

# Optimal Testing of Digital Microfluidic Biochips

Bogdan Paşaniuc

Department of Epidemiology, Harvard School of Public Health, 655 Huntington Avenue, Boston, Massachusetts 02115,  
bpaşaniu@hsph.harvard.edu, <http://www.hsph.harvard.edu/research/bogdan-paşaniuc/>

Robert Garfinkel

Operations and Information Management Department, University of Connecticut, School of Business, Box U411M, Storrs, CT  
06269, robert.garfinkel@business.uconn.edu, <http://www.business.uconn.edu/cms/p461/u34/r0/mc>

Ion Măndoiu

Computer Science and Engineering Department, University of Connecticut, 371 Fairfield Way, Unit 2155, Storrs, CT  
06269-2155, ion@engr.uconn.edu, <http://www.engr.uconn.edu/~ion/>

Alex Zelikovsky

Computer Science Department, Georgia State University, University Plaza, Atlanta, Georgia 30303, alexz@cs.gsu.edu,  
<http://www.cs.gsu.edu/~cscazz/>

Digital microfluidic biochips (DMFB) are rectangular arrays of electrodes, or cells, that enable precise manipulation of nanoliter-sized droplets of biological fluids and chemical reagents. Due to the safety-critical nature of their applications, biochips must be tested frequently, both *off-line* (e.g., post-manufacturing) and *concurrent* with assay execution. Under both scenarios, testing is accomplished by routing one or more test droplets across the chip and recording their arrival at the destination. In this paper we formalize the DMFB testing problem under the common objective of completion time minimization, including previously ignored constraints of droplet non-interference. Our contributions include a proof that the general version of the problem is NP-hard, tight lower-bounds for both off-line and concurrent testing, optimal and approximation algorithms for off-line testing of commonly used rectangular shaped biochips, as well as a concurrent testing heuristic producing solutions within 23-34% of the lower-bound in experiments conducted on datasets simulating varying percentages of biochip cells occupied by concurrently running assays.

*Key words:* Digital microfluidic biochips; lab-on-a-chip; defect testing; exact and approximation algorithms; heuristics

---

## 1. Introduction

Microfluidic biochips, also referred to as lab-on-a-chip systems, enable the precise manipulation of nanoliter volumes of biological fluids and chemical reagents. Due to their high speed and sensitivity and much lower manufacturing and operating costs, microfluidic biochips are increasingly replacing traditional “macrofluidic” laboratory equipment for performing a broad range of biochemical analyses in genomics, proteomics, clinical diagnostics, environmental monitoring, and bio-defense. While early generations of microfluidic biochips used continuous fluid flow through permanently etched capillaries, more recent biochips, commonly referred to as *digital microfluidic biochips* (DMFBs), manipulate the fluids in discrete droplets using a regular array of electrodes, see Pollack et al. (2002). In this biochip architecture, droplets are dispensed from external reservoirs and then moved, split, merged, and mixed based on the electro-wetting principle by controlling the voltage applied to the electrodes. This provides increased flexibility, since many different biochemical analyses can be performed on the same biochip by simply changing the pattern in which electrode voltages are applied. While this requires thorough cleaning and sterilization between uses, this may still prove to be more cost effective than using disposable devices.

Testing of DMFBs has received much attention in the past few years due to the safety-critical nature of many of their applications, see, e.g., Chakrabarty and Su (2007) and Kerkhoff (2007) for

recent reviews. Most common DMFB defects can be detected by routing one or more test droplets across the chip and using video CCD cameras to monitor droplet movement or recording their arrival at the destination. For a single droplet, test schedule optimization has been modeled as a Hamiltonian path problem by Su et al. (2006a), enabling the exact solution of small instances via integer programming techniques. However, using a single droplet for testing large biochips is not practical. For example, single-droplet testing of a rectangular biochip with  $480 \times 640$  electrodes such as the DEParray commercialized by Silicon Biosystems would require over three days at a typical actuation frequency of 1Hz. Testing time can be dramatically reduced by using multiple test droplets, but in this case it is necessary to take into account possible interference between droplets that are too close to each other. Although Su et al. (2006a) extends the single-droplet integer programming formulation to the case of multiple test droplets with pre-specified start and end positions, this extension does not capture all relevant non-interference constraints and has impractical solution time even for moderate chip sizes.

Under different optimization objectives, the problem of coordinating multiple droplets has been previously considered in the context of droplet routing during assay execution on a DMFS. Similar coordination problems have also been considered in the area of robot motion planning. For the case when robot paths are specified, Akella and Hutchinson (2002) presents a mixed integer linear programming (MILP) formulation for the coordination of 10-20 robots by changing robot start times. Also for the case when robot paths are specified, Peng and Akella (2005) show that collision avoidance and dynamics constraints can be formulated as a mixed integer nonlinear program and show the convexity of the constraints for individual robot path segments. When droplet trajectories are not specified Bohringer (2004) notices that coordination of multiple droplets on a DMFS is closely related to the multiple robot motion coordination problem, and proposes an A\* search algorithm within a multi-robot cooperation approach to find a multiple droplet routing plan. Griffith et al. (2006) and Griffith and Akella (2005) use Dijkstra’s algorithm for finding a suitable routing plan for multiple droplets.

In this paper we formalize the DMFB testing problem under the common objective of completion time minimization, including previously ignored constraints of droplet non-interference (Section 2). Unlike Su et al. (2006a), we do not assume that the number of droplets is known in advance, allowing implicit optimization over this important degree of freedom. Indeed, finding the optimal number of droplets is non-trivial due to the complex tradeoffs between gains achieved by sharing the work among multiple droplets and losses due to increased interference constraints generated when more droplets are used.

Our contributions include a proof that the general version of the problem is NP-hard (Section 3), via a reduction from the Hamiltonian circuit problem in grid graphs. In Section 4 we establish lower-bounds and introduce several algorithms for off-line testing of rectangular biochips, including optimal algorithms for square biochips and biochips with aspect ratio of 4 or more. The performance of these algorithms is summarized in Table 1. Lower-bounds and heuristics for concurrent testing are discussed in Section 5, along with results of experiments conducted on synthetic datasets showing an excellent performance, with an average gap of 23-34% between the heuristic upper-bound and our lower-bound. The paper concludes with a discussion of future research directions (Section 6) and an integer programming formulation of the problem (Appendix 6).

### 1.1. Related Literature in Operations Research

The area of operations research most closely related to DMFB testing is vehicle routing, with the vehicles being the analogs of droplets. The general problem is how best to start at one or more depots, and visit a pre-specified subset of nodes in a network with a given number of vehicles. In general it is expected that the vehicles return to their depots although this is not always required. One typical objective, although others abound, is to minimize the total distance traveled. The

number of vehicles is typically taken to be fixed, with one notable exception mentioned below. There are many possible additional considerations: the topology of the network; the presence of time windows for deliveries; capacities of the vehicles; items to be picked up as well as delivered, etc. For reviews of the many variations of the vehicle routing problem the reader is directed to Christofides (1985) and Toth and Vigo (2002).

An application in which the number of vehicles is not fixed, but in fact is often taken to be the objective function, is the school bus routing problem discussed by Angel et al. (1972). There, constraints would generally be on the maximum length of any student’s ride. A distinctive feature of DMFB testing is the presence of “droplet non-interference” constraints. Such constraints are not common in the vehicle routing literature. Perhaps the model most similar to ours is that of Savchenko and Frazzoli (2005). Similar to DMFB testing, the bus routing problem in Savchenko and Frazzoli (2005) has completion time as the objective and does not allow vehicles to get “too close” to each other, depending on how fast they are travelling. On the other hand there are major differences. In particular, for us the number of “vehicles” (droplets) is a variable, and movement is on a grid network as opposed to the plane in Savchenko and Frazzoli (2005). This distinction causes the definition of interference to differ dramatically. Prior works on vehicle routing on a grid network include Frizzel and Griffin (1995) and Rubrico et al. (2004), but these assume a given number of vehicles and do not model non-interference constraints.

## 2. Problem Formulation

Digital microfluidics biochips use identically shaped and sized electrodes forming a regular tiling of the chip. Although electrodes shaped as equilateral triangles or regular hexagons can also be used (advantages for using the latter are discussed by Su et al. (2005)), square electrodes are currently the prevalent choice owing to the less complex manufacturing involved. Thus, for the rest of the paper we focus on biochips with square shaped electrodes arranged in a regular grid. Without loss of generality, we will assume that the Cartesian coordinates of all electrode centers are integer. We will interchangeably refer to electrodes as *cells*, and, by abusing the language, we will identify them with the Cartesian coordinates of their centers.

Formally, we denote by  $R$  the set of cells available on the biochip. We distinguish sets  $I \subset R$  of input cells and  $O \subset R$  of output cells, typically located on the border of the chip. The cells of  $I$  are adjacent to reservoirs from which test droplets are dispensed, while those of  $O$  are adjacent to reservoirs for droplet disposal (output cells may also be equipped with devices such as photodiodes for detecting the arrival of test droplets). To avoid droplet merging with the content of input reservoirs, test droplets are not allowed to move back to an input cell after having visited other cells on the chip. Similarly, test droplets are immediately disposed of upon reaching an output cell.

We denote by  $N(i, j) = \{(i - 1, j), (i + 1, j), (i, j - 1), (i, j + 1)\} \cap R$  the set of cells of  $R$  sharing an edge with cell  $(i, j)$ , and refer to it as the *neighborhood* of  $(i, j)$  (see Figure 1(a)). Droplet movement is controlled by a synchronous system clock signal – in each clock cycle a droplet can either stay put or move to one of its neighbor cells in  $N(i, j)$ .

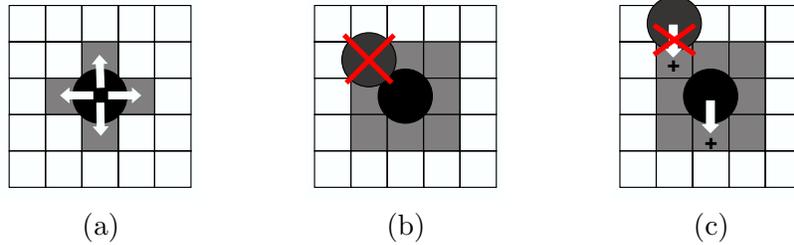
DEFINITION 1. A *droplet trajectory* starting at clock cycle  $t_b \geq 0$  and ending at clock cycle  $t_e > t_b$  is a sequence of cell coordinates  $\pi = (x_t, y_t)_{t_b \leq t \leq t_e}$  such that  $(x_{t_b}, y_{t_b}) \in I$ ,  $(x_{t_e}, y_{t_e}) \in O$ ,

Algorithm	# Droplets	Completion Time	Remarks
Vertical Stripes	$n/3$	$2n + 3m - 5$	$5/4$ approximation for $n = m$
Interleaved rows	$m$	$n + 4m - 3$	optimal for $n \geq 4m$
Interleaved Zig-Zags	$m/2$	$2n + 2m - 3$	optimal for $n = m$

Table 1 Performance of proposed off-line testing algorithms on  $m \times n$  rectangular biochips (under divisibility assumptions stated in Theorems 3-5).

$(x_t, y_t) \in R \setminus (I \cup O)$  for every  $t_b < t < t_e$ , and  $(x_{t+1}, y_{t+1}) \in N(x_t, y_t) \cup \{(x_t, y_t)\}$  for every  $t_b \leq t < t_e$ . The *length* of trajectory  $\pi$  is defined as  $t_e - t_b$ .

For conciseness, when the start and end cells are clear from the context, we will alternatively describe a droplet trajectory by specifying the starting time  $t_b$  and a string of length  $t_e - t_b$  over the alphabet  $\{R, L, U, D, P\}$  describing droplet moves during each clock cycle, with  $R$ ,  $L$ ,  $U$ , and  $D$  denoting a droplet move to the right, left, up, and down, respectively, and  $P$  denoting a clock cycle in which the droplet stays put.



**Figure 1** Neighborhood (a), merging region (b), and interference region (c) for droplets moving on a DMFB with square shaped electrodes.

As noted by Su et al. (2006a), when two or more droplets move simultaneously on the biochip, they must not be allowed to get too close to each other to avoid accidental merging. We let  $M(i, j)$  denote the set of cells that must be kept empty in order to prevent merging with a droplet occupying cell  $(i, j)$ , and refer to this set as the *merging region* of cell  $(i, j)$ . As in previous works, e.g., Su et al. (2006a), the merging region is defined as  $M(i, j) = \{(i', j') \neq (i, j) : |i' - i| \leq 1, |j' - j| \leq 1\} \cap R$  (see Figure 1(b)). In other words, we assume that droplet merging will not only take place between droplets located in horizontally or vertically adjacent cells, but also between droplets in diagonally adjacent cells, which is due to the fact that droplets are slightly larger than electrode size (see Kerkhoff (2007)).

**DEFINITION 2.** Two droplet trajectories  $\pi = (x_t, y_t)_{t_b \leq t \leq t_e}$  and  $\pi' = (x'_t, y'_t)_{t'_b \leq t \leq t'_e}$  are *merge-free* if  $(x'_t, y'_t) \notin M(x_t, y_t)$  for every  $\max\{t_b, t'_b\} \leq t \leq \min\{t_e, t'_e\}$ .

Additional compatibility requirements between droplet trajectories are imposed by the voltage modulation used to actuate droplet movement via electro-wetting. As reviewed by Pollack et al. (2002), moving a droplet from cell  $(i, j)$  to one of the neighbor cells  $(i', j') \in N(i, j)$  involves applying a positive voltage to the destination cell. If a positive voltage is simultaneously applied to a second nearby cell (with the intent of moving another droplet to that location), the droplet located at cell  $(i, j)$  may stay put, move to an unintended location, or even split – see Figure 1(c) for an example. Preventing such actuation-caused interference was ignored in previous works Su et al. (2006a,b). This requires a different set of compatibility constraints between droplet trajectories than preventing droplet merging. In particular, while preventing droplet merging constrains the positions occupied by droplets in any *single* clock cycle, preventing droplet interference constrains droplet positions in *consecutive* cycles. Formally, we let  $I(i, j)$  denote the *interference region* of cell  $(i, j)$ , i.e., the set of cells for which applying a positive voltage during time interval  $(t, t + 1)$  could interfere with the scheduled movement of a droplet located at cell  $(i, j)$  at time  $t$ . For unit square cells,  $I(i, j)$  is taken to be identical to the merging region, i.e.,  $I(i, j) = \{(i', j') \neq (i, j) : |i' - i| \leq 1, |j' - j| \leq 1\} \cap R$ .

**DEFINITION 3.** Two droplet trajectories  $\pi = (x_t, y_t)_{t_b \leq t \leq t_e}$  and  $\pi' = (x'_t, y'_t)_{t'_b \leq t \leq t'_e}$  are *interference-free* if  $(x'_{t+1}, y'_{t+1}) \notin I(x_t, y_t)$  and  $(x_{t+1}, y_{t+1}) \notin I(x'_t, y'_t)$  for every  $\max\{t_b, t'_b\} \leq t \leq \min\{t_e, t'_e\}$ .

Note that, unlike the definition of merge-free trajectories, where symmetry follows automatically, in the definition of interference-free trajectories,  $(x'_{t+1}, y'_{t+1}) \notin I(x_t, y_t)$  does not necessarily imply that  $(x_{t+1}, y_{t+1}) \notin I(x'_t, y'_t)$ .

As in Su et al. (2006a,b), in this work we focus on testing for catastrophic defects that result in a droplet getting stuck at the defective electrode. Such defects are caused, e.g., by dielectric breakdown causing a short between the droplet and the electrode. When this happens the droplet undergoes electrolysis, thereby preventing further movements of the droplet (see Chakrabarty and Su (2007) for a detailed review of DMFB defect types). Testing for such defects can be done by using one or more test droplets that collectively visit all the biochip’s cells, with the presence of defects being inferred from the failure of any of the test droplets to reach the output cells.

**DEFINITION 4.** A *feasible testing schedule* is a set of pairwise merge- and interference-free droplet trajectories  $\Pi = \{\pi_1, \dots, \pi_k\}$  that collectively visit all cells of  $R$ .

Since defects can arise during both manufacturing and field operation, DMFB testing is performed at multiple stages during the lifetime of a biochip – immediately following manufacturing, before the start of a new bioassay, and even during bioassay execution (see Chakrabarty and Su (2007) for further details). In the first two scenarios, referred to as *off-line testing*, all cells (often forming a rectangular array) are available for testing while in the latter, referred to as *concurrent testing*, some of the cells are used to store assay droplets and are thus not available for testing. Under both scenarios, testing cost is predominantly determined by the time needed to complete the testing schedule. Thus, the DMFB testing problem can be formalized as follows:

#### DIGITAL MICROFLUIDICS BIOCHIP TEST PROBLEM (DMFB-TEST)

**Given:** Biochip with cells  $R$ , input cells  $I \subset R$ , and output cells  $O \subset R$

**Find:** Number of test droplets  $k$  and feasible testing schedule  $\Pi = \{\pi_1, \dots, \pi_k\}$  with minimum completion time  $\max_{i=1, \dots, k} t_i$ , where  $t_i$  denotes the ending clock cycle for droplet trajectory  $\pi_i$ .

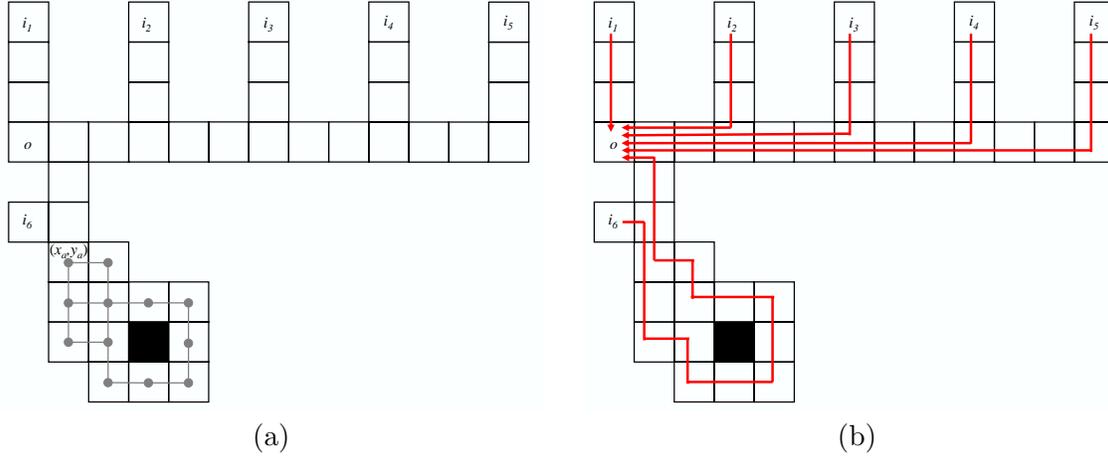
### 3. Computational Complexity

In this section we prove that DMFB-TEST is NP-hard via a reduction from HAMILTONIAN CIRCUIT IN GRID GRAPHS i.e., the problem of deciding whether an input grid graph has a Hamiltonian cycle or not, which was shown to be NP-complete by Itai et al. (1982). A *grid graph* is a finite, node induced subgraph of the infinite graph whose vertices are all points in the plane with integer coordinates, and whose edges connect all pairs of vertices corresponding to points at Euclidean distance 1.

**THEOREM 1.** DMFB-TEST is NP-Hard.

*Proof.* Given a grid graph  $G$  with  $n$  vertices we construct in polynomial time a DMFB-TEST instance  $R(G)$  with  $K = \lfloor \frac{n+6}{3} \rfloor$  input cells  $i_1, \dots, i_K$  and a single output cell  $o$ . First, we include in  $R(G)$  all  $n$  cells corresponding to vertices of  $G$ . Let  $(x_a, y_a)$  be the coordinates of the leftmost vertex of  $G$  among those with largest  $y$  coordinate, i.e.,  $y_a = \max\{y : \exists x \text{ s.t. } (x, y) \in G\}$  and  $x_a = \min\{x : (x, y_a) \in G\}$ . Input  $i_K$  is placed at  $(x_a - 1, y_a + 1)$  and output  $o$  is placed at  $(x_a - 1, y_a + 3)$ . We then add to  $R(G)$  two cells with coordinates  $(x_a, y_a + 1)$  and  $(x_a, y_a + 2)$ , and the  $n$  cells in  $o$ ’s horizontal row, immediately to its right. Input cells  $i_{K-1}, i_{K-2}, \dots, i_1$  are all placed in row  $y_a + 6$ , every third column (in decreasing order) starting from  $x_a + n - 1$ . Each of these inputs is connected to the closest cell in  $o$ ’s horizontal row by adding two cells in the respective columns. The construction of  $R(G)$  is illustrated for a grid graph  $G$  with  $n = 12$  in Figure 2(a).

We claim that  $R(G)$  admits a feasible testing schedule with completion time of  $n + 6$  if and only if  $G$  has a Hamiltonian cycle. Indeed, if  $G$  is Hamiltonian, a feasible testing schedule with this completion time is obtained by simultaneously generating a droplet out of each input cell at time  $t = 0$ . Droplets generated out of inputs  $i_1, \dots, i_{K-1}$  follow the unique simple paths to the output,



**Figure 2** (a) DMFB-TEST instance  $R(G)$  corresponding to a (Hamiltonian) grid graph  $G$  with  $n = 12$  vertices (shown in gray) (b) Feasible testing schedule of  $R(G)$  with completion time  $t = n + 6$ .

reaching  $o$  in that order at 3 cycle intervals ending at cycle  $n + 3$ . The droplet generated from  $i_K$  moves in two cycles to  $(x_a, y_a)$ , visits all vertices of  $G$  along a Hamiltonian cycle and returns to  $(x_a, y_a)$  in  $n$  more cycles, and finally moves to  $o$  in 4 additional steps, reaching it at  $t = n + 6$ . It is straightforward to verify that the  $K$  droplet trajectories are pairwise merge- and interference-free, and thus form a feasible testing schedule. A graphical representation of droplet trajectories for the DMFB-TEST instance in Figure 2(a) is given in Figure 2(b).

Conversely, assume that  $R(G)$  has a feasible testing schedule  $\Pi = \{\pi_1, \dots, \pi_k\}$  with completion time of  $n + 6$ . A simple case analysis shows that a droplet generated at input  $i_j$ ,  $1 \leq j \leq K - 1$  cannot visit all cells in the column of input  $i_{j'}$  with  $j' \neq j$  while still maintaining a completion time of  $n + 6$ . Thus, the schedule must use exactly one droplet out of each of the inputs  $i_1, \dots, i_{K-1}$ . Since these droplets keep the output cell “busy” at least until cycle  $n + 3$ , and at most one droplet can reach the output every 3 cycles due to interference constraints,  $\Pi$  must include exactly one droplet generated at  $i_K$  (i.e.,  $k = K$ ). Since this droplet reaches the output by time  $n + 6$ , it follows that its trajectory visits the  $n$  cells of  $G$  in exactly  $n$  cycles, thus defining a Hamiltonian cycle.

## 4. Off-line Testing of Rectangular Biochips

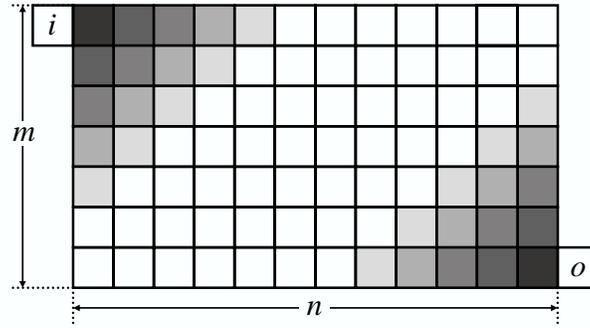
In this section we focus on algorithms for off-line testing of biochips for which  $R \setminus (I \cup O)$  forms a rectangle with  $m$  rows and  $n$  columns. Without loss of generality we assume that  $m \leq n$ . For concreteness we consider biochips with only one input cell  $i$  and only one output cell  $o$ , located at opposite corners of  $R$  in the top and bottom rows (see Figure 3). We first establish a lower bound on the completion time for off-line testing of arbitrarily-sized rectangular biochips, then describe and analyze several algorithms, some of which are proven to be optimal for particular chip sizes, including commonly used square-shaped biochips.

### 4.1. Lower Bound

**LEMMA 1.** *Let  $T_k^*$  denote the completion time of an optimum testing schedule using  $k$  droplets for an  $m \times n$  rectangular biochip. Then*

$$T_k^* \geq \begin{cases} m + n + 3k - 3 & \text{if } k \geq m \\ \frac{mn}{k} + 4k - 3 & \text{if } k < m \end{cases} \quad (1)$$

*Proof.* Assume that at the end of each cycle, up to and including the cycle in which the last droplet reaches the output cell, each droplet puts down one dollar in the cell it occupies. We will



**Figure 3** A rectangular biochip with input and output cells placed in opposite corners. Cells along main diagonals closest to the input/output cells are colored with different shades of gray. Each diagonal forms a cut separating the input and the output cells, and thus must be traversed by each of the  $k$  test droplets.

prove the lemma by counting the minimum total number of dollars that must be left on the chip by any feasible testing schedule.

First, note that, due to non-interference constraints, droplets cannot be generated at the input cell closer than 3 cycles apart. It follows that the input cell collects at least 3 dollars from the second departing droplet, at least 6 from the third, and, in general, at least  $3(l-1)$  dollars from the  $l^{\text{th}}$  departing droplet. Thus, the input cell collects at least  $3 + 6 + \dots + 3(k-1) = \frac{3k(k-1)}{2}$  dollars overall. Similarly, the output cell collects at least  $\frac{3k(k-1)}{2} + k$  dollars, with the  $k$  extra dollars coming from the fact that droplets pay upon arrival (but not upon departure).

Now consider the main diagonals of  $R$ , i.e., non-empty sets of the form  $D_j = \{(x, y) \in R \setminus \{i, o\} : y - x = j\}$  for some integer  $j$  (see Figure 3). For a fixed diagonal  $D_j$ , each one of the  $k$  droplets must visit  $D_j$  at least once. Furthermore, each cell of  $D_j$  must be visited at least once by some droplet. Thus, a diagonal  $D_j$  collects at least  $\max\{k, |D_j|\}$  dollars, and, collectively, the cells in  $R \setminus \{i, o\}$  collect at least  $\sum_j \max\{k, |D_j|\}$  dollars in any feasible testing schedule, where the sum is over all  $m+n-1$  non-empty diagonals  $D_j$ . If  $k \geq m$  then  $\max\{k, |D_j|\} = k$  for each such diagonal  $D_j$ . On the other hand, if  $k < m$  then  $\max\{k, |D_j|\} = k$  for the  $k$  diagonals closest to  $i$  and the  $k$  diagonals closest to  $o$ , while for the remaining diagonals  $\max\{k, |D_j|\} = |D_j|$ . Thus, the total number of dollars left on the chip by the droplets of a feasible testing schedule is

$$\geq \begin{cases} \frac{3k(k-1)}{2} + \left(\frac{3k(k-1)}{2} + k\right) + (m+n-1)k = (m+n+3k-3)k, & \text{if } k \geq m \\ \frac{3k(k-1)}{2} + \left(\frac{3k(k-1)}{2} + k\right) + 2k^2 + (mn - k(k+1)) = mn + 4k^2 - 3k, & \text{if } k < m \end{cases} \quad (2)$$

The lemma follows from (2) by observing that the total number of dollars paid by the droplets of the optimum schedule is exactly  $kT_k^*$ .

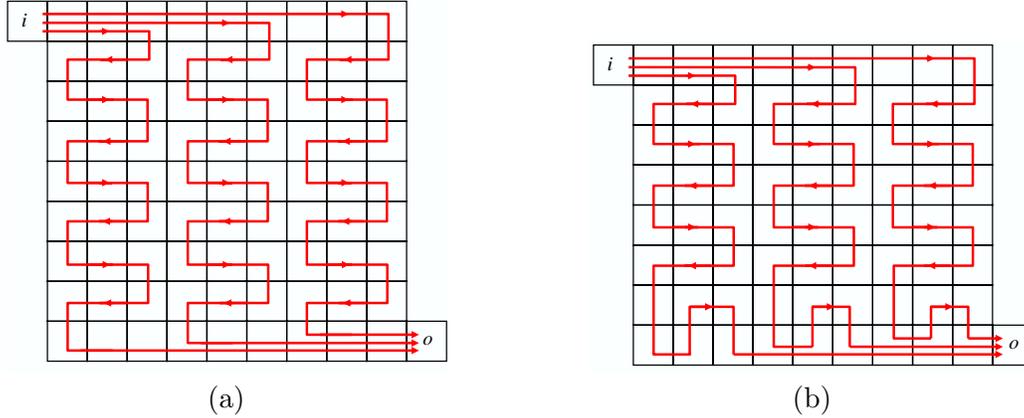
**THEOREM 2.** *Off-line testing of an  $m \times n$  rectangular biochip requires at least  $4\sqrt{mn} - 3$  cycles if the aspect ratio  $n/m$  is smaller than 4, and at least  $n + 4m - 3$  cycles otherwise.*

*Proof.* The theorem follows by minimizing the right-hand side of (1) over possible values of  $k$ . For biochips with aspect ratio smaller than 4 the minimum is achieved for  $k = \frac{\sqrt{mn}}{2}$ , while for aspects ratios of 4 or larger the minimum is achieved for  $k = m$ .

## 4.2. Vertical Stripes Algorithm

In this section we describe a simple algorithm based on partitioning the biochip into groups of 3 adjacent columns referred to as vertical stripes, and then using separate droplets to test the cells in each stripe. Partitioning the biochip into vertical stripes with more than 3 columns or horizontal stripes with 3 or more rows can be done in a similar manner, but results in the same or

greater completion time for  $m \leq n$ . We show that the vertical stripe algorithm achieves a constant approximation factor in the case of square biochips. Although its performance is always dominated by that of the interleaving-based algorithm in Section 4.4, the vertical stripes algorithm is of interest since it forms the basis for the concurrent testing heuristic in Section 5.2.



**Figure 4** Droplet trajectories generated by applying the vertical stripes algorithm to two biochips with odd (a), respectively even (b) number of rows.

For simplicity, we first detail the vertical stripes algorithm for the case when  $n$  is a multiple of 3. In this case the algorithm uses  $n/3$  droplets, which are generated from the input cell at 3 cycle intervals starting from  $t = 0$ . Each droplet moves to the right in the topmost row until it reaches the corresponding stripe, visits its cells, and then finally moves along the bottom row to the right until reaching the output cell (see Fig. 4). For an odd number of rows  $m = 2l + 1$ , the trajectory followed by the  $i^{\text{th}}$  droplet is described by the string  $R^{n-3(i-1)}(DL^2DR^2)^l R^{3(i-1)+1}$ , where  $x^r$  denotes the string obtained by concatenating  $r$  copies of string  $x$ . For an even number of rows  $m = 2l$  we need a small modification to the order in which the cells in the bottom two rows of each stripe are visited: instead of visiting these cells row by row we visit them column by column. Specifically, the trajectory followed by the  $i^{\text{th}}$  droplet is now described by the string  $R^{n-3(i-1)}(DL^2DR^2)^{l-2}DL^2D^2RURDR^{3(i-1)-1}$ .

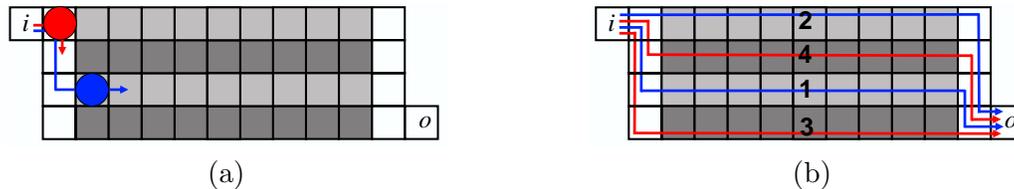
Note that generating the droplets 3 cycles apart ensures that there is no merging or interference between their trajectories. Indeed, in each cycle all droplets present on the chip are located in the same row, with 2 empty cells separating droplets of adjacent stripes. Regardless of the parity of  $m$ , the length of each droplet trajectory is  $n + 3m - 2$ . Assuming that the first droplet starts at cycle 0, it follows that the last droplet starts at cycle  $3(n/3 - 1) = n - 3$ . Thus, the completion time for the testing schedule generated by the algorithm is  $(n - 3) + (n + 3m - 2) = 2n + 3m - 5$ . The following theorem follows directly by using the lower-bound established in Theorem 2:

**THEOREM 3.** *For square biochips with  $m = n \equiv 0(\text{mod } 3)$  the vertical stripes algorithm has an approximation factor of  $5/4$ .*

The vertical stripes algorithm can be extended to the case of rectangular biochips for which the number of columns  $n$  is not a multiple of 3. In this case the algorithm uses  $\lceil n/3 \rceil$  droplets, with the first  $\lceil n/3 \rceil - 1$  visiting vertical stripes of width 3 and the last one visiting the leftmost  $n(\text{mod } 3)$  columns. The first  $\lceil n/3 \rceil - 1$  droplets visit their stripes using “snaking” trajectories similar to those employed in the case  $n \equiv 0(\text{mod } 3)$ . The departure of the last droplet is delayed a small number of cycles to avoid interference with the droplet preceding it. It can be verified that this still results in a completion time of less than  $2n + 3m$ .

### 4.3. Interleaved Rows Algorithm

Note that the vertical stripe algorithm in the previous section produces a testing schedule in which every droplet trajectory has the same length. We will call such a testing schedule *uniform*. We do not know if every rectangular biochip admits an optimum testing schedule that is uniform. However, in this section we show that this is the case for biochips with an even number of rows and aspect ratio greater than 4, for which a particularly simple algorithm, referred to as the *interleaved rows algorithm*, achieves optimality.



**Figure 5** (a) Interference occurs between two droplets generated 3 cycles apart if the second droplet (shown in red) must visit a row that is below or immediately above of the row visited by the first (shown in blue). (b) Trajectories and order of departure for the droplets used by the interleaved rows algorithm to test a biochip with 4 rows.

As the name suggests, the algorithm uses a different droplet to cover each biochip row (thus using  $k = m$  droplets in total). Droplets move from the input to the output along shortest possible trajectories of length  $n + m$ . Specifically, the trajectory of the droplet covering the  $j^{\text{th}}$  row from the bottom is described by the string  $RD^{m-j}R^{n-1}D^{j-1}R$ . To minimize completion time we would like to generate droplets at the maximum possible rate of one droplet every 3 cycles. Achieving this rate while avoiding droplet interference requires careful ordering of start times, as described below.

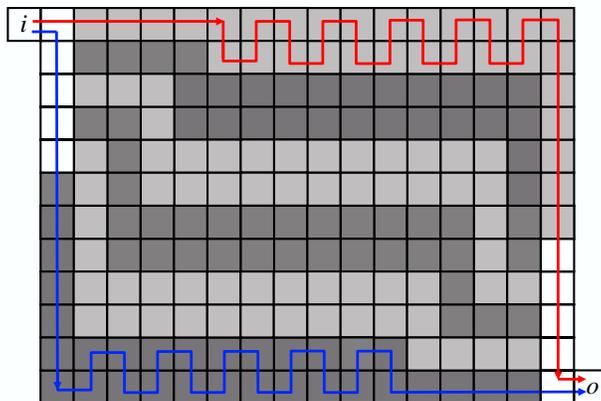
Consider two droplets generated at the input cell 3 cycles apart from each other. Note that, if the first droplet traverses row  $j$  from the bottom for some  $j < m$ , then the second one cannot visit a row  $j'$  with  $j' \leq j + 1$ , since its trajectory interferes with that of the first droplet when the latter attempts to move out of second column (see Figure 5(a)). This suggests scheduling droplet departures in an *interleaved* manner: the algorithm first generates, in bottom-to-top order, droplets corresponding to even rows, followed by droplets corresponding to odd rows, again in bottom-to-top order (see Figure 5(b) for an example). It is easy to verify that all droplets within a parity class can proceed without interference. If the number of rows  $m$  is even, then it is also the case that the first “odd” droplet does not interfere with the last “even” droplet even if they are generated 3 cycles apart. Thus, in this case the completion time for the testing schedule generated by the interleaved rows algorithm is  $3(m - 1) + (m + n) = n + 4m - 3$ . When the number of rows is odd, 4 cycles must be allowed between the departure of the last even droplet and that of the first odd droplet in order to avoid interference between their trajectories, resulting in a completion time of  $n + 4m - 2$ . From Theorem 2 we get:

**THEOREM 4.** *The interleaved rows algorithm produces off-line testing schedules with optimal completion time for rectangular biochips with aspect ratio of 4 or more and an even number of rows.*

### 4.4. Interleaved Zig-Zags Algorithm

For square biochips ( $m = n$ ), Theorem 2 implies a lower-bound of  $4m - 3$  on the optimum completion time. Since the lower-bound is achieved for  $k = m/2$ , we will assume that the number of rows  $m$  is even and seek optimal testing schedules with  $m/2$  droplets. A naïve approach is to partition the chip into horizontal stripes each consisting of two adjacent rows, and then use separate droplets to cover each stripe. This results in a uniform testing schedule with a droplet trajectory length of

$3m - 1$ . However, even at the maximum droplet departure rate of one droplet every 3 cycles, the resulting completion time of such a schedule would be  $3(m/2 - 1) + 3m - 1 = 4.5m - 2$ , which is suboptimal.



**Figure 6** Zig-zag partitioning and trajectories of the last “even” and first “odd” droplets on a  $12 \times 16$  biochip.

We next describe an improved “zig-zag” partitioning of rectangular chips that results in minimum testing time over all testing schedules with  $m/2$  droplets, and thus is optimal when applied to square chips. For simplicity, we will assume that the number of columns  $n$  has the same parity as  $m/2$ . The partition of a  $12 \times 16$  biochip is illustrated in Figure 6. Biochip regions covered by the  $m/2$  trajectories are disjoint apart from shared cells in the top half of leftmost column, respectively bottom half of rightmost column. The bulk of each trajectory consists of a  $2 \times (n - m/2 + 1)$  rectangle, connected via (possibly degenerate) zig-zag paths to the input and output cells. The  $j^{\text{th}}$  trajectory (counting from the bottom) is formally described by the string  $RD^{m/2-j}R^{j-1}D^{m/2-j}(DRUR)^{(n-m/2)/2}D^jR^{m/2-j}D^{j-1}R$ . Thus, each trajectory has length  $2n + m/2$ .

By using the same interleaving strategy as in the previous section (generating “even” droplets in bottom-to-top order, followed by “odd” droplets in bottom-to-top order) droplets can be generated at the maximum rate of one every 3 cycles without any interference. This results in completion time of  $3(m/2 - 1) + 2n + m/2 = 2n + 2m - 3$ , and from Theorem 2 we get:

**THEOREM 5.** *The interleaved zig-zags algorithm produces off-line testing schedules with optimal completion time for biochips with  $n = m \equiv 0 \pmod{4}$ .*

## 5. Concurrent Testing of Rectangular Biochips

Recall that in concurrent testing some of the biochip cells are not available for testing since they hold droplets undergoing various biochemical reactions. We will refer to connected groups of unavailable cells as *obstacles*. Because, as shown in Section 3, the concurrent testing problem is NP-Hard, we focus on deriving useful lower-bounds and heuristics for the practically important case of rectangular biochips with obstacles of small size. An integer programming formulation is given in the Appendix, but solution was possible only for very small problem instances, i.e., for chips of size smaller than  $10 \times 10$ .

### 5.1. Lower Bound with Obstacles

Lemma 1 extends to the case of rectangular biochips with obstacles in a straightforward way. As in the case without obstacles, each one of the  $n + m - 1$  diagonals  $D_j$  forms a cut separating  $i$  and  $o$ , and thus every droplet must pass through it. Using the same counting argument, it follows

that diagonal  $D_j$  must receive at least  $\max\{k, \|D_j\|\}$  dollars, where  $\|D_j\|$  denotes the number of *available cells* in diagonal  $D_j$ . Including the minimum number of dollars paid to the input and output cells, it follows that a global lower-bound on the total number of dollars paid during any feasible testing schedule with  $k$  droplets is

$$\frac{3k(k-1)}{2} + \left( \frac{3k(k-1)}{2} + k \right) + \sum_j \max\{k, \|D_j\|\}$$

Thus, the following bound holds for any biochip with blocked cells

$$T_k^* \geq 3k - 2 + \frac{1}{k} \sum_j \max\{k, \|D_j\|\} \quad (3)$$

## 5.2. A Generalized Vertical Stripes Heuristic

In this section we describe a natural extension of the vertical stripes algorithm given in Section 4.2 to chips that contain obstacles. For simplicity we assume that the top and bottom rows of the chip contain no blocked cells. Let  $Q$  be the maximum obstacle width. We begin by partitioning the biochip into non-overlapping vertical stripes of width  $W \geq Q + 1$ . As in the case of off-line testing, a different droplet is assigned to each stripe. Using stripes of width  $W \geq Q + 1$  ensures that each droplet can be used to cover all available cells in its stripe without having to move into adjacent stripes. Similar to the case of the vertical stripe heuristic for off-line testing, droplets are generated at  $w$  cycle intervals, and move to the right in the top row until they simultaneously reach their assigned stripes. Then, all the droplets start visiting in parallel all available cells in their respective stripes.

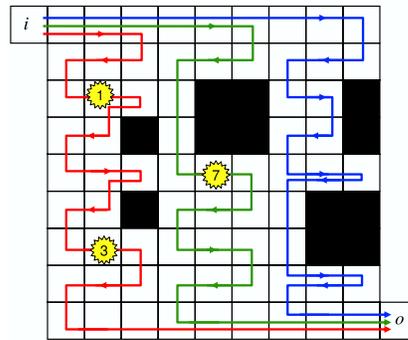
The trajectory of each droplet within its corresponding stripe is found using a two step procedure. First, all the available cells in the stripe are ordered by the time cycle in which they would have been visited by a virtual droplet had there been no obstacles (i.e., assuming off-line testing). The trajectory of the  $i^{th}$  virtual droplet comes from a direct generalization of the vertical stripes algorithm for off-line testing to stripes of width  $W$ :  $R^{n-W(i-1)}(DL^{W-1}DR^{W-1})^l R^{W(i-1)+1}$  if  $m = 2l + 1$  (regardless of the parity of  $W$ ),  $R^{n-W(i-1)}(DL^{W-1}DR^{W-1})^{l-2} DL^{W-1} D^2 (RURD)^w R^{W(i-1)-1}$  if  $m = 2l$  and  $W = 2w + 1$ , and  $R^{n-W(i-1)}(DL^{W-1}DR^{W-1})^{l-2} DL^{W-1} D^2 (RURD)^{w-1} RUDR^{W(i-1)-1}$  if  $m = 2l$  and  $W = 2w$ .

Second, a preliminary trajectory is computed such that all the available cells in the stripe are visited according to the previously computed order. That is, the preliminary trajectory is composed by concatenating shortest paths between consecutive cells in the ordering. Since the droplet is not allowed to exit its assigned stripe, the shortest path is restricted to available cells within the stripe. The existence of a such a path comes from the assumption that the maximum obstacle width is smaller than the stripe width  $W$ . Since the ordering of the cells follows (as much as possible) the trajectory of a droplet in the off-line testing settings, most of the consecutive cells in the ordering are neighbors and the shortest path is trivial. The only instances when consecutive cells are not neighbors are when an obstacle is present. Finding the shortest path in this case amounts to following the contour of the obstacle in either a clockwise or counter-clockwise direction.

Finally, the preliminary trajectories are combined into a feasible testing schedule by adjusting some of the trajectories to remove interferences. Note that, unlike the off-line version of the vertical stripes heuristic, a minimum 2 cell separation between any pair of droplets that follow the preliminary trajectories can no longer be guaranteed. However, interference can only occur at neighboring stripes when the two corresponding droplets access the rightmost, respectively leftmost, cells of their stripes. To address this, starting from the rightmost to the leftmost pair of adjacent stripes, we adjust the preliminary trajectory of the left droplet by forcing it to wait at most  $w$  cycles before entering the rightmost cell if there is interference with the right droplet. Forcing the left droplet to

wait before entering the rightmost cell ensures that the right droplet can continue on its trajectory without delay.

An example of the generalized stripe heuristic on a small chip of size  $9 \times 9$  with  $W = 3$  is shown in Figure 7. The algorithm starts by building trajectories for the 3 droplets following the two step procedure described above. Then, starting from right to left, potential interferences between all pairs of neighbor droplets are analyzed and waiting is forced on the left droplet when required. For example, at time cycle 16 the droplet in the middle stripe is forced to stay put for 7 cycles in order not to interfere with the rightmost droplet.



**Figure 7** Testing schedule produced by the generalized vertical stripes heuristic applied on a biochip of size  $9 \times 9$  with obstacles. Numbers represent the waiting cycles at the respective locations.

### 5.3. Empirical Results

We assessed the performance of the generalized vertical stripes (GVS) heuristic on a chip of size  $(m, n) = (99, 120)$ , in which we randomly designated up to 25% of its cells as unavailable for testing. For all experiments we used randomly located non-adjacent obstacles of size  $2 \times 2$ .

Table 2 gives the completion time obtained by running GVS with different stripe sizes, along with the lower bound (3) for the corresponding number of droplets. As expected, the completion time decreases dramatically when increasing the number of testing droplets. For example, using  $n/3 = 40$  droplets instead of a single droplet decreases the completion time by a factor of 20 to 30. The gain achieved by increasing the number of droplets is more pronounced when a larger percentage of the biochip area is occupied by obstacles. To understand why, note that obstacles have two opposite effects on completion time: on the positive side having more obstacles leaves fewer cells that need to be tested, while on the negative side more obstacles result in more detours for the test droplet trajectories generated by the GVS heuristic. The magnitude of negative effects relative to that of positive effects depends on the number of droplets: for small numbers of test droplets negative effects tend to dominate and completion time monotonically increases with the obstacle area. On the other hand, when stripe width becomes comparable with obstacle size, this trend is reversed, and completion time improves as the obstacle area increases. Naturally, the lower-bound (3) monotonically decreases with obstacle area regardless of the number of droplets, since the formula implicitly captures the reduced number of cells that need to be visited when obstacle area increases, but does not account in any way for droplet detours.

In Table 3 we contrast the results obtained by the GVS heuristic with stripes of width 3 ( $k = 40$ ) with the overall lower bound obtained by minimizing (3) over all possible values of  $k$ . Over the tested range of 0-25% obstacle area, the average completion time achieved by GVS solutions is within 23-34% of the overall lower-bound, suggesting that there is relatively little room for further improvements.

Stripe Width(Q)	Number of Droplets(k)		Obstacle Area			
			0%	5%	10%	25%
120	1	GVS	11880.0	12843.0	13798.2	15275.1
		Lower Bound	11880.0	10900.0	9920.0	8386.0
		Ratio	1.00	1.18	1.39	1.82
12	10	GVS	1211.0	1303.6	1370.3	1480.1
		Lower Bound	1034.0	953.2	872.3	747.1
		Ratio	1.17	1.37	1.57	1.98
6	20	GVS	823.0	872.3	901.3	936.8
		Lower Bound	670.0	622.4	575.2	503.0
		Ratio	1.23	1.40	1.57	1.86
4	30	GVS	629.0	652.8	658.4	667.2
		Lower Bound	512.0	481.7	452.1	407.6
		Ratio	1.23	1.36	1.46	1.64
3	40	GVS	532.0	536.8	528.0	509.0
		Lower Bound	453.0	431.9	411.5	382.0
		Ratio	1.17	1.24	1.28	1.33

**Table 2** Completion time achieved by the GVS heuristic for various stripe widths (results averaged over 10 seeds).

	Obstacle Area			
	0%	5%	10%	25%
GVS with stripe width 3 ( $k = 40$ )	532.0	536.8	528.0	509.0
Overall lower bound (variable $k$ )	432.0	418.3	404.0	380.7
Best $k$ in overall lower bound	54	52	48	43
Ratio	1.23	1.28	1.31	1.34

**Table 3** Completion time achieved by the GVS heuristic for stripes of width 3 versus the overall lower bound (results averaged over 10 seeds).

## 6. Conclusions and Open Problems

In this paper we formalized the DMFB testing problem including previously ignored constraints of droplet non-interference, and presented both optimal and approximation algorithms for off-line testing and a heuristic producing near-optimal results for concurrent testing. In particular, we gave optimal algorithms for off-line testing of square biochips and biochips with aspect ratio of 4 or more (under some mild assumptions on the divisibility of biochip dimensions). Finding optimal off-line testing for chips with aspect ratio between 1 and 4 is an interesting open problem.

Another interesting direction of future research is to find optimal algorithms for “edge” testing of DMFBs. As noted by Su et al. (2007), a short-circuit between two adjacent electrodes causes a droplet to get stuck if the test droplet trajectory visits the two electrodes in sequence, but remains undetected otherwise. Single-droplet test schedules guaranteed to detect all such edge defects correspond to Eulerian paths in the grid graph representing the biochip (see Su et al. (2007)), but such schedules are too slow to be used for large biochips. The “parallel scan-like” test algorithm proposed by Xu and Chakrabarty (2007) achieves a completion time of  $O(n)$  by using  $O(n)$  droplets to test an  $n \times n$  biochip. While  $O(n)$  is the best possible asymptotic result, it is unclear if the algorithm of Xu and Chakrabarty (2007) is in fact optimal since overhead due to droplet interference is ignored in their analysis.

### Acknowledgments

Work supported in part by NSF awards IIS-0546457, IIS-0916948, and DBI-0543365.

## References

- Akella, S., S. Hutchinson. 2002. Coordinating the motions of multiple robots with specified trajectories. *Proc. IEEE International Conference on Robotics and Automation*, 624–631.
- Angel, R., W. Caudle, R. Noonan, K. Whinston. 1972. Computer-assisted school bus scheduling. *Management Science* **18**, 279–288.
- Bohringer, K. F. 2004. Towards optimal strategies for moving droplets in digital microfluidic systems. *Proc. IEEE International Conference on Robotics and Automation*, 1468–1474.
- Chakrabarty, K., F. Su. 2007. *Digital Microfluidic Biochips: Synthesis, Testing and Reconfiguration Techniques*, CRC Press.
- Christofides, C. 1985. Vehicle Routing. In Lawler, L., J. Lenstra and A. Rinnooy Kan, D. Shmoys, eds., *The Traveling Salesman Problem*, John Wiley, 431–448.
- Frizzel, P., J. Griffin. 1995. The split delivery vehicle scheduling problem with time windows and grid network distances. *Computers and Operations Research* **22**, 655–667.
- Griffith, E. J., S. Akella. 2005. Coordinating multiple droplets in planar array digital microfluidic systems. *International Journal of Robotics Research* **24**, 933–949.
- Griffith, E. J., S. Akella, M. K. Goldberg. 2006. Performance characterization of a reconfigurable planar array digital microfluidic system. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems* **25**, 340–352.
- Itai, A., C. Papadimitriou, J. Szwarcfiter. 1982. Hamilton paths in grid graphs. *SIAM J. on Computing* **11**, 676–686.
- Kerckhoff, H. G. 2007. Testing microelectronic biofluidic systems. *IEEE Design and Test of Computers* **24**, 72–82.
- Peng, J., S. Akella. 2005. Coordinating multiple robots with kinodynamic constraints along specified paths. *International Journal of Robotics Research* **24**, 295–310.
- Pollack, M.G., A.D. Shenderov, R.B. Fair. 2002. Electrowetting-based actuation of droplets for integrated microfluidics. *Lab on a chip* **2**, 96–101.
- Rubrico, J., J. Ota, H. Tamura, M. Akiyoshi, T. Higashi. 2004. Route generation for warehouse management using fast heuristics. *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2093–2098.
- Savchenko, M., E. Frazzoli. 2005. On the time complexity of conflict-free vehicle routing. *Proc. American Control Conference*, 3536–3541.
- Su, F., K. Chakrabarty, V.K. Pamula. 2005. Yield enhancement of digital microfluidics-based biochips using space redundancy and local reconfiguration. *Proc. Conference on Design, Automation and Test in Europe*, 1196–1201.
- Su, F., W. Hwang, A. Mukherjee, K. Chakrabarty. 2007. Testing and diagnosis of realistic defects in digital microfluidic biochips. *J. Electronic Testing* **23**, 219–233.
- Su, F., S. Ozev, K. Chakrabarty. 2006a. Concurrent testing of digital microfluidics-based biochips. *ACM Trans. on Design Automation of Electronic Systems* **11**, 442–464.
- Su, F., S. Ozev, K. Chakrabarty. 2006b. Test planning and test resource optimization for droplet-based microfluidic systems. *J. Electronic Testing* **22**, 199–210.
- Toth, P., D. Vigo. 2002. *The vehicle routing problem*. SIAM Monographs on Discrete Mathematics and Applications.
- Xu, T., K. Chakrabarty. 2007. Parallel scan-like test and multiple-defect diagnosis for digital microfluidic biochips. *IEEE Trans. on Biomedical Circuits and Systems* **1**, 148–158.

## Appendix. Mixed Integer Programming Formulation of DMFB-TEST

In this appendix we present a mixed integer programming formulation of the DMFB-TEST problem. The formulation uses only one continuous variable,  $z$ , representing the completion time, and binary decision

variables that model the *transitions* made by the droplets in each cycle. We also explored formulations in which binary decision variables were used to model the *positions* occupied by droplets at the end of each cycle, however experimental results showed that such formulations are computationally less efficient.

### Notations

$N(i, j)$ : the set of usable cells sharing an edge with cell  $(i, j)$

$\bar{N}(i, j) = N(i, j) \cup \{(i, j)\}$

$M(i, j)$ : the merging region of cell  $(i, j)$

$\mathcal{M} = \max_{(i,j) \in R} |M(i, j)|$ : the largest size of a merging region

$I(i, j)$ : the interference region of cell  $(i, j)$

$\mathcal{I} = \max_{(i,j) \in R} |I(i, j)|$ : the largest size of an interference region

$T$ : an upper bound on the maximum completion time, e.g., determined by using a heuristic such as that presented in Section 5.2

### Decision Variables

$x_{(i,j)(k,l)}^t = \begin{cases} 1 & \text{if a droplet moves from cell } (i, j) \text{ to cell } (k, l) \text{ in the time interval } [t, t + 1] \\ 0 & \text{otherwise} \end{cases}$

$z$ : the cycle when the last droplet leaves the chip

### Formulation

Minimize  $z$  (4)

$$\sum_{(i,j) \in R \setminus I} \sum_{(k,l) \in \bar{N}(i,j)} x_{(k,l)(i,j)}^0 = 0 \quad (5)$$

$$\sum_{(i,j) \in R \setminus I} \sum_{(k,l) \in N(i,j) \setminus O} x_{(i,j)(k,l)}^T = 0 \quad (6)$$

$$\sum_{t=1}^T \sum_{(k,l) \in \bar{N}(i,j)} x_{(k,l)(i,j)}^t \geq 1, \quad \forall (i, j) \in R \quad (7)$$

$$\sum_{(k,l) \in \bar{N}(i,j)} x_{(k,l)(i,j)}^t - \sum_{(k,l) \in \bar{N}(i,j)} x_{(i,j)(k,l)}^{t+1} = 0, \quad \forall (i, j) \in R, 0 \leq t < T \quad (8)$$

$$\mathcal{M} \cdot \sum_{(k,l) \in \bar{N}(i,j)} x_{(k,l)(i,j)}^t + \sum_{(k,l) \in M(i,j)} \sum_{(k',l') \in N(k,l)} x_{(k',l')(k,l)}^t \leq \mathcal{M}, \quad \forall (i, j) \in R, 0 \leq t < T \quad (9)$$

$$\mathcal{I} \cdot \sum_{(k,l) \in N(i,j)} x_{(i,j)(k,l)}^t + \sum_{(k',l') \in I(i,j)} \sum_{(k'',l'') \in N(k',l')} x_{(k'',l'')(k',l')}^t \leq \mathcal{I}, \quad \forall (i, j) \in R, 0 \leq t < T \quad (10)$$

$$z - tx_{(i,j)(k,l)}^t \geq 0, \quad \forall (k, l) \in O, 0 \leq t < T \quad (11)$$

$$z \geq 0, \quad x_{(i,j)(k,l)}^t \text{ binary}, \quad \forall (i, j) \in R, (k, l) \in \bar{N}_{i,j}, 0 \leq t < T \quad (12)$$

The constraints (5), (6) force the chip to be free of droplets both initially and at time  $T$ . Every cell is visited at least once by (7), while droplet conservation is enforced by (8). Merging and interference are prohibited by (9) and (10), respectively. Finally the desired definition of the “completion-time” variable  $z$  is enforced by (11).

The formulation has a number of variables and constraints of approximately  $5|R|T$  and  $3|R|T$ , respectively. For an  $m \times n$  rectangular biochip, the number of variables and constraints is  $5nmT$  and  $3nmT$ . The large size makes it infeasible to solve the formulation for all but very small biochips. In addition to the size of the model the minmax objective leads to very poor fractional relaxation lower bounds. In other words, the optimal fractional solution satisfies (11) by assigning very small values to many variables, resulting in an objective value that is very far from that of the optimum integer solution.