrodes. Droplets move to a neighboring cell whose row and column address has been activated. This scheme creates additional constraints on the droplet motion. Two droplets trade places as in Figure 5 above, but here droplets move only to cells whose row and column address has been activated (indicated by triangular arrows). An optimal strategy now requires 9 steps, one more step than in Figure 5. Figure 6 shows the same task as Figure 5 but performed only with row-column addressing, resulting in a longer sequence.

Note that here we assumed that we can activate an arbitrary number of rows and columns simultaneously (for d droplets, up to d active rows and columns are useful). Further hardware constraints could limit this number, possibly to a single row and column. If so, longer control sequences could result, but the branching factor at each step would drop from $O(4^d)$ to $O(d)$.

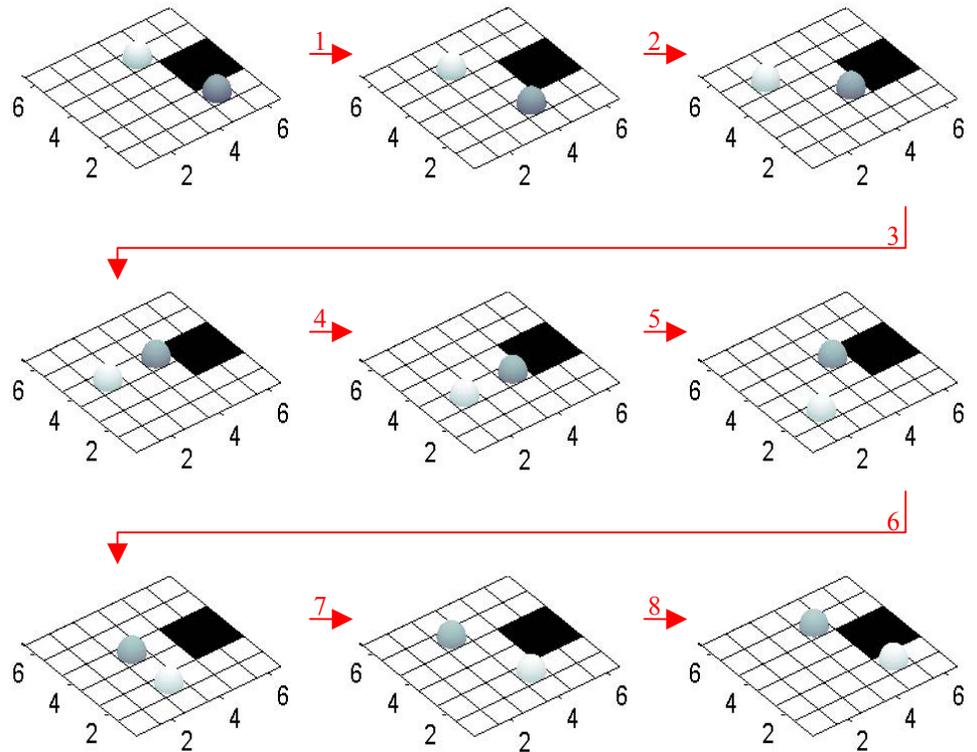


Figure 5. Two droplets moving simultaneously on a 6×6 array while avoiding an obstacle (black cells). The two droplets start at cells (5,2) and (4,5), and require 8 steps to trade places. Solution generated with complete multi-droplet A* algorithm.

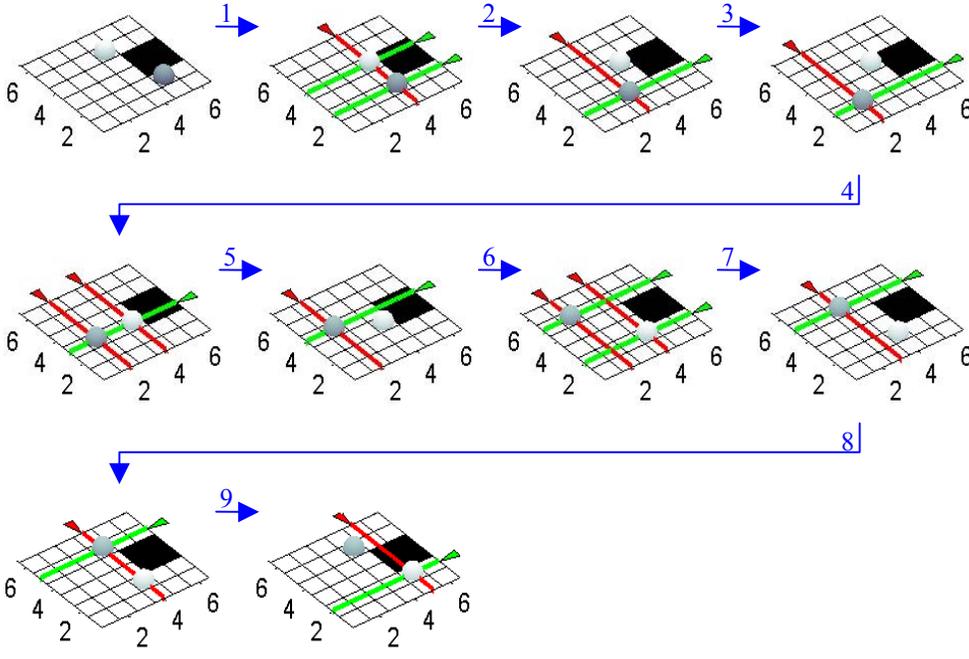


Figure 6. Two droplets trading places as in Figure 5 above, but here droplets move only to cells whose row and column address has been activated (indicated by triangular arrows). An optimal strategy now requires 9 steps, one more step than in Figure 5. Solution generated with complete multi-droplet A* algorithm.

VI. CONCLUSIONS AND FUTURE WORK

Digital microfluidic systems (DMFS) based on droplet manipulation are promising because of their flexibility and reconfigurability: they shift complexity from microfluidics hardware to control software.

Droplet manipulation based on electrowetting on arrays with up to hundred cells has been demonstrated by several groups (e.g., [2, 20, 21]), and electrophoresis-based systems with integrated CMOS addressing include tens of thousands of cells [22]. Unfortunately, our investigations suggest that for such large-scale DMFS, the generation of optimal strategies for droplet manipulation may become computationally intractable. This raises the question whether DMFS can really live up to the promise of full programmability and reconfigurability. Instead of making full use of these advantages, the computational complexity may limit DMFS to much more constraint applications.

In this paper, we have shown one possible answer to this challenge: Instead of insisting on optimal strategies, an algorithm that trades off completeness and optimality for polynomial run-time was presented. Motion planning for multiple simultaneously moving droplets is replaced with a

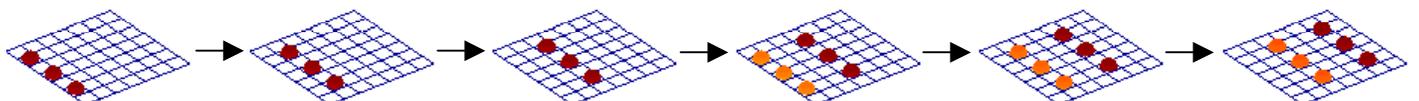


Figure 7. “Peristaltic” droplet motion on DMFS. In this simple example, groups of 3 droplets move in parallel along straight paths without any overlap. Every three steps, the pattern repeats.

sequence of motion planning tasks for individual droplets, whereby higher priority droplets are considered moving obstacles for lower-priority droplets. In our initial experiments, the resulting motion strategies are computed efficiently and are generally near-optimal. Future work will include a more rigorous analysis of this algorithm.

Another possible answer could be to limit the droplet manipulation strategies to a few standard, “pre-packaged” strategies. For example, on a 100×100 array, about 50 droplets could move in parallel across the array, followed by another wave of 50 droplets, etc., resembling a “peristaltic” motion (Figure 7). However, in this case, the fundamental advantage of flexibility and reprogrammability in DMFS versus conventional microfluidic architectures (channel, valve, and pump based) is lost. In addition, the question still remains how to initially generate the “pre-packaged” strategies if they involve

more complex non-linear motions of many droplets.

Thus, our goal is to find efficient algorithms for more general control strategies with DMFS. Towards this end, this paper presented a formal problem definition and several algorithms and examples. Future work can expand in the following directions:

1. Polynomial approximation algorithms exist for NP-hard problems (e.g., traveling salesman), which guarantee a tight limit on non-optimality. If, e.g., a control strategy for a complex DMFS can be generated in polynomial time that is guaranteed to be at most twice as long as an optimal solution then this might be sufficient for most practical purposes.
2. As suggested at the end of Section V, the branching factor for the graph search is greatly reduced if it is assumed that only one droplet can move at each step. Thus, even if the hardware allows simultaneous motion of droplets (e.g., with individually addressable cells), it may be more effective to first generate a motion plan consisting of single droplet moves, and then perform a post-processing step that “parallelizes” the plan as much as possible. At the moment, it is unknown whether the plans thus generated

would still be optimal, and whether the asymptotic complexity of the algorithm would improve.

3. In practical systems, even if there are very large numbers of individual droplets, there may be only a rather small number of different types of droplets. The algorithms presented here do not take full advantage of this fact, even though it is believed that this could lead to a substantial decrease in the search space. The resulting algorithms might be significantly different from those presented in Section V, as there is no a priori correspondence between initial and final placement for each droplet within a set of droplets of a given type.
4. Finally, extending the software to handle other common operations in DMFS, such as splitting and merging of droplets, is an important direction of future research.

Our software is available at www.ee.washington.edu/research/mems/digitalfluidics.

ACKNOWLEDGMENT

The author thanks Srinivas Akella, Bruce R. Donald, Michael Erdmann, Rajinder Khosla, Vamsee K. Pamula, Vijay Srinivasan, and Xiaorong Xiong for helpful insights and comments, and Rohit Malhotra for software programming.

REFERENCES

- [1] Kovacs, G.T.A., *Micromachined Transducers Sourcebook*. 1998: McGraw-Hill.
- [2] Moon, H., S.K. Cho, R.L. Garrell, and C.-J. Kim, *Low voltage electrowetting-on-dielectric*. Journal of Applied Physics, 2002. **92**(7): p. 4080-4087.
- [3] Beni, G. and M.A. Tenan, *Dynamics of electrowetting displays*. Applied Physics, 1981. **52**: p. 6011-6015.
- [4] Pollack, M.G., R.B. Fair, and A.D. Shenderov, *Electrowetting-based actuation of liquid droplets for microfluidic applications*. Applied Physics Letters, 2000. **77**: p. 1725-1726.
- [5] Jones, T.B., M. Gunji, M. Washizu, and M.J. Feldman, *Dielectrophoretic liquid actuation and nanodroplet formation*. Journal of Applied Physics, 2001. **89**(2): p. 1441-1448.
- [6] www.nanolytics.com.
- [7] Kataoka, D.E. and S.M. Troian, *Patterning Liquid Flow at the Microscopic Scale*. Nature, 1999. **402**: p. 794-797.
- [8] Gallardo, B.S., V.K. Gupta, F.D. Eagerton, L.I. Jong, V.S. Craig, R.R. Shah, and N.L. Abbott, *Electrochemical principles for active control of liquids on submillimeter scales*. Science, 1999. **283**: p. 57-89.
- [9] Lahann, J., S. Mitragotri, T.-N. Tran, H. Kaido, J. Sundaram, I.S. Choi, S. Hoffer, G.A. So-morjai, and R. Langer, *A Reversibly Switching Surface*. Science, 2003. **299**: p. 371-374.
- [10] www.dielectrophoresis.org.
- [11] Garey, M.R. and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. 1979: W. H. Freeman & Co.
- [12] Erdmann, M. and T. Lozano-Pérez, *On Multiple Moving Objects*. Algorithmica, 1987. **2**: p. 477-521.
- [13] Akella, S. and S. Hutchinson. *Coordinating the Motions of Multiple Robots with Specified Trajectories*. in *IEEE International Conference on Robotics and Automation*. 2002. Washington D.C.
- [14] Peng, J. and S. Akella. *Coordinating Multiple Robots with Kinodynamic Constraints along Specified Paths*. in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*. 2002.
- [15] Ding, J., 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, 2001. **20**(12): p. 1463-1468.
- [16] Zhang, T., K. Chakrabarty, and R.B. Fair, *Integrated hierarchical design of microelectrofluidic systems using SystemC*. Microelectronics Journal, 2002. **33**: p. 459-470.
- [17] Zhang, T., K. Chakrabarty, and R.B. Fair, *Design of Reconfigurable Composite Microsystems Based on Hardware/Software Codesign Principles*. IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, 2002. **21**(8): p. 987-995.
- [18] Böhringer, K.F. *Optimal Strategies for Moving Droplets in Digital Microfluidic Systems*. in *Seventh International Conference on Miniaturized Chemical and Biochemical Analysis Systems (MicroTAS'03)*. 2003. Squaw Valley, CA.
- [19] Nilsson, N.J., *Principles of Artificial Intelligence*. 1982, Berlin Heidelberg New York: Springer Verlag.
- [20] Fan, S.-K., P.-P.d. Guzman, and C.-J. Kim. *EWOD Driving of Droplet on NxM Grid Using Single Layer Electrode Patterns*. in *Solid-State Sensor, Actuator, and Microsystems Workshop*. 2002. Hilton Head Island, SC.
- [21] Srinivasan, V., V.K. Pamula, M.G. Pollack, and R.B. Fair. *Clinical Diagnostics on Human Whole Blood, Plasma, Serum, Urin, Saliva, Sweat, and Tears on a Digital Microfluidic Platform*. in *Micro Total Analysis Systems (MicroTAS)*. 2003. Squaw Valley, CA.
- [22] Fuchs, A., et al. *A Microelectronic Chip Opens New Fields in Rare Cell Population Analysis and Individual Cell Biology*. in *Micro Total Analysis Systems (MicroTAS)*. 2003. Squaw Valley, CA.