



Test Planning and Test Resource Optimization for Droplet-Based Microfluidic Systems*

FEI SU, SULE OZEV AND KRISHNENDU CHAKRABARTY

Department of Electrical & Computer Engineering, Duke University, Durham, NC, 27708

fs@ee.duke.edu

sule@ee.duke.edu

krish@ee.duke.edu

Received September 30, 2004; Revised March 29, 2005

Editor: C. Landrault

Abstract. Recent years have seen the emergence of droplet-based microfluidic systems for safety-critical biomedical applications. In order to ensure reliability, microsystems incorporating microfluidic components must be tested adequately. In this paper, we investigate test planning and test resource optimization for droplet-based microfluidic arrays. We first formulate the test planning problem and prove that it is NP-hard. We then describe an optimization method based on integer linear programming (ILP) that yields optimal solutions. Due to the NP-hard nature of the problem, we develop heuristic approaches for optimization. Experimental results indicate that for large array sizes, the heuristic methods yield solutions that are close to provable lower bounds. These heuristics ensure scalability and low computation cost.

Keywords: droplet-based microfluidic systems, concurrent testing, microfluidic arrays, test planning, test resource optimization

1. Introduction

Next-generation system-on-chip designs are expected to be composite microsystems with microelectromechanical and microfluidic components [15, 23]. These mixed-signal and mixed-technology systems monolithically integrate microelectronics with microsensors and microactuators, thereby leading to chips that can not only compute and communicate, but also sense and actuate. This high level of integration is enabling a new class of microsystems targeted at health care, environmental monitoring, biomedical analysis, harmful agent detection for countering bio-terrorism, and precision fluid dispensing [13].

In recent years, novel droplet-based microfluidic systems have been developed to analyze nanoliter volumes of agents [18]. These systems reduce the rate of reagent consumption, thereby enabling continuous sampling and analysis for on-line, real-time biological/chemical analysis. By scaling down the concentration of the samples, simple sensing tech-

niques can be utilized to replace conventional, costly, and time-consuming practices involving batch analysis, sample pre-treatment and frequent calibration. Droplet-based microfluidic systems therefore offer a promising platform for massively parallel DNA analysis, and real-time molecular detection and recognition.

As microfluidic systems become widespread in safety-critical biomedical applications, system reliability emerges as an essential performance parameter. In order to ensure reliability, composite microsystems incorporating microfluidic components must be tested adequately. Therefore, there is a pressing need for efficient test methodologies for these microsystems. The ITRS 2003 document recognizes the need for new test methods for disruptive device technologies that underly microelectromechanical systems and sensors, and highlights it as one of the five difficult test challenges beyond 2009 [27].

Recently, a fault classification and a unified test methodology for droplet-based microfluidic systems has been developed [22]. Faults are classified as either catastrophic or parametric, and they are detected by electrostatically controlling and tracking droplet motion. This cost-effective test methodology facilitates concurrent testing, which allows

*This research was supported in part by the National Science Foundation under grant number IIS-0312352. A preliminary version of this paper appeared in *Proc. European Test Symposium*, pp. 72–77, 2004

fault testing and biomedical assays to run simultaneously on a microfluidic system. Test planning and test resource optimization are motivated by the need for concurrent testing.

In this paper, we investigate test planning and test resource optimization problems for droplet-based microfluidic arrays. We first formulate the test planning problem and prove that it is NP-hard. We then show how optimal solutions can be obtained using integer linear programming (ILP). Due to the NP-hard nature of the problem, the ILP model is not applicable to large microfluidic arrays. We therefore develop heuristics to solve this problem in a computationally efficient manner. Experiments show that for large array sizes, the results obtained from the heuristic method are close to provable lower bounds.

The organization of the remainder of the paper is as follows. In Section 2, we present an overview of droplet-based microfluidic systems. Related prior work is discussed in Section 3. Section 4 describes the problem of test planning and test resource optimization. This problem is shown to be NP-hard in Section 5. An optimal solution based on integer linear programming is proposed in Section 6. Section 7 presents several heuristic algorithms, which are evaluated through simulation experiments in Section 8. Finally, conclusions are drawn in Section 9.

2. Background: Droplet-Based Microfluidic Systems

The operation of droplet-based microfluidic systems is based on the principle of electrowetting actuation. By varying the electrical potential along a linear array of electrodes, electrowetting can be used to move liquid droplets of nanoliter volume along this line of electrodes [18]. Droplets can also be transported, in user-defined patterns under clocked-voltage control, over a two-dimensional array of electrodes without the need for pumps and valves.

The basic component of a droplet-based microfluidic system is shown in Fig. 1. The droplet, usually containing biomedical samples, and the filler medium, such as silicone oil, are sandwiched between two parallel glass plates. The bottom plate contains a patterned array of individually controllable electrodes, while the top plate is coated with a ground electrode. The hydrophobic dielectric insulator is added to the top and bottom plates to decrease the wettability of the surface and to add capacitance between the droplet and the control electrode.

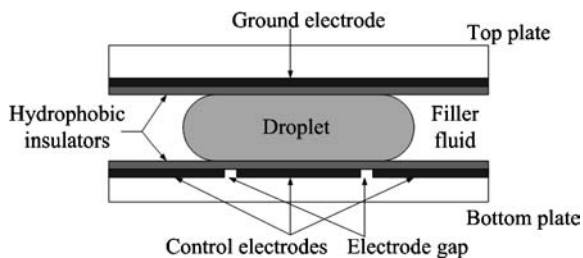


Fig. 1. Basic component of a droplet-based microfluidic system.

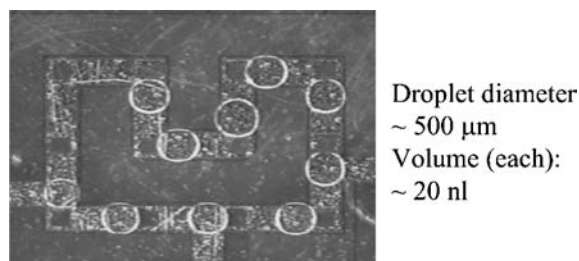


Fig. 2. Droplet transport in a two-dimensional array.

The basic principle underlying droplet transportation is the electrostatic control of the interfacial tension at the droplet/insulator interface. A control (actuating) voltage is applied to an electrode adjacent to the droplet and, at the same time, the electrode just under the droplet is deactivated. This causes an accumulation of charge in the droplet/insulator interface, resulting in an interfacial tension gradient across the gap between the adjacent electrodes, which consequently causes the transportation of the droplet. The velocity of the droplet can be controlled by adjusting the actuation voltage (0–90V), and droplets can be moved at speeds of up to 20 cm/s. Based on this principle, microfluidic droplets can be moved freely to any location of a two-dimensional array; see Fig. 2. This design, which has been fabricated on PCBs at Duke University [18], is ideally suited for a large-scale integrated microfluidic system. Such a system is expected to be common in the near future for various biomedical applications, such as DNA sequencing and bimolecular detection. A droplet can be easily detected using the capacitive sensing circuit shown in Fig. 3.

Using a two-dimensional microfluidic array, many common operations for different biomedical assays can be performed, such as sample introduction (*dispense*), sample movement (*transport*), temporarily sample preservation (*store*), and mixing of different samples (*mix*). Note that these operations can be performed anywhere on the array, whereas in continuous-flow systems they must operate in a specific micromixer or microchamber. The configurations of the microfluidic array, i.e., the routes that sample droplet travel and the rendezvous points of droplets, can be obtained using software running on a PC or an ASIC [20, 21]; they are then programmed into a microcontroller that controls the voltages of the electrodes in the array. Test planning for

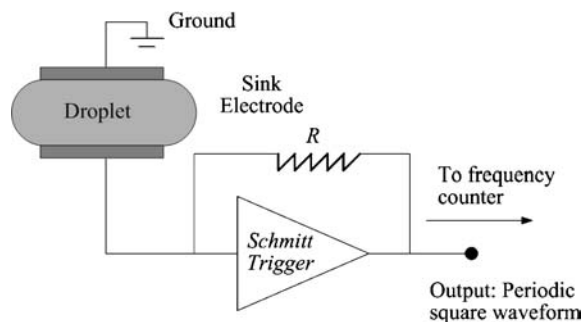


Fig. 3. Simple capacitive sensing circuit.

a microfluidic array can also be implemented using a PC or an ASIC.

3. Related Prior Work

Over the past decade, the focus in testing research has broadened from logic and memory test to include the testing of analog and mixed-signal circuits. MEMS is a relatively young field compared to IC design, and MEMS testing is still in its infancy. Recently, fault modeling and fault simulation in surface micromachined MEMS have received attention [4, 11, 14]. Researchers at Carnegie Mellon University are developing a comprehensive testing methodology for a class of MEMS known as surface micromachined sensors.

However, test techniques for MEMS cannot be directly applied to microfluidic systems, since the techniques and tools currently in use for MEMS testing do not handle fluids. Hence they are of limited use for testing microfluidic devices. Most recent work in this area has been limited to the testing of continuous-flow microfluidic systems [8, 9, 10]. Researchers at the MESA+ Research Institute of the University of Twente have applied mixed-signal testing techniques to the problem of testing a microanalysis system. Also, a design-for-testability (DFT) technique for Flow-FET-based microfluidic systems has been proposed [9]. Similar to the MOSFET, a Flow-FET has source and drain electrodes over which a relatively large voltage (~ 100 V) is applied. Due to the principle of electro-osmotic flow, the electric field moves the charge accumulated between the fluid and the surface of channel, dragging the bulk liquid through the channel.

Optimal strategies for moving droplets in a microfluidic system are proposed in [3]. The A* algorithm from artificial intelligence is used as the basis of a systematic search, which is performed to generate a sequence of control signals for moving one or multiple droplets from the start to the goal positions in the shortest number of steps. This method is closely related to the optimization problem of motion planning with multiple moving robots [1, 12]. There are two different groups of path planning problems for moving robots. Navigation problem attempts to find a path from a start position to a goal position through the shortest path, whereas coverage problem focuses on finding the path of coverage of an environment by mobile robots.

4. Problem Definition

In the test methodology proposed in [22], test stimuli droplets are dispensed into the microfluidic system from the droplet source and transported through the array (traversing the cells) by following the designed testing scheme. As described in [22], most catastrophic faults in droplet-based microfluidic systems can lead to a complete cessation of droplet transportation. Thus, for the faulty case, the test stimuli droplet is stuck at an intermediate point during mo-

tion. On the other hand, the detection of all test stimuli droplets at the droplet sinks indicates fault-free operation. This methodology allows fault testing and biomedical assays to run concurrently on a microfluidic system. An efficient test plan not only ensures that the testing operation does not conflict with the normal biomedical assay, but it also guides test stimuli droplets to cover all the cells available for testing. This test plan can be optimized to minimize the total testing time cost for a given test hardware overhead, which refers here to the number of droplet sources and droplet sinks. Note that some faults such as electrode shorts affect two adjacent electrodes [19, 22]. To detect such faults, defect-oriented test procedures are required, which focus on pairs of cells and the traversal of droplets from one cell to all its neighbors [19]. For simplicity, we do not take into account such types of faults in this paper; only catastrophic faults related to a single cell are targeted.

We can formulate the test planning problem in terms of graph partitioning and the Hamiltonian path problem from graph theory [5]. The key idea underlying this optimization approach is to model the two-dimensional microfluidic array as a directed graph, and then partition it into non-overlapping subgraphs. Each part of the microfluidic array is represented by a subgraph that is tested concurrently and independent of the other parts. In this way, the total test application time is reduced.

First we model the array of microfluidic cells using a directed graph $G = (V, E)$ where the set of vertices V represents the set of available microfluidic cells, droplet sources and droplet sinks, and $e_{ij} \in E$ is a directed edge from vertex i to vertex j if and only if these two vertices represent two adjacent microfluidic cells and they satisfy the criterion described below.

Note that unlike V , E is not determined *a priori*; rather the set of edges is a variable, and the edges are determined through the optimization procedure.

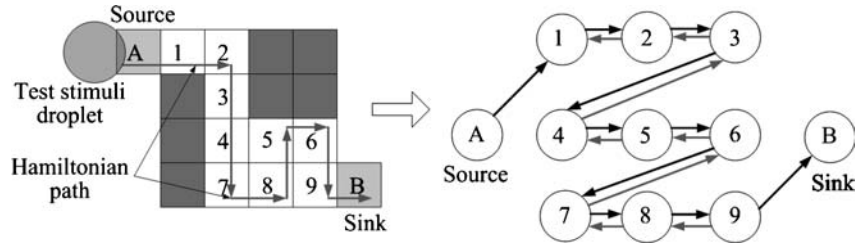
Definition 1. A Hamiltonian path from vertex s to vertex t in a graph G is a path that starts at vertex s , ends at vertex t , and visits every vertex of G exactly once.

We define e_{ij} as follows:

$$e_{ij} = \begin{cases} 1 & \text{if a Hamiltonian path from a droplet source to a} \\ & \text{droplet sink includes vertex } i \text{ and vertex } j \\ & \text{in consecutive order} \\ 0 & \text{otherwise} \end{cases}$$

If a Hamiltonian path exists in an array with n cells, then for any cell i in the array, $\sum_{j=1}^n e_{ij} = \sum_{j=1}^n e_{ji} = 1$.

The problem of finding a Hamiltonian path in graph G from one source to one sink can be expressed as the following problem: find a numerical instance of the set of binary variables $E = \{e_{ij}\}$, e.g., $\{e_{12} = 1, e_{21} = 0, \dots, e_{ij} = 1, \dots\}$, that represents a Hamiltonian path from one source to one sink.

Fig. 4. Graph model for a 4×4 array.

If a Hamiltonian path exists, the cost C for this path is defined as $C = \sum_{i=1}^n \sum_{j=1}^n e_{ij} w_{ij}$, where i represents any vertex in this path, j is the vertex adjacent to i in the path, and w_{ij} is the weight of e_{ij} . Without loss of generality, we set w_{ij} to be a constant value, assuming that the transportation velocity between any two adjacent microfluidic cells is the same. For simplicity, let $w_{ij}=1$. Therefore, $C = \sum_{i=1}^n \sum_{j=1}^n e_{ij} = \sum_{i=1}^n 1 = n$, i.e., the number of vertices on the Hamiltonian path. If G has no Hamiltonian path, the cost C is infinite.

Figure 4 gives an example of a graph model for single source and single sink. In the graph model of this 4×4 array, a black arrow between vertices i and j denotes that $e_{ij} = 1$, while the gray arrow between vertices i and j denotes that $e_{ij} = 0$. The cost C for this example is 11.

Based on the above definitions, we now develop the test planning problem for multiple sources and multiple sinks. We attempt to partition the directed graph representing the microfluidic array into subgraphs, such that in each subgraph there exists a Hamiltonian path from one source to one sink. In this way, the testing of the different partitions can be performed independently and simultaneously in non-overlapping parts of the microfluidic array. The total cost for the array is the maximum of the cost for any of these subgraphs. This leads us to the following optimization problem for minimizing the total cost:

- **Optimal Partitioning Problem (OPP):** Given N source/sink pairs, determine an optimal partition that divides the available cells in the array into N non-overlapping partitions, such that in each partition there exists a Hamiltonian Path from one source to one sink and the maximum of the cost for these Hamiltonian paths is minimized.

5. Analysis of Computational Complexity

In this section we prove that OPP is NP-hard. We first review the following definition from computational complexity theory:

Definition 2 [17]. Let L1 and L2 be two decision problems. L1 is polynomial-time reducible to L2 ($L1 \leq L2$) if a polynomial-time reduction f from L1 to L2 exists, subject to

- $f(x)$ is a yes-input for L2 if and only if x is a yes-input for L1
- f is computable in polynomial-time.

We next note that if L1 is NP-complete, and $L1 \leq L2$, then L2 is NP-hard. This is a common technique to prove that a given optimization problem is NP-hard.

We first consider the decision version D-PP of OPP, which is expressed as follows.

- **D-PP:** Given N source/sink pairs and an upper limit D on the cost, is it possible to partition array into N parts such that there exists a Hamiltonian path of cost C_i for each partition and $\max_{1 \leq i \leq N} \{C_i\} < D$?

Theorem 1. *OPP is NP-hard.*

Proof: We first show that D-PP \in NP. We can non-deterministically generate a N -partition and then verify in polynomial time that $\max_{1 \leq i \leq N} \{C_i\} < D$. To show that D-PP is NP-hard, we reduce the problem of determining a Hamiltonian cycle in grid graph (HC-GG), which is known to be NP-complete [7]. A grid graph G is a finite, induced subgraph of the infinite two-dimensional grid. It has a finite set of vertices $V = \{v_1, v_2, \dots, v_n\}$, where v_i represents a grid point (x, y) . Note that x and y are positive integers, denoting the x and y coordinates, respectively. An edge exists in G between point (x, y) and (x', y') if and only if $|x - x'| + |y - y'| = 1$.

We next define a polynomial-time reduction f from an arbitrarily-chosen instance of HC-GG to an instance of D-PP with $N = 1$ and $D = \infty$. Given a grid graph G , any vertex v_i in G is mapped to a cell c_i in array A , such that cell $c_i = f(v_i)$ and $c_j = f(v_j)$ are adjacent in A if and only if there exists an edge between v_i and v_j in G . We define the vertices with the maximum (or minimum) value x of the x -coordinate (or the y -coordinate y) in the corresponding grid graph to be boundary vertices in G . Similarly, the cells in the array obtained by mapping from the boundary vertices in G are defined as boundary cells in A . Next we attempt to add a droplet source s_1 and a droplet sink s_2 to this array. There are two possible cases. In Case 1, there exist two adjacent boundary vertices (noted as v_1 and v_n) in G , such that there also exist two adjacent cells (noted as c_1 and c_n) on the boundary of array A . We then add s_1 next

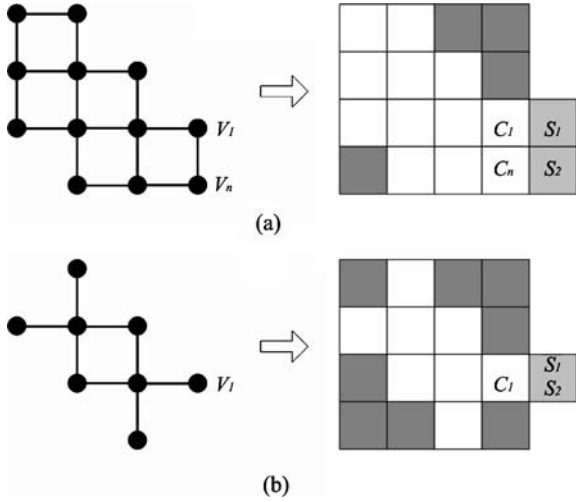


Fig. 5. (a) Illustration of Case1; (b) Illustration of Case2.

to c_1 and s_2 next to c_n ; see Fig. 5(a). In Case 2, if there are no adjacent boundary vertices in G and neither are there adjacent boundary cells in A , we select a single boundary cell denoted by c_1 , and place s_1 and s_2 together adjacent to c_1 ; see Fig. 5(b). It is obvious that the transformation described above can be carried out in polynomial time.

Next we prove that there exists a Hamiltonian path from s_1 to s_2 of cost $C < \infty$ in A if and only if there exists a Hamiltonian cycle in G of cost less than ∞ .

1. Proof for Case 1: Assume there exists a Hamiltonian cycle in G , denoted by $v_1 v_2 \dots v_n v_1$, where v_1 and v_n are two adjacent boundary vertices. Due to the mapping $f: G \rightarrow A$, $c_1 = f(v_1)$, $c_n = f(v_n)$ and they are two adjacent cells on the boundary of array A . In this way, there exists a path $f(v_1)f(v_2) \dots f(v_n)$ from c_1 to c_n that visits every cell exactly once. In addition, s_1 is adjacent to c_1 and s_2 adjacent to c_n . Therefore, there is a Hamiltonian path from s_1 to s_2 in A and cost $C = n < \infty$.

On the other hand, if there exists a Hamiltonian path $s_1 c_1 \dots c_n s_2$ from s_1 to s_2 in array A , a Hamiltonian path from c_1 to c_n also exists. Now by the inverse transformation $f^{-1}: A \rightarrow G$, it is seen that there exists a Hamiltonian path $f^{-1}(c_1) \dots f^{-1}(c_n)$ from $f^{-1}(c_1)$ to $f^{-1}(c_n)$. Moreover, $f^{-1}(c_1)$ and $f^{-1}(c_n)$ are two adjacent vertices. Therefore, there exists a Hamiltonian cycle $f^{-1}(c_1) \dots f^{-1}(c_n) f^{-1}(c_1)$ in G .

2. Proof for Case 2: If there exist no adjacent cells on the boundary of A , we place s_1 and s_2 together next to one boundary cell c_1 . This implies that in any path from s_1 to s_2 , c_1 is visited at least twice. Therefore, there exists no Hamiltonian path in A for this case and $C = \infty$. Similarly in G , since there are no adjacent vertices on the boundary, some boundary vertices have only degree one. This violates the necessary condition for the existence of a Hamiltonian cycle, i.e., every node should have a degree of at least two. Hence there is also no Hamiltonian cycle in G .

Thus we have shown that any instance of HC-GG is polynomial-time reducible to an instance of D-PP ($N =$

1 and $D = \infty$). Since HC-GG is NP-complete, D-PP is at least NP-hard. Moreover, since D-PP is in NP, it is also NP-complete. The optimization version of D-PP, i.e. the Optimal Partitioning Problem is therefore NP-hard. \square

6. Integer Linear Programming Model for OPP

Although OPP has been proven in Section 5 to be NP-hard, we show in this section that it can be solved exactly using integer linear programming (ILP) for a microfluidic array of modest size. An ILP model can be described as follows:

$$\begin{aligned} \text{Minimize: } & \mathbf{Ax} \text{ (objective function)} \\ \text{Subject to: } & \mathbf{Bx} \leq \mathbf{C} \text{ (constraint inequalities),} \end{aligned}$$

where x is a vector of variables, \mathbf{A} is an objective function vector, \mathbf{B} is a constraint matrix and \mathbf{C} is a column vector of constraints. We used a popular public domain ILP solver called *lpsolve* for our work [2].

We formulate the ILP model for OPP as follows. It is obvious that when $N = 1$, OPP is equivalent to the Hamiltonian path problem for a single source and a single sink described in the earlier section.

For $N > 1$, we define a binary variable S_{ik} as follows:

$$S_{ik} = \begin{cases} 1 & \text{if vertex } i \text{ is in subgraph } k, \text{ i.e., microfluidic} \\ & \text{cell } i \text{ belongs to partition } k. \\ 0 & \text{otherwise} \end{cases}$$

where $1 \leq k \leq N$. Since every vertex only belongs to one subgraph, $\sum_{k=1}^N S_{ik} = 1 \quad \forall i$.

Definition 3. Vertex j is the connected neighbor of vertex i , if there is an edge between i and j , and either $e_{ij} = 1$ or $e_{ji} = 1$.

Next we impose the constraint that vertex i is in partition k if and only if its connected neighbor is also in partition k . This is expressed as follows:

$$S_{ik} = 1 \text{ if and only if } \sum_{j=1}^n e_{ij} S_{jk} = 1 \Rightarrow S_{ik} = \sum_{j=1}^n e_{ij} S_{jk}.$$

The existence of Hamiltonian paths in non-overlapping partitions ensures that, for every cell i in array, $\sum_{j=1}^n e_{ij} = \sum_{j=1}^n e_{ji} = 1$.

Finally, we incorporate the objective function into the ILP model. The objective of this optimization problem is to minimize the total cost $C = \max_k \{C_k\} = \max_k \{n_k\}$, $k = 1, 2, \dots, N$, where n_k is the number of vertices visited by Hamiltonian path k . It is easily seen that $n_k = \sum_{i=1}^n S_{ik}$. Therefore, $C = \max_{1 \leq k \leq N} \sum_{i=1}^n S_{ik}$.

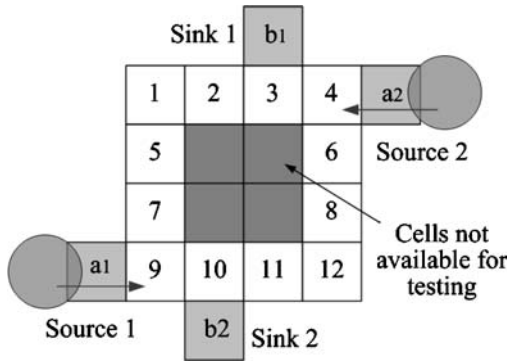


Fig. 6. An example of 4 × 4 microfluidic array.

We now have a mathematical programming model for OPP, described as follows.

Objective: Minimize $C = \max_{1 \leq k \leq N} \sum_{i=1}^n S_{ik}$

- Subject to:
1. $\sum_{k=1}^N S_{ik} = 1 \quad \forall i.$
 2. $S_{ik} = \sum_{j=1}^n e_{ij} S_{jk} \quad \forall i, 1 \leq k \leq N.$
 3. $\sum_{j=1}^n e_{ij} = \sum_{j=1}^n e_{ji} = 1 \quad \forall i.$

In order to solve the above mathematical programming model using *lpsolve*, its objective function and some constraint inequalities must be linearized to match the canonical form of an ILP model. First, the objective function is linearized as: Minimize C , subject to $C \geq \sum_{i=1}^n S_{ik}, 1 \leq k \leq N$. The set of constraints in (2) above contains the non-linear term $e_{ij}S_{jk}$, which can be linearized by introducing a binary variable $Z_{ijk} = e_{ij}S_{jk}$, with the following additional constraints [25]:

1. $e_{ij} + S_{jk} - Z_{ijk} \leq 1.$
2. $e_{ij} + S_{jk} - 2Z_{ijk} \geq 0.$

This transformation is verified as follows: If $S_{jk} = 0$, from (1) and (2), $Z_{ijk} + 1 \geq e_{ij}$ and $2Z_{ijk} \leq e_{ij}$; since $e_{ij} \leq 1$, $Z_{ijk} = 0$. If $S_{jk} = 1$, we get $Z_{ijk} \geq e_{ij}$ and $2Z_{ijk} \leq e_{ij} + 1$. Therefore, $Z_{ijk} = e_{ij}$.

We now describe the ILP model for OPP, which includes the new variable and constraints.

Objective: Minimize C
 Subject to: 1. $C \geq \sum_{i=1}^n S_{ik} \quad 1 \leq k \leq N.$

2. $\sum_{k=1}^N S_{ik} = 1 \quad \forall i.$
3. $S_{ik} = \sum_{j=1}^n Z_{ijk} \quad \forall i, 1 \leq k \leq N.$
4. $e_{ij} + S_{jk} - Z_{ijk} \leq 1 \quad \forall i, \forall j, 1 \leq k \leq N.$
5. $e_{ij} + S_{jk} - 2Z_{ijk} \geq 0 \quad \forall i, \forall j, 1 \leq k \leq N.$
6. $\sum_{j=1}^n e_{ij} = \sum_{j=1}^n e_{ji} = 1 \quad \forall i.$

The above ILP model can now be solved using *lpsolve*. The complexity of this model, measured by the number of variables and the number of constraints, is $O(n^2 \times N)$, where n is the number of cells in an array and N is the number of source/sink pairs.

The following example illustrates an optimal partitioning of a 4 × 4 microfluidic array with two sources a1 and a2, and two sinks b1 and b2, respectively; see Fig. 6. The result is obtained using *lpsolve*. It took 10 minutes of CPU time on a 1.6 GHz Pentium-IV PC with 392 MB of RAM. An optimal partitioning generated by *lpsolve* is as follows: Partition 1 = {1, 2, 3, 5, 7, 9, a1, b1}, Partition 2 = {4, 6, 8, 10, 11, 12, a2, b2}; see Fig. 7. Based on this test plan, the total time cost C is $\max\{8, 8\} = 8$.

We have shown that an ILP model can be used to solve this optimization problem exactly for a microfluidic array of modest size. However, there exist several major limitations inherent in OPP:

1. Sometimes there exists no Hamiltonian path in the array. Even if Hamiltonian paths exist, optimal partitioning

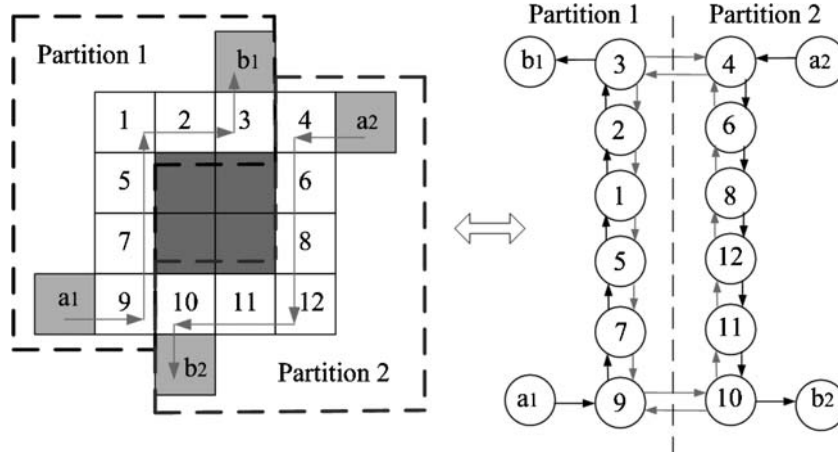


Fig. 7. An optimal partition and droplet flow path for the 4 × 4 microfluidic array of Fig. 6.

obtained by solving OPP may not be the best solution for optimal test planning. The suboptimal nature of the test plan derived from the optimal solution to OPP results from the property of a Hamiltonian path that every node in the path should be visited exactly once. Lower-cost solutions can be obtained if we allow a cell to be visited more than once.

2. The partitioning in OPP does not take into account the constraint that droplets can never be in a cell directly adjacent or diagonally adjacent to another droplet. The optimal solution to OPP may be not a feasible test plan because perhaps some test stimuli droplets would be adjacent to each other, thereby leading to the mixing of these droplets.

Moreover, due to the inherent complexity of the model, there is a need for heuristic algorithms that can be applied to a large array and that can eliminate the limitations inherent in the OPP problem.

7. Heuristic Algorithms

One possible heuristic method is motivated by the similarity of the test planning problem for a microfluidic array to the robot motion planning problem, where we view every test stimuli droplet as a mobile robot. However, there are a number of important differences:

1. The test planning problem can be considered as a combination of both the navigation problem and the full coverage problem. It attempts to minimize the total time cost from the starting point (droplet source) to the end point (droplet sink), while it also requires all available cells to be covered in the droplet path. Therefore it is more complicated than either the navigation problem or the coverage problem alone.
2. A major constraint in the application of multiple test stimuli droplets is that droplets can never be in a cell directly adjacent or diagonally adjacent to another droplet except in the case of mixing of two droplets. This restriction increases the complexity of the problem of test planning and resource optimization.

7.1. Simple Monte-Carlo Search Algorithm (SMC)

Monte-Carlo based search algorithms have been proposed in the literature for problems with a large number of constraints [16]. The key idea underlying these algorithms is that random points are generated in the search space and the point with the lowest value for the objective function is taken to be the global optimum. In this modified random walk method, a large number of simulation runs are carried out to generate enough samples. First we apply the simple Monte-Carlo search algorithm to heuristically solve the problem of test planning and optimization. In each run, the test stimuli droplet starts from the cell directly adjacent

to the droplet source and ends in the droplet sink. It randomly moves to the neighboring cell with some probability p . We mark the cell if it has been visited, then the larger p is assigned to the motion towards the unmarked cell. After randomly selecting the new positions of test stimuli droplets, the procedure checks if no two droplets are directly adjacent or diagonally adjacent in their new positions. If this restriction is satisfied, test stimuli droplets move to these new positions. Otherwise the new positions are selected again. If all available cells have been visited and test stimuli droplets have reached the droplet sinks, the test process is concluded. Here we assume that each droplet move only once in each time slot. Therefore, the test plan with the smallest number of total time slots, i.e., total test time, is selected as the optimal solution.

7.2. Modified Real-Time Algorithm (MRT)

We can further leverage real-time search algorithms and incorporate them into the heuristic algorithm for test planning. While the previous Monte-Carlo search algorithm simply marks the cell with a binary variable (0/1) based on whether it has been visited, this modified algorithm associates an evaluation function U with each cell. It always decides which neighboring cell to move to based only on the U -values of its neighbors. That is, the droplet always greedily moves to an adjacent cell with the smallest U -value. Ties due to same U -value neighbors are broken randomly. Similar to the Monte-Carlo search algorithm, the new positions of test stimuli droplets should be verified to satisfy the physical restriction. Then the U -value of the current cell is updated according to a predefined rule. We study four different U -value update rules, which been used successfully in robot motion planning, as listed in Table 1. Each rule assigns a different meaning to the U -value. For example, Node Counting interprets the U -value as the number of times the location has been visited, while LRTA* interprets U -value as approximations of the goal distances of the location [12, 21]. The introduction of the evaluation function U decreases the arbitrariness of the selection of new positions in the Monte-Carlo search algorithm and therefore increases the possibility of finding a better solution for the same number of simulation runs.

7.3. Proposed Improved Heuristic Algorithm for Multiple Droplets (PIH-MD)

When multiple test droplets are used, the above heuristic algorithms might move two droplets closer to each other. Additional effort may therefore be needed to prevent droplets from being directly or diagonally adjacent to each other. Moreover, if these two droplets are too close, the overlap of their coverage areas might increase, consequently leading to low efficiency in searching. Therefore we modify the heuristic algorithm for multiple test

Table 1. Different U -value update rules.

Value-update rules	Real-time search algorithms
$U(\text{current}) = 1 + U(\text{current})$	Node counting [1]
$U(\text{current}) = 1 + U(\text{New})$	Learning Real-time A* (LRTA*) [12]
If $U(\text{current}) \leq U(\text{New})$, $U(\text{current}) = 1 + U(\text{current})$	Wagner's value-update rule [24]
$U(\text{current}) = \max(1 + U(\text{current}), 1 + U(\text{New}))$	Thrun's value-update rule [23]

droplets by attempting to separate two droplets. We add a new evaluation function ΔP to approximate the relative distance between two droplets. When ties for new positions with the lowest U -value are encountered, we evaluate the ΔP function for every two possible positions of these droplets and select the new positions with smallest value of ΔP . Instead of breaking such ties arbitrarily as

in MRT, this approach adds more guidance to heuristically find the near-optimal solution for test planning. Simulation results presented in the next section show that it provides better performance than the simple Monte-Carlo search algorithm and the modified real-time search algorithm for multiple test stimuli droplets. The procedure is outlined in Fig. 8.

```

Loop: For  $n = 1$  to  $N$  (the maximum number of simulation runs)
  Initialization: Status initialization:
    All cells available for testing are set to '0';
    All cells not available for testing are set to '2'.
    (Here '0' denotes that the cell is not visited yet.
     '1' denotes that the cell has been visited
     '2' denotes that the cell is not available for testing)
  Evaluation function value initialization:
    The  $U$  values of all cells are set to 0.
  Starting point:
    The cell adjacent to source is set to be '1' when  $t = 1$ 
  Loop: For  $t = 2$  to  $T$  (maximum index of time-slot)
    1. Select new location of test stimuli droplet:
      Droplet moves to its neighbor cell with smallest  $U$ -value. That is,
       $U(\text{new location}) = \min(U(\text{neighbors of current location}))$ . When
      there are ties, we evaluate  $\Delta P$  between two droplets.
    2. Verify relative distance between new locations:
      We select the new locations which satisfy the restriction and
      have lowest  $\Delta P$ .
    3. Update  $U$ -value of current location, then go on to next time-slot.
      If all available cells have been visited and test stimuli droplets have
      reached the sink, (Test finished)
      Record the time cost;
      Record test planning;
      Break; End;
  End
  If time cost < minimum cost
    minimum cost = time cost.
    Record the best test planning.
  End

```

Fig. 8. Sketch of the improved heuristic algorithm.

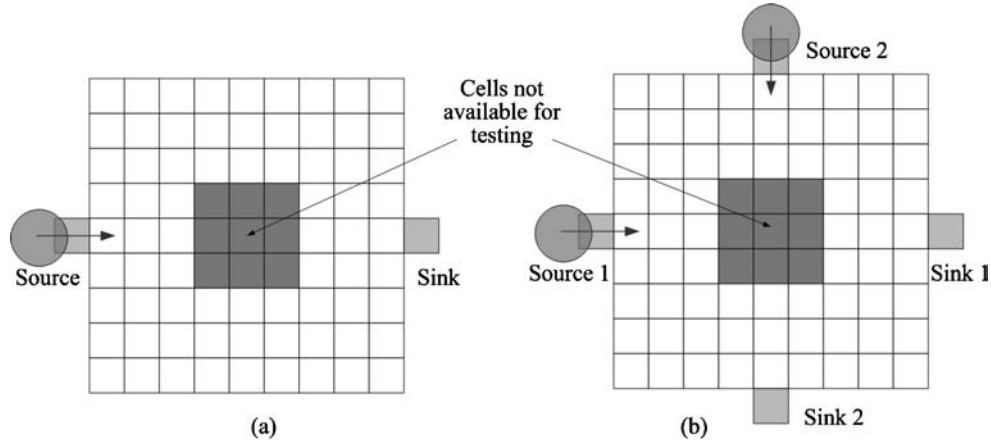


Fig. 9. (a) Microfluidic array configuration in the first set of experiments. (b) Microfluidic array configuration in the second set of experiments.

8. Experimental Results

In this section, we report simulation results on test planning and resource optimization for droplet-based two-dimensional microfluidic arrays. Note that there exists an inherent tradeoff between test hardware overhead, i.e., test stimuli droplet source/sink pairs, and test application time. The location and the number of droplet sources and sinks can affect the time cost of the associated test plan [19]; a higher test hardware overhead usually leads to less testing time. Here we attempt to minimize the test application time for a given test hardware overhead. In addition, the concurrent test plan for a microfluidic array is also affected by the array configuration, i.e., the usage of cells in normal biomedical assays. In fact, design-for-testability (DFT) techniques can be used to optimize the assay schedule and array configuration to increase the efficiency of the corresponding concurrent test plan. In order to facilitate the evaluation and comparison, a set of given array configurations are deployed here for the different proposed test planning methods.

In the following experiments, two sets of cases are analyzed:

1. A single source and a single sink;
2. Two sources and two sinks.

The configurations of the microfluidic arrays, e.g., the assignment of cells used for biomedical assays, as well as the locations of the source and the sink used in both sets of experiments, are shown in Fig. 9.

For arrays of modest size, optimal solutions can be obtained using ILP model. Therefore, we can compare the result of the heuristic algorithms with the optimal solution (OPT). However, for arrays of the larger size, optimal solutions are not available. The performance of heuristic algorithms in these cases can only be compared with a lower bound (LB) and an upper bound (UB) on the optimal solution as described next.

In an ideal case, the available cells of the array can be partitioned evenly. In each partition, there exists a Hamiltonian path from one droplet source to one droplet sink. Multiple tests can be run in non-overlapping parts in parallel without violating the restriction on droplet motion. Therefore, we have a lower bound LB on optimal solution of $[n/k]$, where n is the number of available cells in the system and k is the number of source-sink pairs. The tightness of this lower bound is determined by the topological configuration of the microfluidic array. In addition, an upper bound on the optimal solution can be shown to be $2 \times n$, which results from the depth-first search on a grid graph [6].

Table 2. Simulation results for Case (1). The entries in the table denote testing time (in time-slots).

	3×3	3×5	4×4	5×5	6×6	7×7	8×8	9×9
OPT	8	12	14	23	N/A	N/A	N/A	N/A
LB	8	10	14	22	30	41	52	64
UB	16	20	28	44	60	82	104	128
SMC	8	12	14	30	39	54	84	91
NC	8	12	14	23	34	47	66	77
LRTA*	8	12	14	25	34	47	66	81
Wagner	8	12	14	25	34	49	70	78
Thrun	8	12	14	23	32	47	62	77

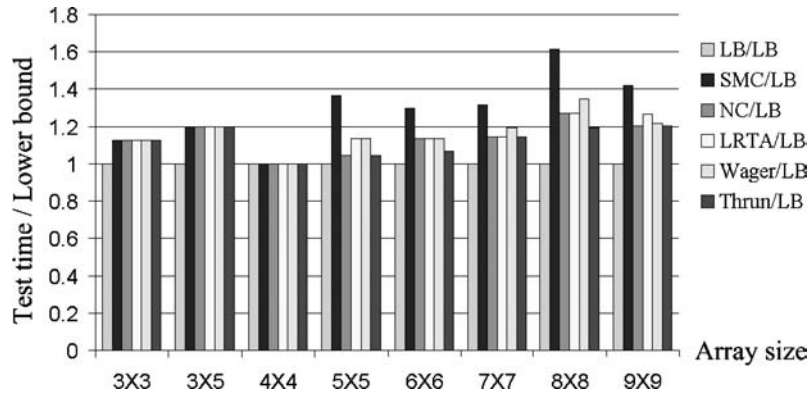


Fig. 10. Comparison of various heuristic approaches.

In the first set of experiments, we determined the test time for two different heuristic algorithms, i.e., the simple Monte-Carlo algorithm and the modified real-time algorithm (four different U -value update rules) for Case (1). We assigned 10,000 runs to the simple Monte-Carlo algorithm and 1,000 runs to the modified real-time algorithm. Table 2 shows the simulation results. Some optimal solutions obtained from the ILP model, as well as lower bounds and upper bounds, are also listed. The results show that heuristic algorithms provide close-to-optimal solutions for small array sizes, such as 3×3 , 4×4 and 5×5 . When the array size increases, the results for heuristic algorithms are still contained between the lower bound and upper bound

of the optimal solution. The results for modified real-time algorithm are much closer to the lower bound than the simple Monte-Carlo algorithm; see Fig. 10. These experimental results highlight the advantage of adding the evaluation function U -value.

In the second set of experiments for multiple test stimuli droplets (Case 2), we compare the modified real-time algorithms (MRT) to the proposed improved heuristic algorithm (PIH-MD). Here the arrays of larger sizes are considered. Simulation result shows that the improved heuristic algorithm significantly outperforms the modified real-time algorithm for larger array sizes; see Fig. 11. The ratio of the actual testing time to the lower bound is always under 1.8

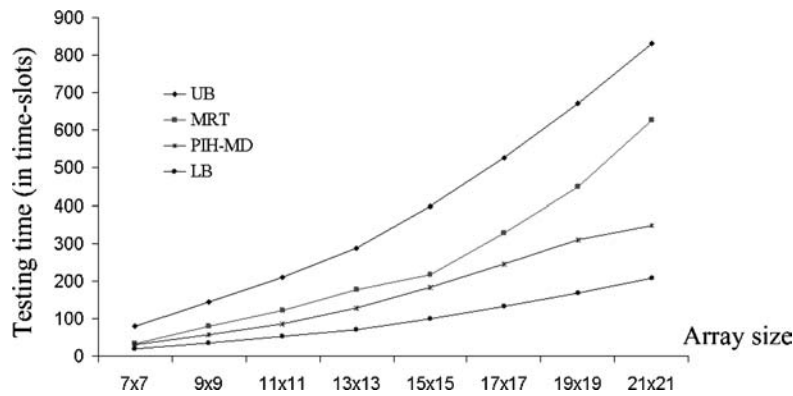


Fig. 11. Simulation result for Case(2).

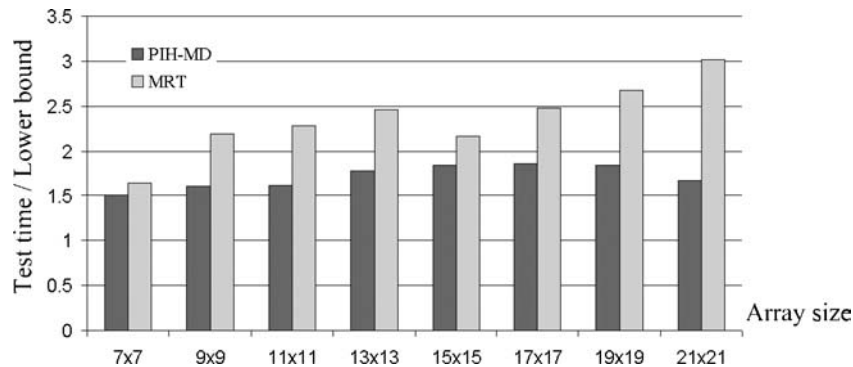


Fig. 12. Scalability of PIH-MD compared to MRT.

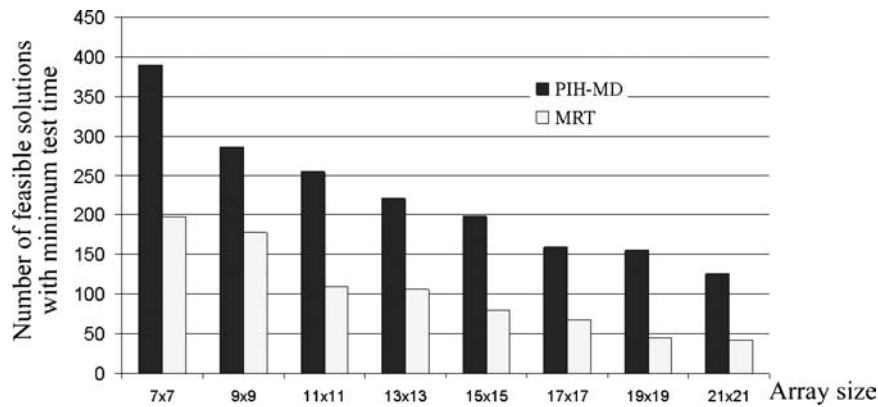


Fig. 13. Comparison of the number of available solutions for 500 simulation runs.

for the improved heuristic algorithm, while this ratio for the modified real-time algorithm increases with the array size; see Fig. 12.

Finally, we study the number of available solutions for each heuristic algorithm when the number of simulation runs is fixed, i.e., 500. Figure 13 shows that the proposed improved heuristic algorithm (PIH-MD) generates many more available solutions than the modified real-time algorithm (MRT). This advantage results from adding a new evaluation function ΔP to reduce the overlap between the coverage areas of the two test stimuli droplets, and it leads to a better solution for test planning and resource optimization.

9. Conclusions

In this paper, we have presented an analysis of the test planning problem for droplet-based microfluidic systems. Due to NP-hard nature of the problem, heuristic approaches are needed. We have developed heuristic algorithms that are applicable to droplet-based microfluidic arrays of large sizes. Experiment results have shown that the heuristic solutions are close to the lower bounds on the optimal solutions. The advantage of the improved heuristic algorithm for multiple test stimuli droplets has been evaluated. In our ongoing work, we are investigating fault tolerance techniques based on the concurrent testing and reconfigurability of the droplet-based microfluidic systems.

References

1. T. Balch and R. Arkin, "Avoiding the Past: A Simple, But Effective Strategy for Reactive Navigation," in *Proc. Int. Conf. on Robotics and Automation*, 1993, pp. 678–685.
2. M. Berkelaar, Ipsolve. Eindhoven Univ. Technol., Eindhoven, The Netherlands. [Online]. Available: ftp://ftp.ics.ele.tue.nl/pub/lp_solve.
3. K.F. Böhringer, "Optimal strategies for moving droplets in digital microfluidic systems," in *Int. Conf. on Miniaturized Chemical and Biochemical Analysis Systems (MicroTAS'03)*, 2003, pp. 591–594.
4. N. Deb and R.D. Blanton, "Analysis of Failure Sources in Surface-Micromachined MEMS", in *Proc. IEEE Int. Test Conf.*, 2000, pp. 739–749.
5. J. Gross and J. Yellen, *Graph Theory and its Applications*, Boca Raton, FL: CRC Press, 1999.
6. C. Icking, T. Kamphans, R. Klein and E. Langetepe, "Exploring an Unknown Cellular Environment", in *Proc. European Workshop on Computational Geometry*, 2000, pp. 140–143.
7. A. Itai, C.H. Papadimitriou and J.L. Szwarcfiter, "Hamilton Paths in Grid Graphs," *SIAM Journal on Computing*, vol. 11, pp. 676–686, 1982.
8. H.G. Kerkhoff, "Testing Philosophy Behind the Micro Analysis System," in *Proc. SPIE: Design, Test and Microfabrication of MEMS and MOEMS*, vol. 3680, 1999, pp. 78–83.
9. H.G. Kerkhoff and M. Acar, "Testable Design and Testing of Micro-Electro-Fluidic Arrays," in *Proc. IEEE VLSI Test Symposium*, 2003, pp. 403–409.
10. H.G. Kerkhoff and H.P.A. Hendriks, "Fault Modeling and Fault Simulation in Mixed Micro-Fluidic Microelectronic Systems," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 17, pp. 427–437, 2001.
11. A. Kolpekwar and R.D. Blanton, "Development of a MEMS testing methodology," in *Proc. IEEE Int. Test Conf.*, 1997, pp. 923–931.
12. R.E. Korf, "Real-Time Heuristic Search," *Artificial Intelligence*, vol. 42, pp.189–211, 1990.
13. W. Menz and A. Guber, "Microstructure Technologies and Their Potential in Medical Applications," *Minimally Invasive Neurosurgery*, vol. 37, pp. 21–27, 1994.
14. S. Mir, B. Charlot and B. Courtois, "Extending Fault-Based Testing to Microelectromechanical Systems," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 16, pp. 279–288, 2000.
15. S. Mir, H. Kerkhoff, R.D. Blanton, H. Bederr, and H. Klim, "SoCs with MEMS? Can we include MEMS in the SoCs design and test flow?," in *Proc. IEEE VLSI Test Symposium*, pp. 449, 2002.
16. S.D. Nigam and J.U. Turner, "Review of statistical approaches to tolerance analysis", in *Computer-Aided Design*, vol. 27, pp. 6–25, 1995.
17. C.H. Papadimitriou, *Computational Complexity*, Reading, MA: Addison Wesley, 1993.
18. M. G. Pollack, R. B. Fair, and A. D. Shenderov, "Electrowetting-based actuation of liquid droplets for microfluidic applications", *Applied Physics Letters*, vol. 77, pp. 1725–1726, 2000.
19. F. Su, W. Hwang, A. Mukherjee, and K. Chakrabarty, "Defect-Oriented Testing and Diagnosis of Digital Microfluidics-Based Biochip", *Proc. IEEE Int. Test Conf.*, 2005.
20. F. Su and K. Chakrabarty, "Design of Fault-Tolerant and Dynamically-Reconfigurable Microfluidic Biochips", in *Proc. Design, Automation and Test in Europe (DATE) Conference*, pp. 1202–1207, 2005.

21. F. Su and K. Chakrabarty, "Architectural-level Synthesis of Digital Microfluidics-Based Biochips", in *Proc. IEEE International Conference on CAD*, 2004, pp. 223–228.
22. F. Su, S. Ozev and K. Chakrabarty, "Testing of Droplet-Based Microelectrofluidic Systems", in *Proc. IEEE Int. Test Conf.*, 2003, pp. 1192–1200.
23. S. Thrun, *Efficient Exploration in Reinforcement Learning*. Technical Report CMU-CS-92-102, Carnegie Mellon University, 1992.
24. I. Wagner, M. Lindenbaum, and A. Bruckstein, "On-line Graph Searching by A Smell-Oriented Vertex Process", in *Proc. AAAI Workshop on On-Line Search*, 1997, pp. 122–125.
25. H. P. Williams, *Model Building in Mathematical Programming*, John Wiley & Sons Ltd., 1999.
26. T. Zhang, K. Chakrabarty, and R. B. Fair, *Microelectrofluidic Systems: Modeling and Simulation*, Boca Raton, FL: CRC Press, 2002.
27. International Technology Roadmap for Semiconductor (ITRS), <http://public.itrs.net/Files/2003ITRS/Home2003.htm>.

Fei Su received the B.E. and the M.S. degrees in automation from Tsinghua University, Beijing, China, in 1999 and 2001, respectively, and the M.S. degree in electrical and computer engineering from Duke University, Durham, NC, in 2003. He is now a Ph.D. candidate in electrical and computer engineering at Duke University. His research interests include design and testing of mixed-technology microsystems, electronic design automation, mixed-signal VLSI design, MEMS modeling and simulation.

Sule Ozev received her B.S. degree in Electrical Engineering at Bogazici University in 1995, and her M.S. and Ph.D. degrees in Computer Science and Engineering at University of California, San Diego in 1998 and 2002 respectively. Since 2002, she has been a faculty member at Duke University, Electrical and Computer Engineering Department. Her research interests include RF circuit analysis and testing, process variability analysis, and mixed-signal testing.

Krishnendu Chakrabarty received the B. Tech. degree from the Indian Institute of Technology, Kharagpur, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively, all in Computer Science and Engineering. He is now Associate Professor of Electrical and Computer Engineering at Duke University. Dr Chakrabarty is a recipient of the National Science Foundation Early Faculty (CAREER) award and the Office of Naval Research Young Investigator award. His current research projects include: design and testing of system-on-chip integrated circuits; design automation of microfluidics-based biochips; microfluidics-based chip cooling; distributed sensor networks. Dr Chakrabarty has authored three books *Microelectrofluidic Systems: Modeling and Simulation* (CRC Press, 2002), *Test Resource Partitioning for System-on-a-Chip* (Kluwer, 2002), and *Scalable Infrastructure for Distributed Sensor Networks* (Springer, 2005) 3/4 and edited the book volume *SOC (System-on-a-Chip) Testing for Plug and Play Test Automation* (Kluwer 2002). He has published over 200 papers in journals and refereed conference proceedings, and he holds a US patent in built-in self-test. He is a recipient of best paper awards at the 2005 IEEE International Conference on Computer Design and 2001 IEEE Design, Automation and Test in Europe (DATE) Conference. He is also a recipient of the Humboldt Research Fellowship, awarded by the Alexander von Humboldt Foundation, Germany.

Dr Chakrabarty is an Associate Editor of *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on VLSI Systems*, *IEEE Transactions on Circuits and System I*, *ACM Journal on Emerging Technologies in Computing Systems*, and an Editor of *Journal of Electronic Testing: Theory and Applications (JETTA)*. He is a member of the editorial board for *Sensor Letters* and *Journal of Embedded Computing* and he serves as a subject area editor for the *International Journal of Distributed Sensor Networks*. He has also served as an Associate Editor of *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*. He is a senior member of IEEE, a member of ACM and ACM SIGDA, and a member of Sigma Xi. He serves as Vice Chair of Technical Activities in IEEE's Test Technology Technical Council, and is a member of the program committees of several IEEE/ACM conferences and workshops. He served as the Program Co-Chair for the 2005 IEEE Asian Test Symposium.