# 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
E-mail: {fs, sule, krish}@ee.duke.edu

## 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 methods for droplet-based microfluidic arrays. We first outline a methodology based on integer linear programming (ILP) that yields optimal solutions. Due to the NP-complete nature of the problem, we develop heuristic approaches for optimization. Experimental results indicate that for large array sizes, heuristic methods yield solutions that are close to provable lower bounds. These heuristics ensure scalability and low computation cost.*

## 1 Introduction

Next-generation system-on-chip designs are expected to be composite microsystems with microelectromechanical and microfluidic components [1, 2]. 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 [3].

In recent years, novel droplet-based microfluidic systems have been developed to analyze nanoliter volumes of agents [4]. 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 techniques 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 the reliability, composite microsystems incorporating microfluidic components must be tested adequately. Therefore, there is a pressing need for efficient testing methodologies for these microsystems. The ITRS 2001 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 2007 [5].

Recently, a fault classification and a unified test methodology for droplet-based microfluidic systems has been developed [6]. Faults are classified into catastrophic and parametric categories, and techniques are developed to detect these faults by electrostatically controlling and tracking droplet motion. This cost-effective test methodology facilitates concurrent testing, which allows 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 outline an optimal solution based on integer linear programming (ILP). Due to the NP-complete 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 a droplet-based microfluidic system. Related prior work is discussed in Section 3. Section 4 describes the problem of test planning and test resource optimization. An optimal solution based on integer linear programming is outlined and the problem is shown to be NP-complete. Section 5 presents several heuristic algorithms, which are evaluated through simulation experiments in Section 6. Finally, conclusions are drawn in Section 7.

*This research was supported in part by the National Science Foundation under grant number EIA-0312352.

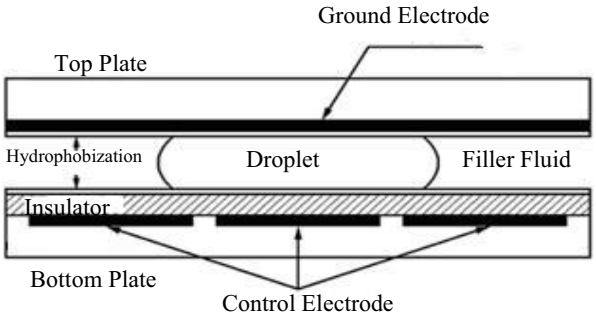


**Figure 1:** Basic components of a droplet-based microfluidic system.

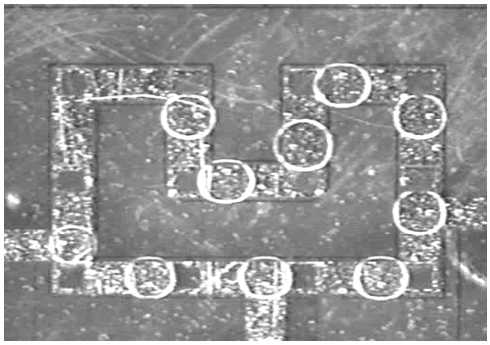


**Figure 2:** Droplet transport in a two-dimensional array.
(detailed video available at http://www.ee.duke.edu/Research/microfluidics)

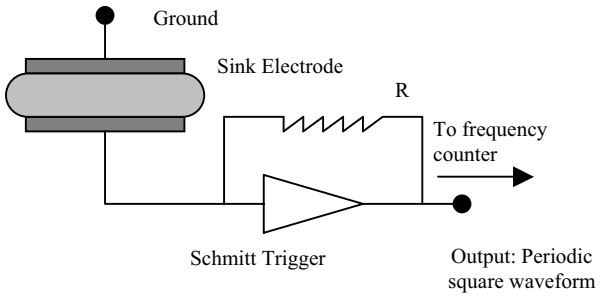


**Figure 3:** Simple capacitive sensing circuit.

## 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 [4]. 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 Figure 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.

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 a surface 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 Figure 2. This design, which has been fabricated on PCBs at Duke University [4], 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 Figure 3.

## 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 has received attention [7, 8, 9]. Researchers in 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 [10, 11, 12]. 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 [12]. Similar to the MOSFET, a Flow-FET has source and drain electrodes over which a relatively large voltage (~100V) 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 [13]. 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 [14, 15]. 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 [6], 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. For the faulty case, the test stimuli droplet is stuck at an intermediate point during motion. On the other hand, the detection of all test stimuli droplets at the droplet sink 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.

We can formulate the test planning problem in terms of graph partitioning and the Hamiltonian path problem from graph theory. 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. The test plan obtained from this method allows multiple tests to run in different non-overlapping parts of the microfluidic array in parallel, which results in the reduction of test application time. Figure 4 shows an example of a test plan, where the white cells are available for testing, and the black cells are in use by a biomedical assay and therefore temporarily unavailable for testing.

This optimization problem can be proven to be NP-complete. The analysis of computational complexity is based on the reduction from the problem of determining a Hamiltonian cycle in grid graphs, which is known to be NP-complete [16]. Details of this proof can be found

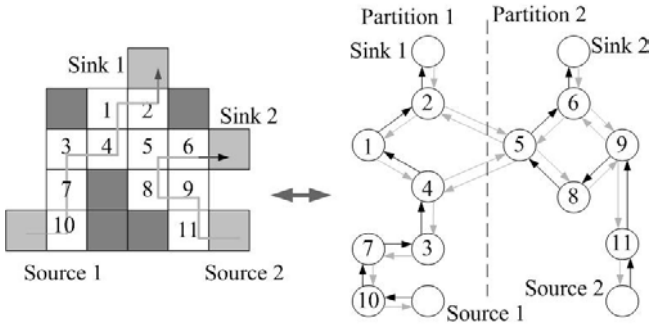


**Figure 4:** Optimal partitioning for a 4x4 array.

in [17]. An integer linear programming (ILP) model can be formulated to solve this optimization problem exactly for a microfluidic array of modest size. However, due to the inherent complexity of the model, there is a need for heuristic algorithms that can be applied to a large array.

# 5 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.

### 5.1 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 [18]. 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.

### 5.2 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 [15, 19]. 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.

| Value-Update Rules | Real-time search algorithms |
|---|---|
| $U$(current) = 1 + $U$(current) | Node counting [14] |
| $U$(current) = 1 + $U$(New) | Learning Real-time A* (LRTA*) [15] |
| If $U$(current)≤ $U$(New), $U$(current) = 1 + $U$(current) | Wagner's value-update rule [19] |
| $U$(current) = max(1 + $U$(current), 1 + $U$(New)) | Thrun's value-update rule [20] |

**Table 1:** Different $U$-value update rules.

***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(new location)=min(U(neighbors of current location). When there are ties, we evaluate ΔP between two droplets.*
*2.* ***Verify relative distance between new locations:***
*We select the new locations which satisfy the restriction and have lowest ΔP.*
*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***

**Figure 5:** Sketch of the improved heuristic algorithm.

### 5.3 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 droplets by attempting to separate two droplets. We add a new evaluation function $\Delta P$ to approximate the relative distance between two droplets. When ties for new positions with the lowest $U$-value are encountered, we evaluate the $\Delta P$ function for every two possible positions of these droplets and select the new positions with smallest value of $\Delta P$. 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 Figure 5.

## 6. Experimental Results

In this section, we report experimental simulation results on test planning and resource optimization for droplet-based two-dimensional microfluidic arrays. We attempt to minimize the test application time for a given test hardware overhead, i.e., the number of droplet source/sink pairs. In the following experiments, two sets of cases are analyzed:

(1) One single source and one single sink;

(2) Two sources and two sinks.

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 [21].

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

| | 3x3 | 3x5 | 4x4 | 5x5 | 6x6 | 7x7 | 8x8 | 9x9 |
|---|---|---|---|---|---|---|---|---|
| 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 |

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

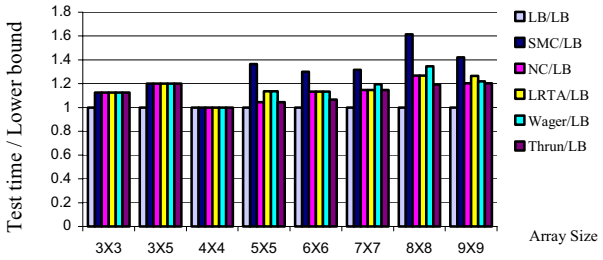


**Figure 6:** Comparison of various heuristic approaches.

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 Figure 6. 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 Figure 7. The ratio of the actual testing time to the lower bound is always under 1.8 for the improved heuristic algorithm, while this ratio for the modified real-time algorithm increases with the array size; see Figure 8.

Finally, we study the number of available solutions for each heuristic algorithm when the number of simulation runs is fixed, i.e., 500. Figure 9 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 $\Delta P$ 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.

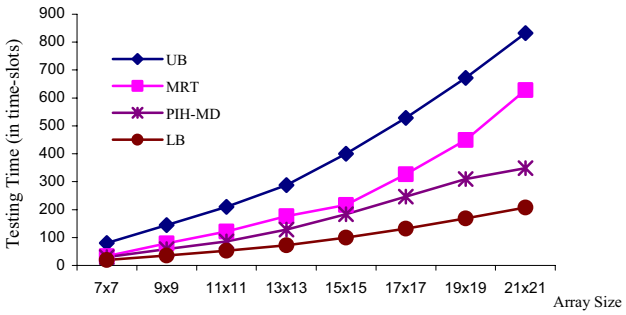


**Figure 6:** Simulation results for Case (2).

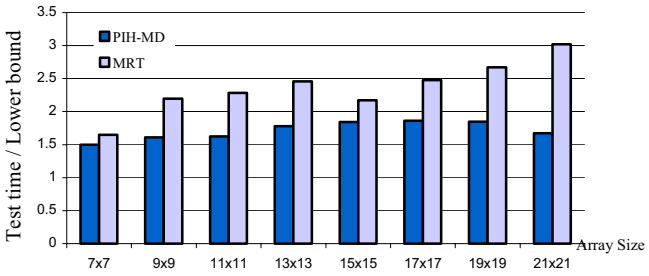


**Figure 7:** Scalability of PIH-MD compared to MRT.

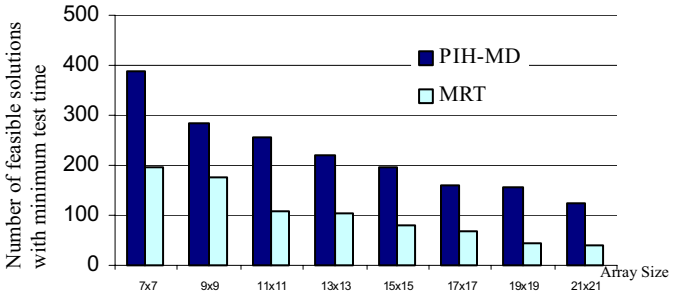


**Figure 8:** Comparison of the number of available solutions for 500 simulation runs.

# 7 Conclusions

In this paper, we have presented an analysis of the test planning and test resource optimization problems for droplet-based microfluidic systems. Due to NP-complete 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] S. Mir, H. Kerkhoff, R.D. Blanton, H. Bederr, H. Klim, "SoCs with MEMS? Can we include MEMS in the SoCs design and test flow?", *Proc. IEEE VLSI Test Symposium*, p. 449, 2002.

[2] T. Zhang, K. Chakrabarty and R. B. Fair, *Microelectrofluidic Systems: Modeling and Simulation*, CRC Press, Boca Raton, FL, 2002.

[3] W. Menz and A. Guber, "Microstructure technologies and their potential in medical applications", *Minimally Invasive Neurosurgury*, vol. 37, pp.21-27,1994.

[4] M. G. Pollack, R. B. Fair and A. D. Shenderov, "Electrowetting-based actuation of liquid droplets for microfluidic applications", *Applied Physics Letters*, vol. 77, 2000.

[5] International Technology Roadmap for Semiconductors (ITRS), http://public.itrs.net/Files/2001ITRS/Test.pdf.

[6] F. Su, S. Ozev and K. Chakrabarty, "Testing of droplet-based microelectrofluidic systems", *Proc. IEEE Int. Test Conf.*, pp. 1192-1200, 2003.

[7] A. Kolpekwar and R. D. Blanton, "Development of a MEMS testing methodology", *Proc. Int. Test Conf.*, pp. 923-93, 1997.

[8] N. Deb and R. D. Blanton, "Analysis of failure sources in surface-micromachined MEMS", *Proc. Int. Test Conf.*, pp. 739-749, 2000.

[9] S. Mir, B. Charlot and B. Courtois. "Extending fault-based testing to microelectromechanical systems." *Journal of Electronic Testing: Theory and Applications,* vol. 16, pp. 279-288, 2000.

[10] H. G. Kerkhoff, "Testing philosophy behind the micro analysis system", *Proc. SPIE: Design, Test and Microfabrication of MEMS and MOEMS,* vol. 3680, pp.78-83, 1999.

[11] 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*, vol. 17, pp. 427-437, 2001.

[12] H. G. Kerkhoff and M. Acar, "Testable design and testing of micro-electro-fluidic arrays", *Proc. IEEE VLSI Test Symposium*, pp. 403-409, 2003.

[13] K. F. Böhringer, "Optimal strategies for moving droplets in digital microfluidic systems." *Int. Conf. on Miniaturized Chemical and Biochemical Analysis Systems (MicroTAS'03),* pp. 591-594, 2003.

[14] T. Balch and R. Arkin. "Avoiding the past:A simple, but effective strategy for reactive navigation." *Proc. Int. Conf. on Robotics and Automation* , pp. 678 -685, 1993.

[15] R. E. Korf, "Real-time heuristic search", *Artificial Intelligence*, v.42 n.2-3, p.189-211, March 1990.

[16] A. Itai, C. H. Papadimitriou and J. L. Szwarcfiter, "Hamilton paths in grid graphs", *SIAM Journal on Computing*, vol. 11, pp. 676-686, 1982.

[17] F. Su, "*Testing for Catastrophic and Parametric Faults in Droplet-Based Microfluidic Systems",* M.S. thesis, Duke University. 2003. Available online at http://www.ee.duke.edu/~fs/thesis.

[18] S. D Nigam and J. U Turner, "Review of statistical approaches to tolerance analysis", *Computer-Aided Design*, vol.27, pp. 6-25, 1995.

[19] I. Wagner, M. Lindenbaum and A. Bruckstein. On-line graph searching by a smell-oriented vertex process. *Proc. AAAI Workshop on On-Line Search* , pp. 122 - 125,1997.

[20] S. Thrun. *Efficient exploration in reinforcement learning.* Tehnical Report CMU-CS-92-102, Carnegie Mellon University, 1992.

[21] C. Icking, T. Kamphans, R. Klein and E. Langetepe, "Exploring an Unknown Cellular Environment." *Proc. European Workshop on Computational Geometry*, pp. 140-143, 2000.