Very Large Scale Droplet Microfluidic Integration (VLDMI) using Genetic Algorithm

Job S. Kasule[1], Jeevan Maddala[1,3,*], Parham Mobed[1], Raghunathan Rengaswamy[1,2,**]

[1]Department of Chemical Engineering, Texas Tech University, Box 43121, 6th Street and Canton Ave, Lubbock, TX 79409-3121

[2]Department of Chemical Engineering, Indian Institute of Technology Madras, IIT P.O., Chennai 600 036, India

[3]SysEng, LLC, Lubbock, Texas, 79453, India

Abstract

Droplet microfluidics is likely to play a central role in the development of lab-on-a-chip technologies and as a result, significant research is directed towards this field. Understanding the spatio-temporal dynamics of discrete droplets inside microfluidic devices and the design of microfluidic devices for specific tasks are some of the dominant research topics. These works have since resulted in the development of microfluidic devices with functionalities such as sorting, merging, synchronization, storing etc. However, the anticipated application of microfluidic devices to more complex problems will require more integrated devices that can incorporate the above functionalities on a single chip. In the current work, we present a genetic algorithm optimization based design tool for discovering very large scale integration of discrete microfluidic networks for a given objective function. The application of the algorithm is demonstrated through a combinatorial sequencing problem, where the objective is to achieve three different droplet combinatorial sequences for three different droplet types. Multiple fascinating, but non-obvious designs were discovered for this application. It is difficult to imagine such devices being designed using trial and error experimental procedure, which has been the main route for obtaining microfluidic device designs. With advances in technologies for fabrication of microfluidic devices, the current tool can be a significant step towards drastically cutting down on the laborious trial-and-error design process and help in developing droplet microfluidics based lab-on-a-chip platforms cheaper and faster.

*corresponding author; Tel : +1-806-853-7362
** corresponding author; Tel: +91 44 2257 4159
Email: jeevan.maddla@ttu.edu (Jeevan Maddala); raghur@iitm.ac.in (Raghunathan Rengaswamy)

1. Introduction

As a cornerstone for lab-on-a-chip technologies, the field of droplet microfluidics has become an important research field. Sustained research interest in this field is expected in the foreseeable future due to the remarkably wide range of applications for droplet microfluidics within biology and engineering fields. Some of the proposed applications include its utilization in chemical kinetics and time-controlled synthesis (Wang et al., 2006), high-throughput screening of cells (Brouzes et al., 2009), combinatorial chemistry (Song et al., 2006), fabrication of micro and nano-particles (Song, et al., 2006; Xu et al., 2005; Nie et al., 2005); integration and automation of biochemical assays (Haeberle & Zengerle, 2007), single cell analysis (Huebner et al., 2007), usage in non-electronic coding or computer functions (Fuerstman et al., 2007), etc.

In most of the aforementioned applications, microfluidic devices are often restricted to being droplet generators (Maddala et al., 2011). To extend the use of these devices to more complex biological analysis, additional functionality needs to be incorporated beyond the role of droplet generation. Such functionality may include droplet sorting (Christopher et al., 2009; Cristobal et al., 2006; Hatakayema et al., 2006; Sessoms et al., 2010), synchronization (Prakash & Gershenfeld, 2007), merging (Jin et al., 2010), and storing (Boukellal et al., 2008; Laval et al., 2009; Shim et al., 2007; Bithi & Vanapalli, 2010; Abbyad et al., 2011) among others. An attractive approach for designing such integrated microfluidic devices would be to transport confined droplets through cleverly engineered fluidic networks (Maddala & Rengaswamy, 2014; Prakash & Gershenfeld, 2007) where these tasks can be accomplished passively. This requires a good understanding of the spatiotemporal dynamics of droplets or bubbles confined inside microfluidic device networks. Whereas the dynamics of droplet behavior inside confined microfluidic devices has been relatively well studied (Maddala et al., 2013; Fuerstman, et al., 2007; Schindler & Ajdari, 2008; Cybulski & Garstecki, 2009; Sessoms et al., 2010; Sessoms et al., 2009; Smith & Gaver, 2010), the movement of droplets inside the simplest of devices can be very complex, which makes the design of very large scale integrated microfluidic devices challenging.

Conventionally, rigorous experimentation has been the main state-of-the-art approach (bottom-up) for obtaining optimal designs of such microfluidic devices. However, due to its trial-and-error nature, this approach is slow and can be expensive for designing very large scale integrated microfluidic networks. Thus, it is anticipated that advanced computational techniques

(Maddala & Rengaswamy, 2014) will become the main route for designing such devices. This approach would involve the use of rigorous simulations to obtain optimal designs for a given design objective.

Recently, the use of computational techniques to design a multi-functional microfluidic device, to passively control droplet spacing, using genetic algorithm was highlighted (Maddala & Rengaswamy, 2014). The capability of such an approach was demonstrated through three case studies: namely, the synchronization of droplets for a particular input spacing, the synchronization of droplets for a range of input spacing, and realization of constant delay between droplets exiting the ladder network. The model discovered a number of interesting ladder network orientation designs to achieve the above objectives.

In the current work, the above model has been extended to the design of very large scale droplet-based microfluidic platforms that may comprise of thousands of individual devices in a large interconnected network. To achieve this, we proposed a representation of any droplet-based microfluidic platform as a combination of nodes and branches using graph theory principles. Using genetic algorithms, optimal designs of such structures are discovered for a specified objective function. The major contributions of the work are: (i) reformulation of the microfluidic designs as a combination of nodes and edges that results in enhancing the generality of the approach, (ii) study of the coding schema and genetic operators for this representation, (iii) metrics to quantify the architecture of microfluidic devices, and (iv) the discovery of multiple interesting but non-obvious very large scale microfluidic network designs for a combinatorial sequencing problem in which three different droplet sequences were obtained from three different droplet types. To the best of our knowledge, no such study has been performed before.

In section 2 and 2.1, the model description and the simulation equations are presented. The problem statement for the combinatorial sequencing is presented in section 2.2; genetic algorithm formulation for this objective function is presented in section 2.3. Results obtained from the genetic algorithm along with the sensitivity analysis are presented in section 3. The paper ends with conclusions and future work.

2. Model description and simulation setup

Microfluidic design problems are characterized by combination of structural and continuous variables to which bounds and constraints are applied. The structural elements are related to the topology of the network, these include the number of branches (parallel, convergent or divergent) and sources and sinks through which droplets enter or leave the network. The continuous variables are the channel lengths and droplet feeding frequencies. In this work, the continuous variables are taken from literature (Maddala et al., 2011; Maddala & Rengaswamy, 2014) and the discrete variables are optimized using genetic algorithm.

The simulation tool used in the current study predicts the dynamics of droplets in any microfluidic device. A decomposition strategy (Figure 1) is used to decouple the droplet motion and the flow-pressure calculations. This strategy not only has advantages in terms of faster simulations but also in generalizing the design approach.

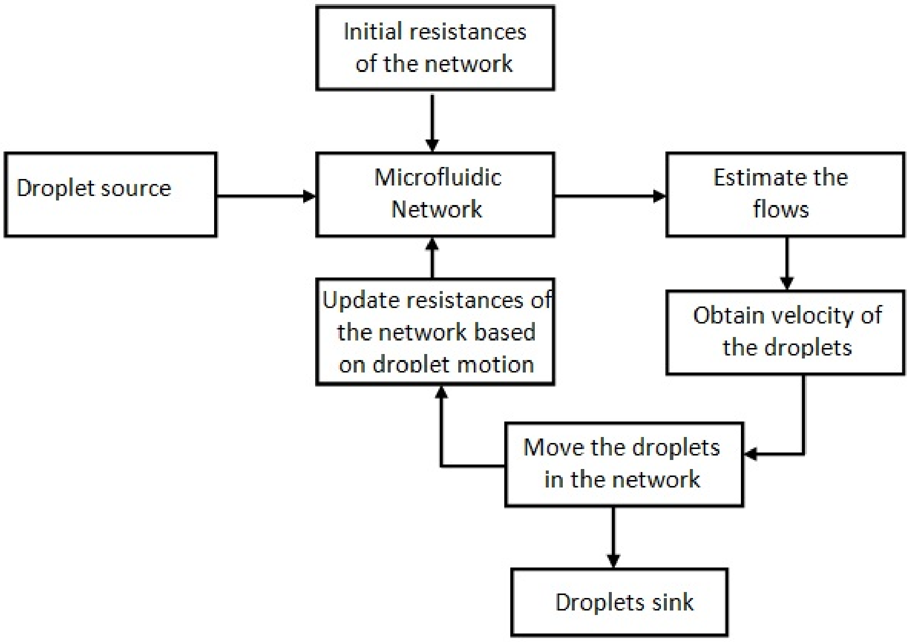

Initial resistances of the network
Droplet source
Microfluidic Network
Estimate the flows
Update resistances of the network based on droplet motion
Obtain velocity of the droplets
Move the droplets in the network
Droplets sink

Figure 1: The decomposition technique strategy for designing large scale microfluidic networks

The simulations are started by initializing the branch resistances without any droplets. The flow-pressure calculations are then performed to establish the flow of the bulk fluid in all the branches of the network, thereby allowing the calculation of all the droplet velocities. After the velocity computations, the droplets can be transported in the network until they reach various nodes and are ready to relocate to different branches. After the droplets decisions are made, the branch resistances are recalculated. At this point, the flow-pressure network calculations are performed again, and this is repeated until all the droplets exit the network or the termination criterion is met. The other aspect of the current simulation approach deals with the granularity of the network representation for the flow-pressure simulations as explained in section 2.1.

2.1 Main simulation equations

The equations used to describe the movement of droplets in the current microfluidic network systems are similar to those used in previous studies (Maddala & Rengaswamy, 2014; Schindler & Ajdari, 2008; Ajdari, 2008; Sessoms, et al., 2009). As the bulk flow is laminar, the problem involves the solution of a system of linear equations. The objective is to solve for all branch flows, nodal flows and pressures of the network. The corresponding variables are:

i. Pressures at all the nodes, P_{node}
ii. Flows in all the branches, Q_{branch}
iii. Flows into all the nodes, Q_{node}

In the first stage, the network is converted into an incidence matrix (A) whose elements are 0, 1 or -1 and of dimensions 'b' by 'n' wherein the number of rows and columns represent the number of branches and nodes in the network respectively. If there is no branch connection between two given nodes, the value of that particular element in the incidence matrix is zero. Otherwise, based on the flow direction, it is either 1 or -1. Since every branch is between two nodes, the summation of all the elements in each row is zero.

The flow equations are described as follows: mass must be conserved at every node implying that $Q_{node} = \sum_{node} Q_{branch}$ and therefore, $A^T_{nxb} Q_{branch} = Q_{node}$. The pressure across a given branch is the product of the flow rate in the branch and the hydrodynamic resistance: $A^T_{nxb} Q_{branch} = Q_{node}$. Based on the supply and demand at each node, $C_{mXn} Q_{node} = Q_0$. Positive (negative) flows will imply that the node is a source (sink) and zero flows will imply this is an intermediate junction. Matrix C has dimensions mxn, where m is the number of flows that are not specified. Q_0 is the flow boundary condition at the nodes. Likewise the boundary conditions for the pressure are given as $D_{qXn} P_{node} = P_0$. Matrix D has dimensions q x n, where q is the number of specified nodal pressures. Consolidated modeling equations are given as follows:

$$A^T Q_{branch} - Q_{node} = 0 \tag{3}$$

$$AP_{node} - RQ_{branch} = 0 \tag{4}$$

$$CQ_{node} = Q_0 \tag{5}$$

$$DP_{node} = P_0 \tag{6}$$

Converting the equations into block matrix form:

$$\begin{pmatrix} A & -R & 0 \\ 0 & A^T & -I \\ 0 & 0 & C \\ D & 0 & 0 \end{pmatrix} \begin{pmatrix} P_{node} \\ Q_{branch} \\ Q_{node} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ Q_0 \\ P_0 \end{pmatrix} \qquad (7)$$

The dimension of the left hand side matrix is $(\boldsymbol{m}+\boldsymbol{q}+\boldsymbol{b}+\boldsymbol{n})$by $(\boldsymbol{2n}+\boldsymbol{b})$. To obtain a unique solution, the number of boundary conditions $(\boldsymbol{m}+\boldsymbol{q})$ must be equal to $\boldsymbol{n}$. Therefore, at every node either the pressure or flow needs to be specified to solve the system of equations. The droplet velocities and the times at which the droplets reach the nodes are calculated based on the solution to these equations.

Each droplet enters through a source node based on a predefined inlet feed frequency. As they travel along the network, they constantly make decisions based on predefined rules, entering and exiting different branches, changing directions along the way until they exit the network. At a given node the droplet decision is determined based on the resistances of the branches that are connected to this node. The drop is moved into the branch that has the least resistance or maximum instantaneous flow rate. The resistances of the branches change dynamically as the droplets move along the network. The time required for droplets in the network to reach the next node along with the droplet entrance and exit times are calculated. A minimum of these times $(\mathbf{t_{min}})$ is then determined to decide which droplet makes the next decision. The droplets are then transported along the network based on $\mathbf{t_{min}}$. This process is performed at every time step until all the termination criteria are met.

The network model has been validated in a number of studies. In our prior work (Maddala et al., 2011), we have shown that the model can predict complex behavior that are experimentally observed in simple loop devices. Further, we have shown that the network model can also be used in ladder devices (Maddala & Rengaswamy, 2014). However, experimental validation of these models in complex interconnected networks still needs to be undertaken.

2.2 Problem statement

Very large droplet-based microfluidic integration (VLDMI) that incorporate precise control of droplets in space and time is demonstrated through the design of platforms that achieve a specific objective. The objective in this paper is to sort - different droplets entering through different arms in a network – into exit arms at pre-specified sequences as shown in Figure 2

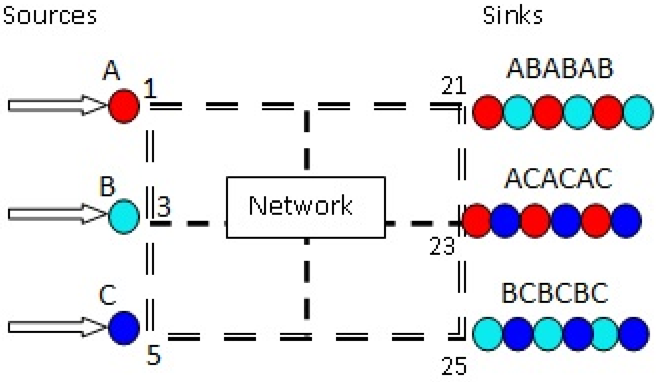


Figure 2: Targeted exit droplet sequence

To obtain a specific sequence of drops at a given exit, an objective function based on the Hamming distance is given as follows:

$$H = abs(S_i - S_d) \qquad (8)$$

where, S_i is the drop sequence obtained by the simulation while S_d is the desired sequence. In a platform with multiple exits, the value of the objective function is obtained by summing over all the exits. The problem formulation for optimizing the structures for a given sequencing objective function is then as follows:

$$Minimize\ H(x_1, x_2, \ldots x_p) \qquad (9)$$

$$Subject\ to:$$

$$[I + \Omega]^k(\gamma, \zeta) > 1$$

$$QX \leq 1$$

$$x_i \epsilon \{0,1\} \, \forall \, i = 1,2, .. p$$

where, X is a column matrix containing all the decision variables. The dimensions of the matrix Q are based on the maximum number of intersections possible in a grid. An intersection in this case is a point, other than the node, where two branches may cross each other. The maximum possible intersections for a n x n grid is $(n-1)^2$. Therefore the dimensions of Q are $(n-1)^2$x p. The value of p is obtained from Equation 10.

$$p = (m-1)(4m-2) \qquad (10)$$

$Q(i,j) = 1$, if x_j is a decision variable that corresponds to an intersection as shown in Figure 3(d), else $Q(i,j) = 0$. Therefore each row of the Q matrix will contain two values of 1 and the rest zero. The initial network is transformed into an adjacency matrix (**Ω**) with the following properties:

i. The adjacency matrix (**Ω**) represents a graph and has dimensions n x n, where n is the number of nodes in the graph/network.

ii. $\Omega(i,j) = 0 \; or \; 1; i \leq n \; and \; j \leq n$, the value of 1 implies that a branch exists between node i and j, otherwise is 0.

In converting a given network into the nodal, it is realized that the network might be disjoint, *i.e.*, there might be some branches without any connections and some branches that are not connected to a source or sink. This could arise if a user or the design algorithm leaves some branches unconnected or intersect the branches without specifying an intersection point (Figure 3) or a source/sink not connected resulting in an infeasible solution.

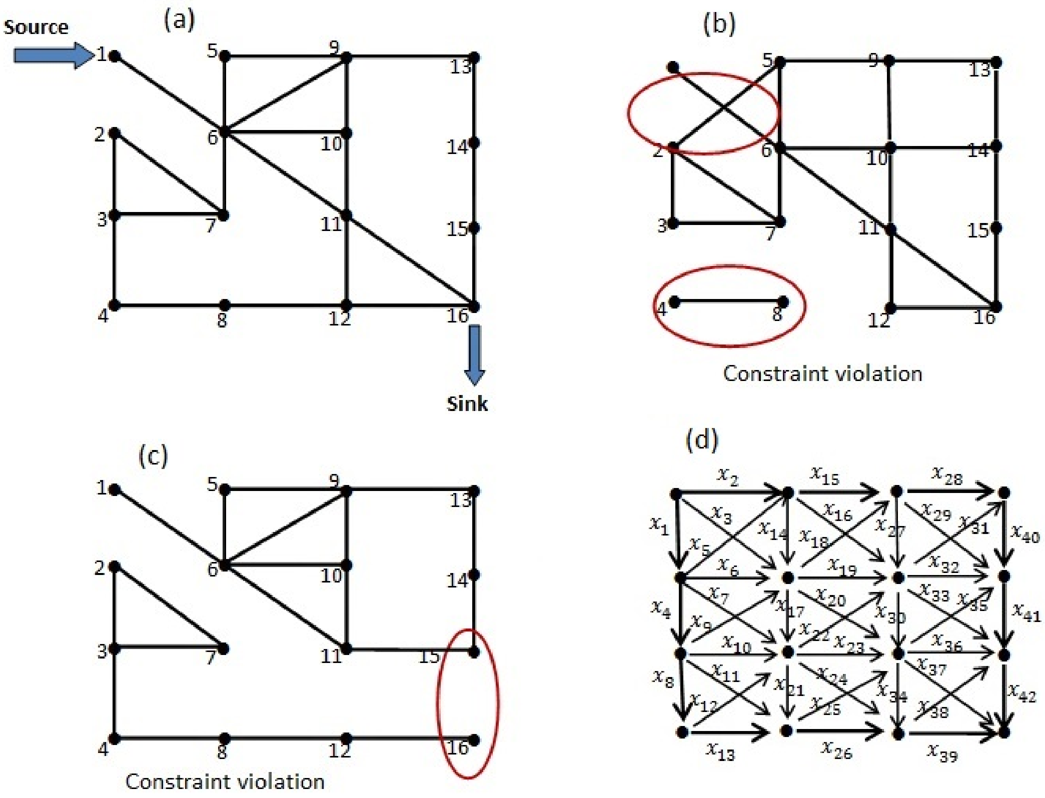


Figure 3: A normal network (a) and one that violates constraints ((b) and (c)) and network variables (d).

Thus, graph theory concepts were employed to implement the first set of constraints to avoid such disjointed networks and ensure that the network problem has a solution. The k in the above constraints equations refers to the maximum distance between two nodes of a graph and this value is $n-1$. The term γ consists of all the source nodes. $\zeta$is the set of all the sink nodes plus the nodes that have a value of one.

As both the objective function and the constraints are nonlinear, the formulated problem can be characterized as a binary integer nonlinear programming problem. In general these problems are solved by classical deterministic gradient-based methods that are used for solving mixed integer nonlinear programming (MINLP) problems. Such methods may include the branch-and-bound (Parker & Rardin, 1988; Gupta & Ravindran, 1985; Quesada & Grossmann, 1992), Extended Cutting Plane methods (Westerlund & Petersson, 1995), Generalized Benders Decomposition (Geoffrion, 1972) etc. However, the lack of a closed form analytical expression for the objective function for typical design problems would make it difficult to use such deterministic methods. Additionally, repeated gradient evaluations during the simulation would be numerically cumbersome as the objective function is calculated through numerical simulations (Maddala & Rengaswamy, 2014). Use of alternative derivative-free methods/algorithms that require only the availability of objective function values but no derivative information is an alternative approach. Examples of such methods include the Nelder-Mead simplex algorithm (Nelder & Mead, 1965), particle swam algorithms (Kennedy & Eberhart, 1995; Eberhart & Kennedy, 1995), genetic algorithms (Holland, 1975).

A genetic algorithm (GA) method was adopted for the current design approach. GAs have shown promise in efficiently solving combined structural and continuous problems (Garrard & Fraga, 1998; Maddala & Rengaswamy, 2014; Ravagnani et al., 2005). GAs are also reported (Maddala & Rengaswamy, 2014) to have the added advantage of identifying several acceptable designs due to the principle of hyper-plane sampling. The appeal for genetic algorithm formulation is that they also are easily scalable through parallel computing and are extendable to MINLP problems. However, it is difficult to prove convergence of GA algorithms and global optimality is hard to establish.

2.3 Design approach

As mentioned, a genetic algorithm (GA) optimization approach is used for the design of large scale microfluidic networks. Genetic algorithm was successfully implemented in our previous work (Maddala & Rengaswamy, 2014) to design ladder networks. The major difference is in the node-branch architecture of the current GA which helps to break down the droplet-based microfluidic networks to the lowest possible model description. This means that any complex architecture can be represented by assembling the lowest level model fragments. An implementation of this GA approach is illustrated in Figure 4.

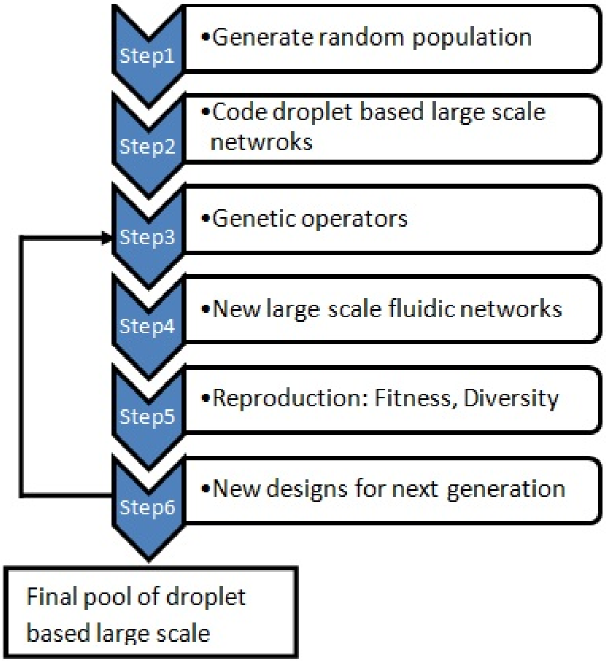
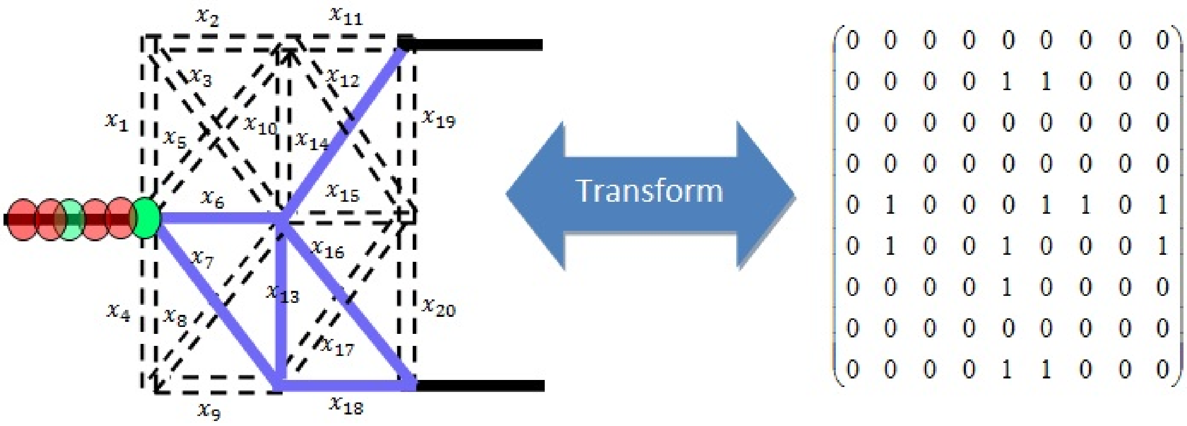

Step1
•Generate random population
Step2
•Code droplet based large scale netwroks
Step3
•Genetic operators
Step4
•New large scale fluidic networks
Step5
•Reproduction: Fitness, Diversity
Step6
•New designs for next generation
Final pool of droplet based large scale

Figure 4: Proposed approach for modeling and simulation of large scale microfluidic devices

Once the structure grid and general properties of the network are defined, the GA is initialized by generating a random population of M networks that satisfy the constraints. Later on, the GA operators are applied in order to progress the population to obtain networks that meet the design objective. The key GA components: mutation, cross-over, reproduction or selection, and fitness function are described in the next subsection.

Coding-Decoding

A coding scheme that involves the representation of microfluidic networks as an adjacency matrix is formulated. The variables $\{x_1, x_2, x_3, \ldots\ldots x_p\} \epsilon \{0,1\}$ characterize connections in the network; a value of 1 implies that there is a connection and 0 otherwise. As an illustration, the adjacency matrix for a 3x3 grid is shown in Figure 6. For a given network of grid size m x m, the maximum number of variables required in the matrix is given by Equation 10. Therefore, the total variables required to define the geometry of the network on this grid system is 20.

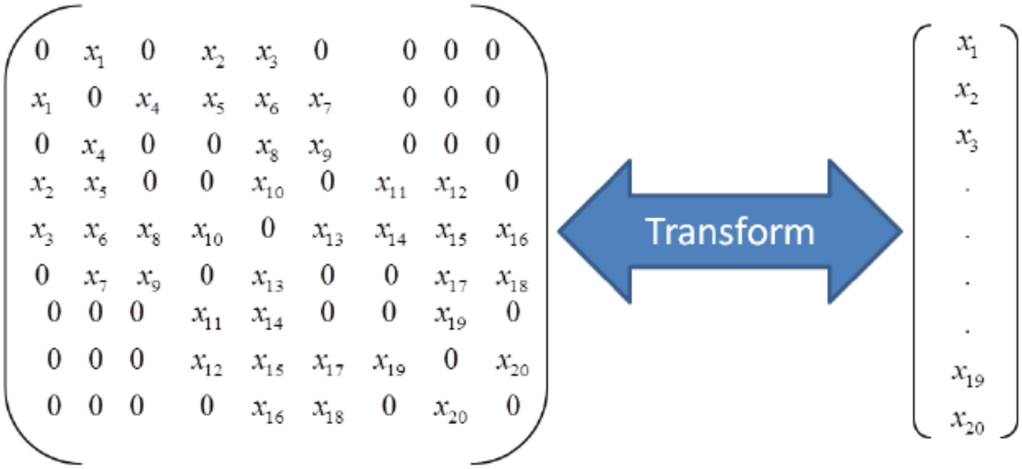


$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Figure 5: Coding of Microfluidic network: grid network as represented by adjacency matrix

In Figure 5, five out of nine nodes are utilized to construct the network. Solid lines between the nodes imply that a branch exists, whereas a dashed line between the nodes implies the absence of a branch. This information is transformed into the adjacency matrix by making the corresponding rows in the matrix completely zero. Figure 6 shows a generic adjacency matrix for a 3x3 grid system. However, this approach requires implementation of genetic operations on large sparse matrices, resulting in considerable computation time. Therefore, the coding scheme was reformulated by transforming the adjacency matrix to a single vector consisting of binary values, as illustrated in Figure 6. The binary string contains all the information from the sparse adjacency matrix. This transformation results in an easier decoding and decreased computation time.

The adjacency matrix shown in Figure 5 contains redundant information because of having zero rows. The redundant information is also transformed into the coded binary string as shown in Figure 6. It is important to retain this redundant information because it results in coded structures of fixed lengths. The development of the algorithm would become extremely cumbersome on binary strings of varying size. For example, the generation of the initial population of design structures is straight forward for random binary strings for fixed length. Moreover, it is easier to identify and eliminate any repeating structures using strings, which enhances diversity in the population.



$$\begin{pmatrix} 0 & x_1 & 0 & x_2 & x_3 & 0 & 0 & 0 & 0 \\ x_1 & 0 & x_4 & x_5 & x_6 & x_7 & 0 & 0 & 0 \\ 0 & x_4 & 0 & 0 & x_8 & x_9 & 0 & 0 & 0 \\ x_2 & x_5 & 0 & 0 & x_{10} & 0 & x_{11} & x_{12} & 0 \\ x_3 & x_6 & x_8 & x_{10} & 0 & x_{13} & x_{14} & x_{15} & x_{16} \\ 0 & x_7 & x_9 & 0 & x_{13} & 0 & 0 & x_{17} & x_{18} \\ 0 & 0 & 0 & x_{11} & x_{14} & 0 & 0 & x_{19} & 0 \\ 0 & 0 & 0 & x_{12} & x_{15} & x_{17} & x_{19} & 0 & x_{20} \\ 0 & 0 & 0 & 0 & x_{16} & x_{18} & 0 & x_{20} & 0 \end{pmatrix} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ . \\ x_{19} \\ x_{20} \end{bmatrix}$$

Figure 6: Coding of microfluidic networks: Adjacency matrix into single vector

Genetic operators

Three standard genetic algorithm operators; mutation, two-point crossover and tournament selection are proposed in the current work.

Mutation

During mutation, a bit in a given binary structure is randomly changed from 0 to 1 and vice versa thereby modifying the structure of the microfluidic network. This operator results in branches from a given structure being randomly added or deleted, thereby increasing the diversity of the population. The mutation operation is performed based on a predefined probability which is an input parameter to the algorithm.

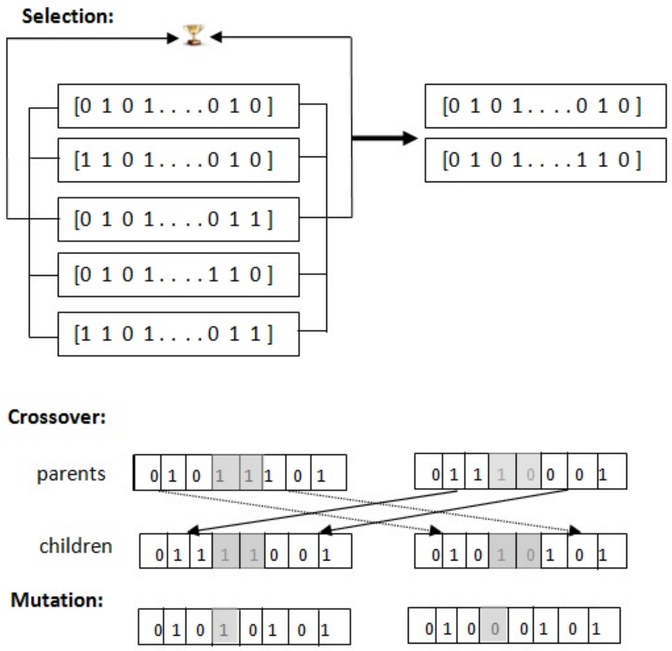


Figure 7: Genetic operations

Two-point crossover

In two-point crossover operation, two structures (parents) are combined to create a new pair of structures (children) as shown in Figure 7. Two parent microfluidic structures are picked randomly from the population and two daughter structures are generated. Consider the two parent structures to have strings $PS_1$and PS_2, the off-spring structures will have strings, PS' and PS'' which are calculated as:

$PS' = [PS_2(1{:}\,\alpha - 1)PS_1(\alpha{:}\,\beta)PS_2(\beta + 1{:}\,p)]$ (11)

$PS'' = [PS_1(1{:}\,\alpha - 1)PS_2(\alpha{:}\,\beta)PS_1(\beta + 1{:}\,p)]$ (12)

Where α and β represent a location within the parent structures where crossover occurs. They are integers randomly generated from 2 to the length of the parent structures(p). A one-point crossover operation can also be used for this purpose. The off-springs generated from the cross-over operation inherit properties from both their parents. It is important to note that once the children are produced, they are tested for constrain violation. The process of reproduction is repeated until children that meet the specified constraints are obtained. They are then retained as part of the new population.

Selection

The main objective of the selection operation is to create future generations of structures from the existing population by terminating non-elitist individuals (structures). After the GA is initialized by generating "N" random structures, the genetic operators are performed on these structures generating new structures. The selection step is performed in combination of elitist and diversity principles. The elitist approach will pick structures with maximum fitness. The selection continues until a new population of structures of size "N" is generated. However, if all these structures are similar, the proposed GA leads to a local optimal solution. Hence the diversity approach helps maintaining a diverse population at every generation. This diversity principle also ensures that the final population has several solutions for a given fitness function. The future generations are made up of the elite solutions (whose size is an input parameter to the model) that are carried over from the current generation and the children generated from the selection operator.

The above two principles are captured by fitness (F) and diversity (D) metrics. The former quantifies the structures with respect to a given objective function while the latter quantifies the similarity of a given structure with the rest of the population (Maddala & Rengaswamy, 2014). The diversity metric (D) used to estimate the diversity between two microfluidic structures in the current study is defined based on the "2-norm"vector norm as shown below:

$T_i(j) = \sum_{j=1}^{p}(PS(i) - PS(j))^2, \ \forall i = 1{:}N \& j \neq i$ (13)

$D(i) = \ minimum(T_i); \ \ \forall i = 1{:}N$ (14)

The above equation to calculate D is equivalent to calculating the Euclidean distance between two vectors. It can be seen that each element (j) in vector T_i quantifies the difference between the i^{th} and j^{th} structures in the population. The smaller the value of $D(i)$, the less the diversity between two given vectors and a zero value of $D(i)$ implies that an exactly similar structure exists in the population.

The fitness function is generally determined based on the design objective. In fact the success of the GA algorithm depends on identifying appropriate fitness function. The fitness function of the current work is provided in the next section.

Fitness Function (***F***)

The emphasis of the current work is to design VLDMI networks that can achieve specific objectives. In doing so, we ask the question: can we use the GA approach to discover microfluidic networks that can achieve combinatorial sequencing of three different droplets to three specific pre-defined sequences? In answering the above question, it was important that an appropriate objective/fitness function is defined for the GA algorithm. A pictorial representation of the problem is shown in Figure 2.

The three droplet sequence types ("AAAA", "BBBB", "CCCC") enter the device via three different sources (A,B &C) located at nodes (1, 3 & 5) respectively and each of the desired sequences must exit via the three sink nodes (21, 23, & 25). The sequences 'ABAB','ACAC' and 'BCBC' are desired in the top, middle and bottom exit sinks respectively. The three droplet types were color coded as red, blue, and cyan. A Hamming distance-based algorithm was utilized in formulating the objective or fitness function using the following considerations:

i. The total number (N_t) of droplets exiting in a given sink and subsequently the total number of droplet pairs (n_p) in this set of droplets are determined. The droplets that are not part of the desired sequence in that particular sink are also counted (N_1). The remaining droplets ($N_2 = N_t - N_1$) are the set of droplets constituting the desired sequence for that particular outlet.

ii. The pair(s) of droplets ($n_{p,d}$) making up the desired sequence in each sink are also determined.

iii. A penalty (φ_i) is associated with the presence of the undesired droplets in the set of drops exiting in the i^{th}sink or outlet. It is calculated as the ratio of the number of undesired droplets to the total number of droplets in the i^{th}sink or outlet nodes is

$$\varphi_i = \frac{N_1}{N_t} \qquad (15)$$

This penalty is however applicable if there are droplets leaving in a given sink otherwise the maximum penalty value of -1 is imposed if no droplets are leaving in that sink. A value (ε_i) given in Equation 16 is calculated for every sequence in the i^{th}sink. This represents the value of the fitness function in every exit arm i.e.

$$\varepsilon_i = \frac{n_{p,d}}{n_p} - \varphi_i \qquad (16)$$

where, $n_{p,d} = floor\left(\frac{N_2}{2}\right)$ is the number of pairs of droplets available to form the desired sequence.

The overall value of the objective function or fitness is then given as

$$F = \left(\sum_i^k \varepsilon_i\right) \qquad (17)$$

It can be noted that the maximum value of the fitness function is 3 if a perfect sequence is achieved in each sink. The GA progresses in such a way that the fitness function value is maximum and this ensures that the perfect droplet sequences exit the device.

3. Results and Sensitivity Analysis

3.1 Results

The designs that satisfy the objective function along with their design parameters is presented in the beginning of the section. Several metrics are proposed that help in comparing these results. A thorough analysis on the performance of genetic algorithm along with the sensitivity analysis to the parameters of algorithm are presented later in this section

A total of 71 droplets of each type (color) enter the device. The maximum number of nodes in the microfluidic network is fixed as 25; this translates into a network with grid of size 5x5 and the size/length of design variables vector(p) to 72. The source nodes are 1, 3, and 5 and the sink nodes are 21, 23 and 25. The bulk fluid flow rate is 450 μL/hr, the droplet resistance is 10.016 kg/mm^{4}s, and the slip factor (β) is 1.3717. Experimental uncertainties such as: noise in droplet feeding frequency, and fabrication errors in lengths of the channels are incorporated in the model. A uniform distribution-based randomly generated value is added to both channel lengths and droplet frequency. Additional model parameters used in this work are provided in Table 1.

Table 1.0 Summary of key simulation parameters that were used in the current study

Parameter	**Value**
Population size	50-100
Elite population size range	4-15
Droplet entry frequency($\boldsymbol{f_d}$) (s^{-1})	21.70
No. of droplets in each source	71
Network grid dimension	5x5
Grid spacing (µm)	1500
Rectangular channels dimensions, µm (height x width)	50x100

Design I, shown in Figure 8, is a network discovered by the GA that satisfies the objective function. The figure shows four snapshots taken at different times after the drops enter the platform; these snapshots highlight the dynamics of the drops within the platform before attaining steady state. Simulation is started with no droplets in the network, as shown in Figure 8(a). In the transient phase Figures (8(b) and (c)), interesting behavior is observed, for example, a certain droplet translates back and forth as shown in the highlighted area of Figure 8(b) before moving into another branch. Steady state behavior emerges after several transients and the combinatorial sequencing is achieved as shown in Figure 8(d). There are approximately the same number of drops of each type in the network due to the uniform feeding frequency

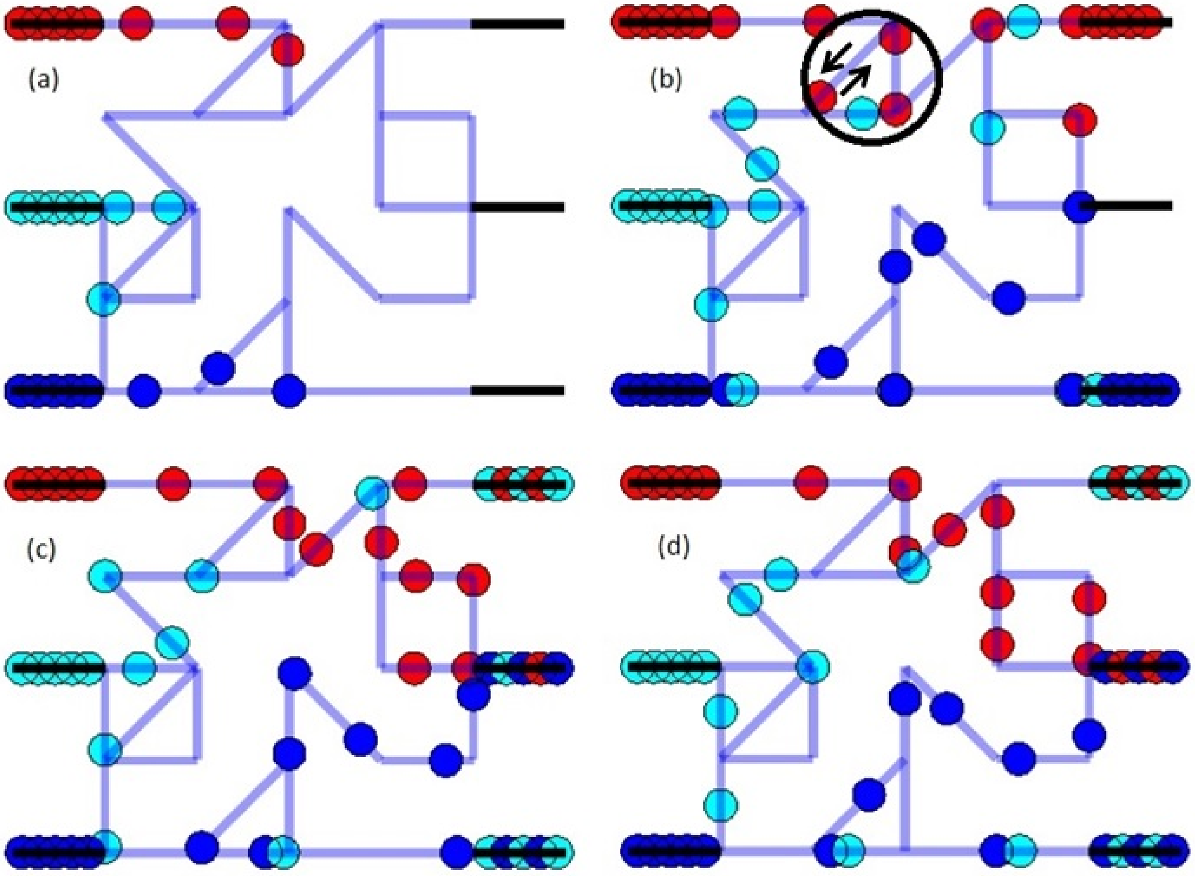


Figure 8: *Design 1*(a) simulation started with no drops in the network (b) transient of droplet motion (highlighted area shows droplet moving back) (c)network close to reaching a steady state (d) network at steady state

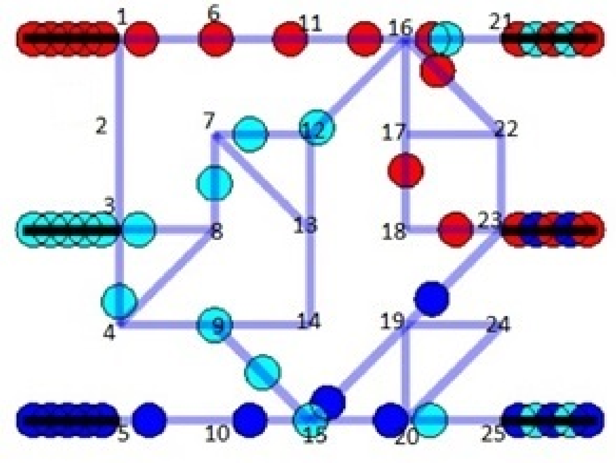


Figure 9: *Design* II

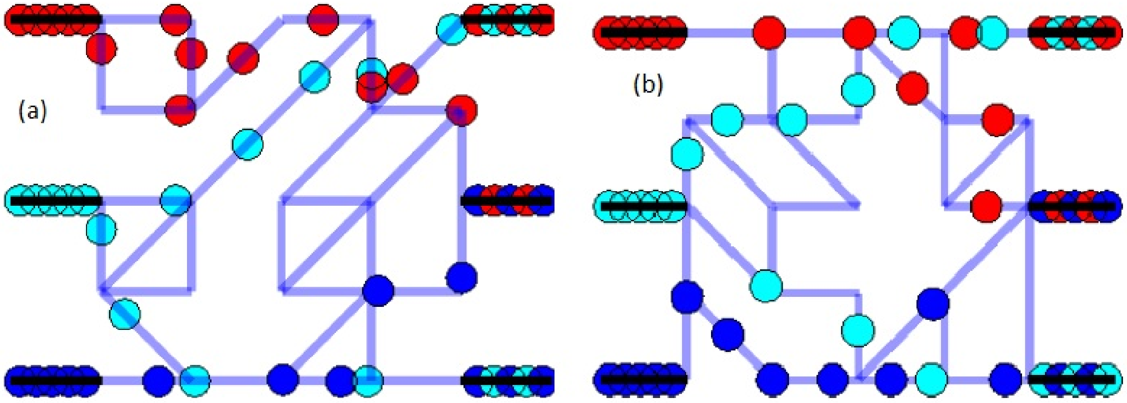


Figure 10: Additional microfluidic network designs (a) *Design III* (b) *Design IV*

Additional designs, that satisfy the objective function, are presented in Figures 9 and 10 (a, b). In these designs, only the final steady state position of the droplets is depicted. Despite the differences in the geometrical structures of these networks, it was interesting to find similarities in the droplet dynamics. In most of the designs, the top and bottom sequences are achieved faster compared to the middle sequence. The sequences in the top and bottom can be achieved via a direct route, whereas the middle sequence is obtained via a convoluted route; therefore it takes longer time to achieve the sequence in the middle exit. These results demonstrate the ability of the GA approach to optimally identify diverse solutions for a given objective function.

In this work, we propose novel metrics to quantify the microfluidic networks based on droplet motion, geometric structures and noise tolerance. The objective of this quantification is to understand the advantages and disadvantages of choosing a particular design over another; this information will be invaluable in fabrication. The unique advantage of GA is that, these metrics can be further incorporated in the optimization using a penalty. The metrics proposed to quantify the results are as follows:

(i) ***Number of branches*** ($\overline{\boldsymbol{N}}_{\boldsymbol{brchs}}$)- The total number of branches that comprise the network normalized with the maximum possible number of branches.

(ii)***Minimum decision time difference*** ($\overline{\boldsymbol{t}}_{\boldsymbol{min}}$)**:**This is the minimum time difference between two decision events normalized with t_{dl}. The time required to travel one droplet length is defined as $t_{dl} = \frac{d_l}{\left(\beta Q_o/_S\right)}$, where d_l is the droplet length and S is the cross sectional area of the channel. The value is obtained from the experimental data(Maddala, et al., 2011). It is shown that, at $\bar{t}_{min} \approx 0$, the droplet dynamics transitions from a steady periodic state to a chaotic state (Maddala et al., 2014) , therefore a design with higher $\bar{t}_{min}$ is better as it will be robust to experimental fluctuations .

(iii) ***Minimum difference in branch resistance***($\overline{\boldsymbol{R}}_{\boldsymbol{diff,min}}$)**:** The difference in resistance between two branches when a drop makes a decision. A higher $\bar{R}_{diff,min}$ value implies a higher stability for the network.

(iv) S***teady state time*** ($\overline{\boldsymbol{t}}_{\boldsymbol{ss}}$)**:** The time it requires to reach steady state normalized to the time ($t_{\max}$) droplet would take to travel between two corners in the longest possible distance for a given grid dimension. The value of $t_{\max}$for a grid dimension of $mxm$is $\frac{(m-1)L_{grd}}{\left(\beta Q_o/_S\right)}$, where L_{grd} is the distance between two grid points or the length of the branch. The lower the value of $\bar{t}_{ss}$ the better the design as the droplets would have travelled less distance before achieving steady state.

(v) ***Steady state exit drops***($\boldsymbol{N}_{\boldsymbol{drps,exit}}$)***:*** The percentage of the feed drops that exit before the desired sequences is obtained. The lower the value of $N_{drps,exit}$, the better the performance of the network.

(vi) ***Noise tolerance***($\boldsymbol{\chi}$)**:** The simulation is performed by gradually increasing the noise levels in the droplet inlet time (f_d^{-1}). The standard deviation in the f_d^{-1} is increased from 10^{-10} to 10^{-3}.

The highest noise level for which the network satisfies the objective function is labeled as χ. The higher the value of χ, the higher robustness to disturbances in droplet generators.

The proposed metrics are evaluated for all the 4 designs and are tabulated in Table 2.

Table 2: Summary of major design performance metrics

Design (Figure #)	$\bar{t}_{min}$	$\bar{R}_{diff,min}$	$\bar{N}_{brchs}$	$N_{drps,exit}(\%)$	$\bar{t}_{ss}$	χ
1(Fig. 8)	5.08E-05	2.86E-11	0.682	98 (46.0)	10.979	10^{-4}
2(Fig. 9a)	2.24E-03	6.62E-11	0.750	27 (12.7)	4.478	10^{-4}
3 (Fig. 10a)	9.62E-03	7.28E-11	0.750	29 (13.6)	4.724	10^{-5}
4 (Fig. 10b)	3.55E-09	9.12E-11	0.773	21 (9.85)	4.016	10^{-4}

From Table 2, designs 2, and 4 have the same number of branches and approximately same $N_{drps,exit}$and $\bar{t}_{ss}$. In these designs, $\bar{t}_{ss}$ is directly proportional to the $\bar{R}_{diff,min}$ but no specific trend on $\bar{t}_{min}$. In case of designs 1 and 4, the increase in the number of branches is compensated by delay in achieving the steady state and a low value of $\bar{R}_{diff,min}$. Decrease in the number of branches makes the network more congested as same number droplets travel in these networks. This is reason for having a lower $\bar{R}_{diff,min}$ and a higher $\bar{t}_{ss}$, in case of design 1. Based on the metrics provided in Table 2, the network with highest stability is design 3; this design has the highest $\bar{t}_{min}$ and $\bar{R}_{diff,min}$. Under increased experimental fluctuations, design 4 still emerges as the best design since it achieves the sequences within the shortest time. It was interesting to note that designs 1,2& 4 were robust to the same levels of noise.

In addition to the above design parameters, a number of visual topological features were noted some of which can be used to explain the above observations. The droplets (cyan) from the middle sink travel in two different directions towards the top and bottom sinks/exits where they are utilized to obtain the desired sequences. The shortest distance traveled by these droplets is equivalent to 5 branch lengths as seen in designs II, III and IV. Similarly, the minimum distances traveled by the droplets (red) from the top source is 4 branch lengths towards the top sink (designs II and IV) and 5 branch lengths towards the middle sink (design IV). Lastly the droplets (blue) from the bottom source also traveled a minimum of 4 branch lengths on their way to the bottom and middle sinks as seen in designs I, II and III. In all the designs, the droplets traveled the longest distances in design I and this explains the longest time it took for the design to achieve the desired sequences. It was particularly interesting to note that the desired sequences were formed quicker in cases where the two droplets forming the sequences traveled the same number of branches. This was the case in design IV, in which the droplets forming the sequences in the bottom and middle sinks/exits covered same distances. In combination with the minimum distance covered by the red droplets towards the top sink, the above observations are consistent with the observation that design IV achieved steady state in the shortest time. Similar reasoning can be applied to explain the observed design steady state times. It was also observed that not all the branches in the designs were utilized to achieve the desired sequences. The percent of branches utilized in the branches were 86.7, 69.7, 66.7, and 76.5% for designs I, II, III and IV respectively. This can be useful information during the fabrication of the designs if the redundant branches can be eliminated without affecting the performance of the network.

Whereas each of the above designs were obtained from different runs, it was quite remarkable that in each of the given runs, it was possible for the GA to obtain multiple designs/solutions. This was the case for the run corresponding to the design of Figure 10a. A solution is obtained when the maximum fitness is obtained as shown in Figure 11.

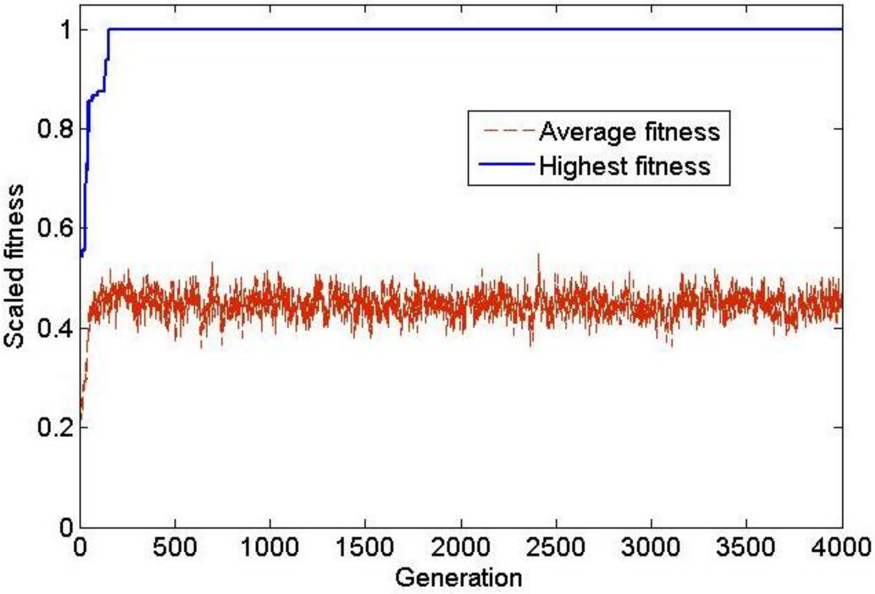


Figure 11: Highest and average fitness values as a function of generation.

In Figure 11, the profiles of the highest and average fitness values of the population as obtained from the objective function are plotted. It can be seen that the simulation progresses with the highest fitness values steadily increasing until it reaches the maximum value of 1 at which point the GA has found networks that achieve the desired sequences.

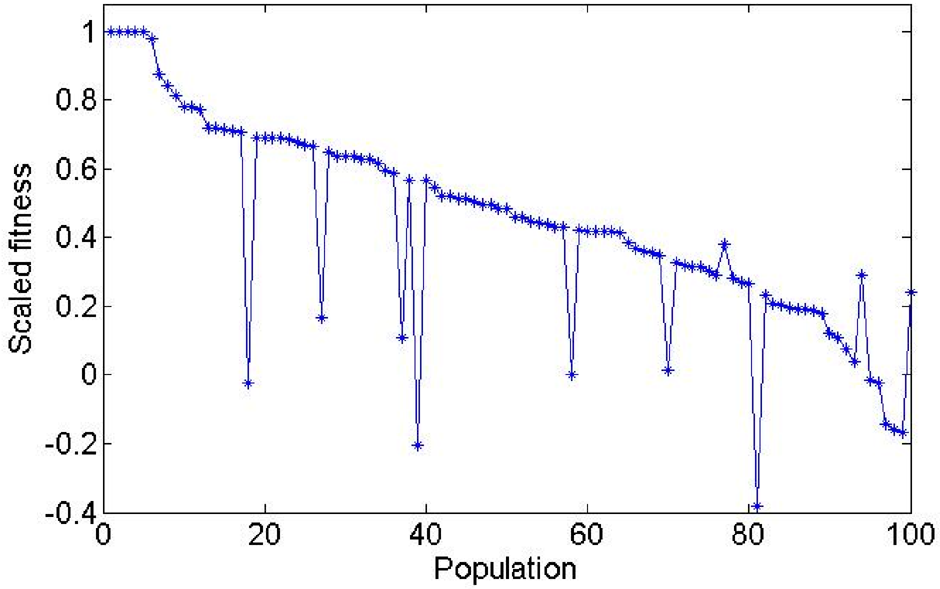


Figure 12: Final population fitness

When the fitness distribution of the final population was plotted as shown in Figure 12, it is evident that a total of five structures/designs (with maximum fitness values) were identified by the GA since it is ensured that there are no repeating structures or individuals in the population. This is an important advantage of the GA approach as compared to other optimization approaches in which a single solution is obtained.

3.2 Sensitivity analysis

Sensitivity analysis on the key parameters, of the genetic algorithm, is presented in this section. The parameters include mutation probability, population size, elite population size and the type of cross-mutation. The effect of varying each of these parameters on the genetic algorithm is presented in Figures 13-15

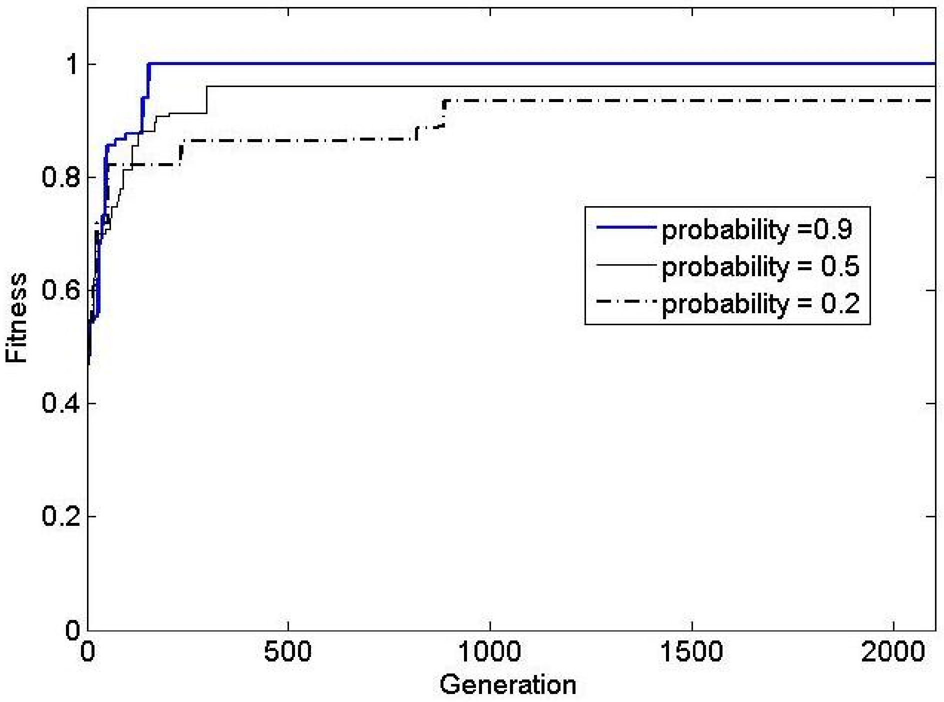


Figure 13: The effect of mutation probability on the maximum population fitness

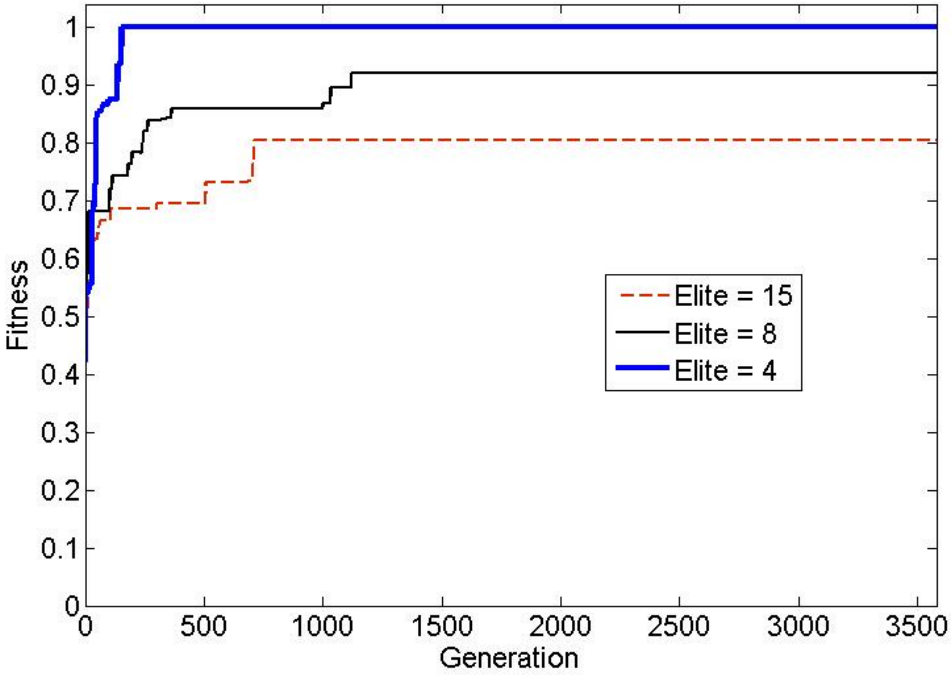


Figure 14: Effect of elite population size on the maximum population fitness

Figure 13 shows the effect of mutation probability on the performance of the GA. A mutation probability of 0.9 is found optimal; GA has attained solution just under 500 iterations at this probability. As the mutation probability is reduced the number of iterations to attain a solution has increased. GA never converged to a solution for mutation probabilities less than 0.05. At the mutation probability of 1, the algorithm took more generations to achieve the solution; further having only mutation made the final population less diverse. Therefore optimal mutation probability of 0.9 is used in the current simulation.

The effect of the elite solution size is contrary to that of the mutation probability as shown in Figure 14. The smaller the elite population sizes the faster the GA in achieving maximum fitness. Increasing the elite size resulted in lower fitness values and ultimately a failure to reach the maximum fitness values. It is suspected that a lower value of the elite size results in the GA obtaining a more diverse population which seems to aid in the attainment of the maximum fitness values.

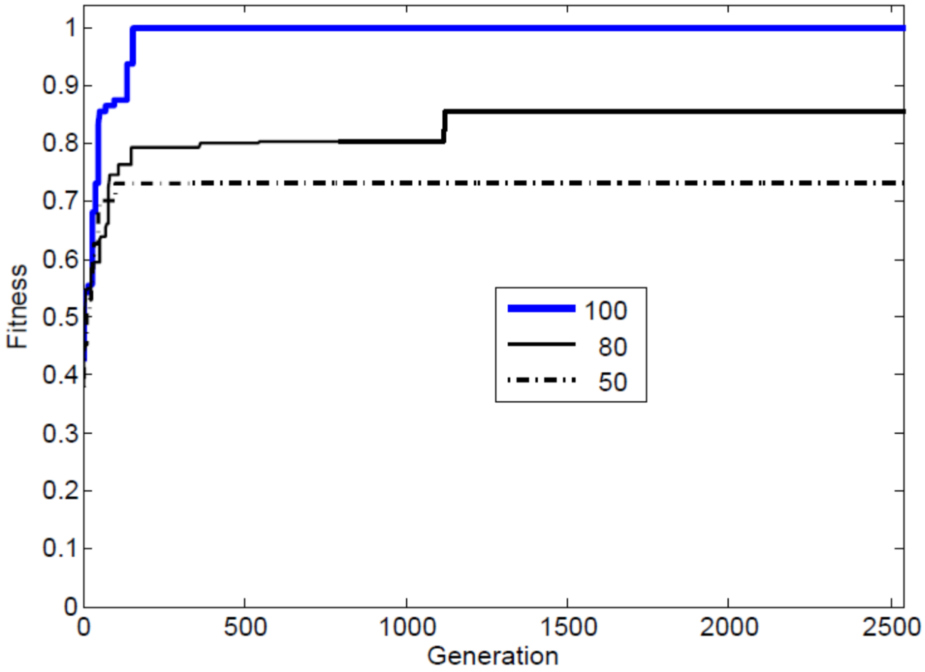


Figure 15: Variation of maximum fitness values with population sizes of 100, 80 and 50.

Figure 15 shows the effect of population size on achieving the results. The variation of the maximum fitness was studied with population sizes of 100, 80 and 50.GA was able to reach a solution (at maximum fitness value of 1.0) in the fewest number of iterations for a population size of 100. GA was not able to obtain a solution even after prolonged simulation time (2500 iterations) for population sizes of 80 and 50. The algorithm reached a maximum fitness value of ~0.85 and ~0.73 for population sizes of 80 and 50 respectively. GA was able to attain a solution at less number of iterations for population sizes more than 100 but this resulted in increase in computational burden because of more function evaluations. Therefore a population size of 100 was used in the current study.

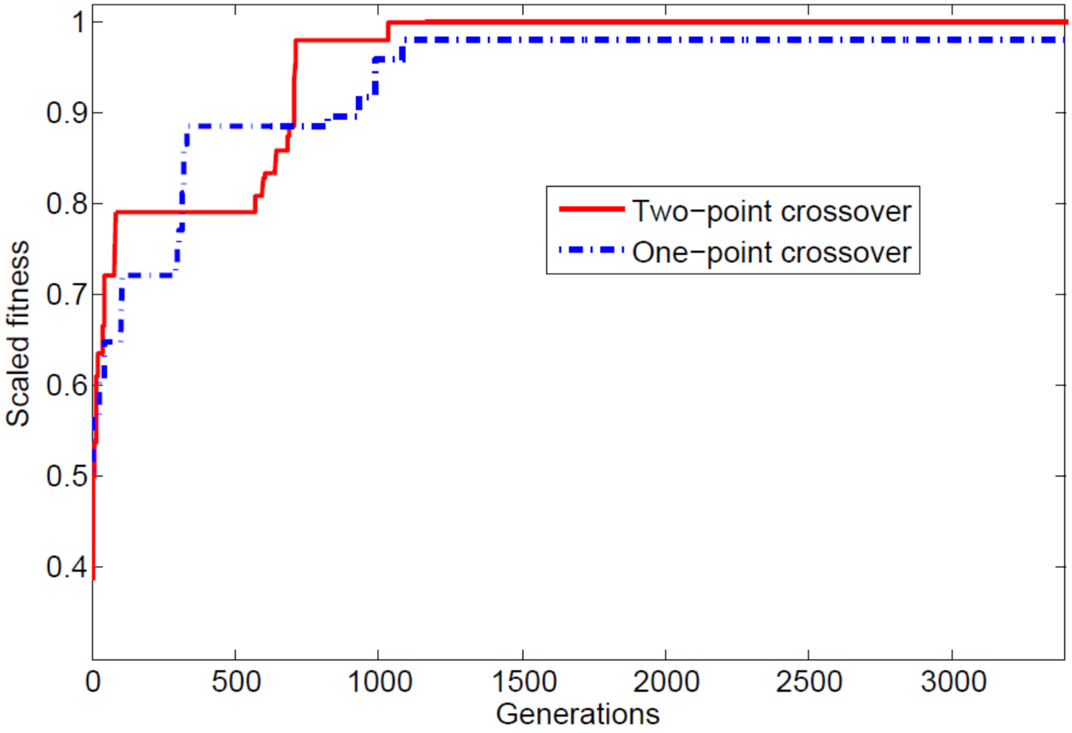


Figure 16: Comparison of performance of two-point and one-point crossover mutation

The effect of a one-point crossover and a two-point crossover process on the progress of the GA is shown in Figure 16. The highest fitness value of the population from the two-point and one-point crossover operations runs are plotted in the figure. Both runs were performed with the same initial population to avoid the well-known dependence of the GA on the quality of the initial population. From the results, the run with the two-point crossover obtained a solution after about 1000 generations while that with a one-point crossover did not obtain a solution even after about 3500 generations. Clearly, the two-point crossover operation performed much better than the one-point crossover operation and thus was preferred in the current study.

4. Conclusions

In the current study, we presented a novel route for potentially identifying very large scale integrated droplet-based microfluidic networks for a given objective function. The study revealed multiple interesting but non-obvious designs that satisfy the given objective function.

The current work identifies only the geometry of the microfluidic devices given an objective function. In addition to the geometric parameters, the design parameters of these structures such as branch dimensions, flow rates etc., have to be carefully identified by the GA approach.

The trial and error experimental procedure to identify such designs is time and labor intensive. As the number of chemicals and sequences scale-up it would be impossible to identify the design parameters using pure experimental techniques. Moreover, if one were to require intermediate processing steps with different residence times, the design problem would be even more complex. The power of the GA approach is that any of these design requirements, and combinations thereof can be incorporated into the fitness function with minimal effort and the other part of the design framework need not be modified at all.

Through sensitivity studies, it was observed that with a population of 100 individuals, a smaller elite population size and a higher mutation probability favored faster convergence of the GA to solution when a two-point crossover operation was used.

The generic nature of the current design approach helps in obtaining microfluidic platforms for different objective functions. Moreover, with recent advances in the micro-fabrication techniques, the current tool can be a significant step towards drastically cutting down on expensive and time intensive trial-and-error design process. One immediate avenue for future work is the experimental verification of the GA inspired large-scale microfluidic designs.

Acknowledgement

The authors gratefully acknowledge partial support from NSF Grant CMMI-1124814 titled “CDI-Type I: Engineering Massively Parallelized Fluidics Processors: From Data to Predictive Models to Functional Designs”. The authors further acknowledge the support from NSF through grant no. NSF STTR-1321506 titled " Development of a Computational Tool for Modeling, Simulation, and Design of Next Generation Discrete Droplet Microfluidic Systems"

Abbyad, P., Dangla, R., Alexandrou, A., & Baroud, C. (2011). Rails and anchors: guiding and trapping droplet microreactors in two dimensions. *Lab Chip*, 813.

Ajdari, M. S. a. A. (2008). Droplet Traffic in Microfluidic Networks: A Simple Model for Understanding and Designing *Physical Review Letters*.

Bithi, S., & Vanapalli, S. (2010). Behavior of a train of droplets in a fluidic network with hydrodynamic traps *Biomicrofluidics* 044110.

Boukellal, H., Selimovic, S., Jia, Y., Cristobal, G., & Fraden, S. (2008). Simple, robust storage of drops and fluids in a microfluidic device. *Lab on a Chip*, 331-338.

Brouzes, E., Medkova, M., Savenelli, N., Marran, D., Twardowski, M., Hutchison, J. B., et al. (2009). Droplet microfluidic technology for single-cell high-throughput screening. *Proceedings of the National Academy of Sciences, 106*(34), 14195-14200.

Christopher, G. F., Bergstein, J., End, N. B., Poon, M., Nguyen, C., & Anna, S. L. (2009). Coalascence and splitting of confined droplets at microfluidic junctions. *Lab on a Chip*, 1102-1109.

Cristobal, G., Benoit, J. P., Jaonicot, M., & Ajdari, A. (2006). Microfluidic bypass for efficient passive regulation of droplet traffic at a junction *Applied Physics Letters*, 1-3.

Cybulski, O., & Garstecki, P. (2009). Dynamic memory in a microfluidic system of droplets traveling through a simple network of microchannels. *Lab on a Chip*, 1102-1109.

Design of a model-based feedback controller for active sorting and synchronization of droplets in a microfluidic loop. (2011). *AIChE*, 2120-2130.

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *In Proceedings of the sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan*, 39-43.

Fuerstman, M., Garstecki, P., & Whitesides, G. (2007). Coding/decoding and reversibility of droplet trains in microfluidic networks. *Science*, 828-832.

Garrard, A., & Fraga, E. (1998). Mass exchange network synthesis using genetic algorithms. *Computers & Chemical Engineering*, 22 (12), 1837-1850.

Geoffrion, A. M. (1972). A Generalized Benders Decomposition *J. Optim. Theory Appl.*, 10 (14), 237-260.

Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading, Massachusetts: Addison-Wesley.

Gupta, O. K., & Ravindran, A. (1985). Branch and Bound Experiments in Convex Nonlinear Integer Programming. *Manage Sci*, 31 (12), 1533-1546.

Haeberle, S., & Zengerle, R. I. (2007). Microfluidics platforms for lab-on-chip applications. *Lab Chip*, 1094-1110.

Hatakayema, T., Chen, D. L., & Ismagilov, R. F. (2006). Microgram-scale testing of reaction condition in solution using nanoliter plugsin microfluidics with detection by MALDI-MS *Journal of the American Chemical Society* 2518.

Holland, J. (1975). *Adaptation in natural and artificial systems*. Cambridge, Massachusetts: The MIT Press.

Huebner, A., Srisa-Art, M., Holt, D., Abell, C., Hollfelder, F., deMello, A., et al. (2007). Quantitative detection of protein expression in single cells using droplet microfluidics *Chem Commun.*, 1218-1220.

Jeevan Maddala, R. R. (2014). Design of multi-functional microfluidic ladder networks to passively control droplet spacing using genetic algorithm. *computers and Chemical Engineering*, 413-425.

Jin, B., Kim, Y. W., Lee, Y., & Yoo, J. Y. (2010). Droplet merging in a straight microchannel using droplet size or viscosity difference. *Journal of Micromechanics and Microengineering*, 1-3.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *In Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, USA*, 1942-1948.

Laval, P., Crombez, A., & Salmon, B. J. (2009). Microfluidic droplet method for nucleation kinetics measurements. *Langmuir*, 1836-1841.

Maddala, J., Srinivasan, B., Bithi, S. S., Vanapalli, S. A., & Rengaswamy, R. (2011). Design of a model-based feedback controller for active sorting and synchronization of droplets in a microfluidic loop. *AIChE Journal, 58*(7), 2120-2130.

Maddala, J., Vanapalli, S. A., & Rengaswamy, R. (2014). Origin of periodic and chaotic dynamics due to drops moving in a microfluidic loop device. *Physical Review E, 89*(2), 023015.

Maddala, J., Wang, W. S., Vanapalli, S. A., & Rengaswamy, R. (2013). Traffic of pairs of drops in microfluidic ladder networks with fore-aft structural asymmetry. *Microfluid Nanofluid*, 377-344.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, 7: 308-313.

Nie, Z., Xu, S., Seo, M., & PC. Lewis, E. K. (2005). Polymer particles with various shapes and morphologies produced in continouous microfluidic reactors. *J Am Chemic Soc*, 127: 734-738.

Parker, R. G., & Rardin, R. L. (1988). *Discrete optimization*. Reading, Massachussetts: Academic Press.

Prakash, M., & Gershenfeld, N. (2007). Microfluidic bubble logic. *Science*, 832-835.

Quesada, I., & Grossmann, I. E. (1992). An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems. *Computers CHem. Eng.*, 16 (10/11), 937-947.

Ravagnani, M., Silva, A., Arroyo, P., & Constantino, A. (2005). Heat exchanger network synthesis and optimization using genetic algorithm. *Applied Thermal Engineering*, 25(27), 1003-1017.

Schindler, M., & Ajdari, A. (2008). Droplet traffic microfluidic networks: A simple model for understanding and designing. *Physical Review Letters*, 100.044501.

Sessoms, D., Belloul, M., Engl, W., Roche, M., Courbin, L., & Panizza, P. (2009). Droplet motion in mmicrofluidic networks: Hydrodynamic interactions and pressure-drop measurements. *Phys. Rev. E*, 1-10.

Sessoms, D. A., Amon, A., Courbin, L., & Panizza, P. (2010). Complex dynamics of droplet traffic in a bifurcating microfluidic channel: periodicity, multistability and selection rules. *Physical Review Letters*, 154501.

Shim, J., Cristobal, G., Link, D. R., Thorsen, T., Jia, Y., & Piatelli, K. (2007). control and measurement of the phase behavior of aqueous solutions using microfluidic. *Journal of the American Chemical Society*, 8825-8835.

Smith, B. J., & Gaver, D. P. (2010). Agent-based simulationsof complex droplet pattern formationin a two-branch microfluidic network. *Lab Chip*, 303-312.

Song, H., Chen, D., & Ismagilov, R. (2006). Reactions in droplets in microfluidic channels. *Angew Chem Int Ed*, 7336.

Wang, J., Sui, G., Mocharla, V. P., Lin, R. J., Phelps, M. E., Kolb, H. C., et al. (2006). Integrated microfluidics for parallel screening of an in situ click chemistry library. *Angewandte Chemie, 118*(32), 5402-5407.

Westerlund, T., & Petersson, F. (1995). A cutting plane method for solving convex MINLP problems. *Computers Chem. Eng.*, 19, 131-136.

Xu, S., Nie, Z., Seo, M., Lewis, P., Kumacheva, E., Stone, H., et al. (2005). Generation of monodisperse particles by using microfluidics: control over size, shape, and compositions. *Angew Chem Int Ed.*, 117:734-738.