Computer-Aided Spatially Optimized Design of Layer-Assembled Micro-Scale Mechanical Resistance Networks for 3D printing

By
Elishai Ezra Tsur

M.Sc. dissertation, submitted in partial fulfillment of the requirements for the M.Sc. degree, research track, School of Computer Science

The Interdisciplinary Center, Herzliya

January 2017

To my wife Tamara, my son Ben and my daughter Maya
for never having anything but confidence in me

This work was carried out under the supervision of Prof. Ariel Shamir at the Efi Arazi School of Computer Science, The Interdisciplinary Center, Herzliya.

Acknowledgments

First and foremost I would like to thank my adviser Prof. Ariel Shamir. Beyond the insightful discussions, Ariel had given me the freedom to pursue research on topics about which I was truly passionate.

I would like to convey special thanks to the staff of the Center for 3D functional printing at the Nanoscience and Nanotechnology facility at the Hebrew University. Especially to Dr. Michael Layani for his useful advice and support, which greatly supported this work.

I am eternally grateful to my parents Dalia and Yaakov, for their eternal support. Most importantly, I would like to thank my wife Tamara, for bearing with me during this quest of scientific enrichment.

Elishai Ezra Tsur

Abstract

Current design streamline of modern microfluidic circuits is not cost effective as it relies on extensive manual intervention. In this work, we exploit the hydraulic-electric circuit analogy to define abstract models of microfluidic circuits. Based on fabrication-related constraint propagation and optimization protocols, given an electrical model, our algorithm suggests a spatially optimized hydraulic design. Finally, we generate a vector-graphics model for 3D printing. Our approach can significantly reduce the design time of microfluidic circuits, allowing a seamless computer-aided transition from concept to production.

Contents

List of Figures

1 Background

1.1 Microfluidics

1.1.1 Applications to Control

Prof. George Whitesides, one of the founding fathers of the polymer-based microfluidics, suggested that the field of microfluidics has four parents: molecular analysis, biodefence, molecular biology and microelectronics [4]. Methods such as gas-phase chromatography (GPC), high-pressure liquid chromatography (HPLC) and capillary electrophoresis (CE), revolutionized chemical analysis by providing high content and high resolution measurements using small amounts of sample. Another important motivation for microfluidics developments was a series of grants by the Defense Advanced Research Projects Agency (DARPA) of the US Department of Defense, originated after the cold war, for the development of field-deployable biothreats monitoring microfluidic systems. The third parent of microfluidics is blossom of genomics in the 1980s which give rise to the need of sensitive and accurate analytical methods. The four parent of microfluidics is microelectronics, and specifically the utilization of photolithography technologes for the fabrication of silicon-based Micro-Electro-Mechanical-

Systems (MEMS). The first microfluidics devices were did, in fact, made out of silicon and glass, but they were not widely adopted by the research community due to their cost and poor optical properties. In 1993 Prof. Whitesides developed the first polymer-based microfluidic device, which was fabricate from Polydimethylsiloxane (PDMS), an optically transparent, soft elastomer. The ease with which new concepts can be tested in PDMS, and its ability to support the implementation of pneumatic valves (as will be descibed later), have made it the key material for exploratory microlfuidic-based research at the early stages of development [5].

Today, microfluidic technology delivers the potential for high-throughput and high-content studies in a wide spectrum of experimental sciences, from biology to chemistry and physics. It provides the ability to precisely control experiments at the microscale [6, 7, 8], and enables easy and fast automation using computer-controlled micro-mechanical valves [9, 10]. Indeed, the ever increasing desire to delve into biology at single entity level has made microfluidics remarkably relevant for the biological domain. By virtue of microfluidic's compatible length scale, compliant surface chemistry and the minuscule reagent volume usually required, microfluidics has opened up new research opportunities in the field of biological science, which even a decade ago, looked utterly unattainable [11]. Microfluidic platforms revolutionized traditional sample handling, detection and analysis for biological assays, with applications ranging from basic biological research, pharmaceuticals and medicine (such as drug design and diagnostic devices) to industrial applications of combinatorial synthesis (such as high throughput screening and rapid chemical analysis). Moreover, the ability to design fluidic structures that resemble biological organization with microscale precision has given rise to new research paradigms such as the 'Organ on a Chip' concept. The vast invasion of microfluidics into biological research and technologiesis is attested by its expanding applications in cell biology, protein and DNA biophysics, structural biology, biosensors and tissue engineering [11]. For example, Bhatia and colleagues used microfabricated stamps to gener-

ate a micropattern co-culture of Hepatocytes and 3T3 fibroblasts for the study of cell-cell interactions [6]; Ho and colleagues used an enhanced field-induced di-electrophoresis trap to pattern Hepatocytes and Endothelial cells in a radial pattern to mimic the lobular morphology of liver tissue [12]; and Quake and colleagues developed a microfluidic device consisting of thousands of microfabricated switches for a genomic analysis at single cell resolution [13].

The underlying quality of microfluidics to precisely control stimuli at a subcellular scale has paved the way for new approaches to single cell biology. The notion of integrating several time-consuming assay steps into a solitary monolithic micro-platform is of great importance as it is attested by the recent developments of multifunctional platforms for obtaining fundamental biophysical and biochemical insights on cells and tissues (such as cell-cell interaction and cellular differentiation), as well as for the development of cell-based sensors with biochemical, biomedical and environmental applications [11]. For example, a microfluidics-controlled long-term cell culture platform with a stable environment [14] was used to expose patterned cells to gradients of chemical concentrations. Typically, cells can be micropatterned using differential adhesion of ECM proteins (that promote cell adhesion), trap arrays [15] or dielectrophoretic traps [16]. Patterned cells can be spatially or temporally exposed to in-vivo like chemical stimuli.

1.1.2 Design Stages and Tools for Development

The flow chart presented in **Fig. 1.1** illustrates the design stages typically followed in the development process of a microfluidic-based system (inspired by [17]). The apparent iterative motive is inherently embedded within the design process, a fact that stands for the 'rapid prototyping' paradigm, which governs the practice of microfluidics design. Note that development costs and time increase at later design stages.

During the design process system specifications are iteratively compared with the system design using design rules, physical knowledge, fabrication pro-

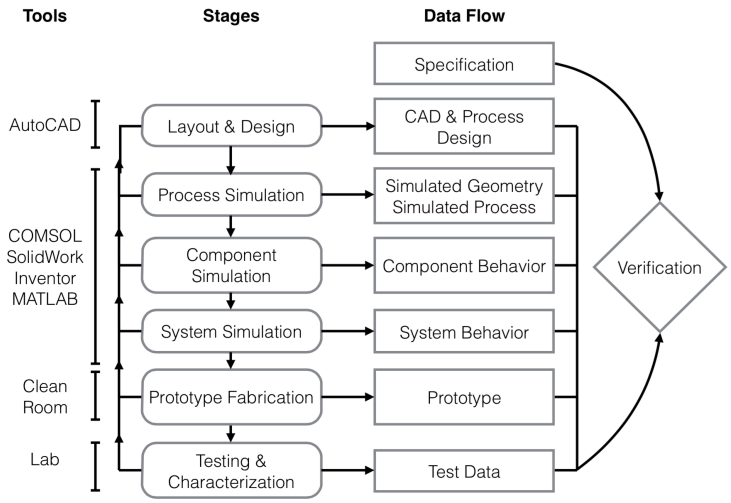

Figure 1.1: Design stages and tools for the development of a microfluidic-based system

cess, materials, numerical models and finally, the evaluation and testing of the assembled device. Fortunately, by means of computer-aided-design (CAD) the conceptual design can be quickly drawn and tested using numerical simulation packages or/combined with analytical models for optimization. The optimized design can be built as a prototype and experimentally tested. This design trajectory supports a process of " trial and error and enables rapid lean prototyping.

One important lesson which should be taken from the microfluidics design process is that the operation of the system has to support and follow the same 'prototyping notion' - it should be modular and it should support iterative work flow. In this work, we tried to support and contribute to this notion.

1.1.3 Lithogrphy-Based Fabrication

** Detailed protocols for the processes described in this section are given as appendices to this work.*

Microfabrication techniques have been developed in the microelectronics industry to create complex electronic circuits with minimum feature sizes currently as small as 7 nm [18]. Techniques for creating the insulating or conducting

features include physical vapor deposition, chemical vapor deposition, sputter coating, molecular beam epitaxy, and chemical beam epitaxy [19]. Integrated circuits are created using deposition of conductive areas in a process known as photolithography [20]. Photolithography allows the creation of complex geometric patterns with small feature sizes and it is widely utilized for microfluidics and lab-on-a-chip fabrication. Photolithography is commonly performed using a light sensitive material (photoresist) and a photomask. A positive photoresist becomes soluble where it has been exposed to light, while a negative photoresist becomes insoluble where it has been exposed to light. Following exposure, devices are typically heated and treated with solvents to remove the soluble photoresist and develop a cured device. Using conventional optical techniques, the minimum feature size that can be created is limited by the diffraction limit of light. For ultraviolet (UV) light ($248nm$) features as small at $70nm$ can be defined.

For microfluidic applications, a variation known as soft photolithography is often used. The hardened product of standard photolithography is used as a master and elastomeric poly(dimethyl-siloxane) (PDMS) is poured and cured over it. The channels' width is determined by the photo-mask and the channels' height is determined by the thickness of the patterned photoresist. PDMS is often chosen for chemical and biological experiments due to its low cost, high oxygen permeability, fast curing time, hydrophobicity, and low toxicity. PDMS forms an irreversible bond to glass using plasma activation, allowing its use with high magnification inverted microscope objectives. Since three-dimensional (3D) control of features is generally limited to time consuming layer-by-layer realignment [21], most microfluidic designs are limited to two-dimensional (2D) geometries. However, recent fabrication techniques such as 3D printing aim to change that [22, 23]. For example, Bhargava and colleagues showed that 3D printing can be used to rapidly fabricate discrete microfluidic elements that can be assembled into complex 3D circuits [24].

1.1.4 Bottlenecks in Microfluidic Design

The design streamline process of a microfluidic network starts with system specifications and ends with a topological graph which describes the networks 2-dimensional architecture [25], that can then be physically fabricated. Currently, most microfluidics designs are manually drawn using vector-graphics editors such as AutoCAD. Subsequently, system specifications need to be iteratively compared with the system design using design rules, physical knowledge, fabrication process, materials properties, and numerical models. Finally, the assembled device is evaluated and tested experimentally [26]. This design streamline relies on numerical analysis and experimental results, which are rarely suitable for parameter controlled model validation, it is often computationally intensive and not cost-effective [27]. Moreover, since the fabrication stage is usually based on soft-lithography, the design to production transition is not straightforward, requiring manual intervention and operation throughout the process. Aside from being long and inefficient, this design paradigm is not aimed toward the optimization of footprint nor the satisfaction degree of physical and fabricational constraints.

Numerical modeling and iterative design are the first bottleneck of the development of many microfluidic-based applications. They are computationally intensive and rarely allow for parameter-controlled validation. The second bottleneck of microfluidic design is a result of the current design paradigm of microfluidic circuits in which the circuits topology is manually defined with vector-graphic editors, leading to sub-optimized design and prolonged process of design-simulation cycle. The third bottleneck of microfluidics-based application is the fabrication process. Common fabrication practice of microfluidic circuits includes soft-lithography, which allows the creation of complex geometric patterns with small feature sizes. It is commonly performed using a light sensitive material (photoresist) and a photomask, which defines the geometry. The photoresist is spin-coated over a silicon wafer and then exposed to a light through the photomask. Following exposure, devices are typically heated and

treated with solvents to remove the soluble photoresist and develop a cured device. The hardened product of standard photolithography is used as a master and elastomeric poly(dimethyl-siloxane) (PDMS) is poured and cured over it. PDMS forms an irreversible bond to glass using plasma activation, allowing its use with high magnification inverted microscope objectives [28]. Since 3D features is generally limited to time consuming layer-by-layer realignment [21], most microfluidic designs are limited to two-dimensional (2D) geometries.

One attempt to relax the first bottleneck of microfluidic circuits design is to utilize the hydraulic-electric circuit analogy as an alternative strategy to numerical simulations for microfluidics modeling. The Hagen-Poiseuille equation that governs hydraulic behavior is considered equivalent to Ohms law, which describes electrical currents in a resistive conductor [1]. As will be described later, under the assumption that flow is , viscous and incompressible, as is usually the case for flow in micro-scale networks, voltage drop and electrical resistance can be treated as analog to pressure drop and hydraulic resistance, respectively. From the application of Kirchhoff laws for electrical circuits, assuming constant input pressure and a passive geometry, fluid velocity in each of the network's segments can be realized from the channels hydraulic resistance and total pressure drop. From the electric-hydraulic analogy, a useful abstract model of a microfluidic network arises: a network of resistors that defines a topological microfluidics design. The designer can use the electrical model to derive the flow rate in each segment of the network, potentially allowing for parameter-controlled optimization. An example of a geometric realization of an electrical model is shown in figure 1.2.

The second bottleneck of microfluidic circuits design is yet to be systematically addressed. This is even though constraints-satisfactory geometrical realizers have been under intensive research in the micro-electronics industry. For example, Valliant and colleagues developed an algorithm for the embedding of the interconnection pattern of a circuit into a 2-dimensional surface of minimal area for VLSI design [29]. Keisuke and colleagues developed a graph-embedding

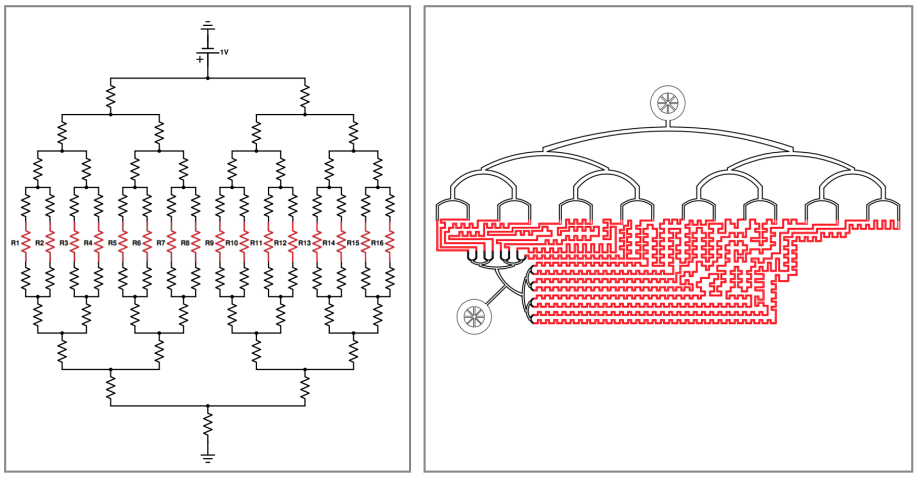

Figure 1.2: An example of a geometric realization of an electrical model.

algorithm for the reconstruction of a solid model from wireframe diagrams [30]. Even though these algorithms cannot be directly applied for hydraulic layouts, the same approach should also be applied for microfluidic circuits design.

Attempts to relax the third bottleneck of microfluidic circuits design include the introduction of 3D printing, which is destined to revolutionize the field [31, 32]. For example, Bhargava and colleagues showed that 3D printing can be used to rapidly fabricate discrete microfluidic elements that can be assembled into complex 3D circuits [33]. Recently, Dolomite has commercialized the first microfluidic 3D printer, named fluidic factory, allowing for direct fabrication of fluidically-sealed microfluidic devices for as little as $1 per chip, using Cyclic Olerin Copolymer (COC) as the printing material. However, current microfluidic printers require specialized and expensive equipment, preventing widespread adoption of the technology [34].

1.1.5 Hydraulic - Electric Analogy

Under the assumption that flow is laminar, viscous and incompressible, as is usually the case for flow in micro-scale networks, voltage drop and electrical resistance can be treated as analogous to pressure drop and hydraulic resistance,

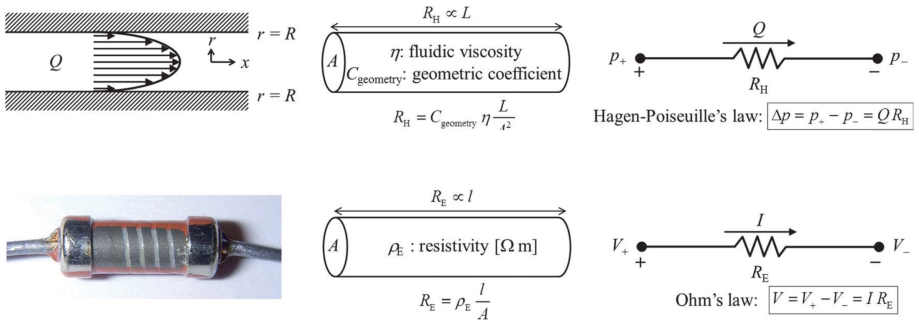

Figure 1.3: The physical similarities between the flow of a fluid and the flow of electricity [1]

respectively (figure 1.3).

From the application of Kirchhoff laws for electrical circuits, assuming constant input pressure and passive geometry, fluid velocity in each of the network's segments can be realized from the channels hydraulic resistance and total pressure drop. The Hagen-Poiseuille law can be expressed as:

$$Q = \frac{dp}{R_H} \tag{1.1}$$

where Q is the volumetric flow rate $[\frac{m^3}{sec}]$, dp is the pressure difference $[Pa]$ and R_H is the hydraulic resistance. The hydraulic resistance for rectangular channels - the most common geometry in microfluidic networks - is given by:

$$R_H = \frac{12\eta L}{wh^3\left(1 - \frac{h}{w}\left(\frac{192}{n^5}\sum_{1,3,5,\ldots}\frac{1}{n^5}tan\left[h\left(\frac{n\pi w}{2h}\right)\right]\right)\right)} \tag{1.2}$$

where η is the fluid viscosity $[Pa \cdot sec]$, and L, w, and h are the channels length, width and height $[m]$, respectively. Since N serially connected fluid resistors carry the same volumetric flow from one terminal to the other, the equivalent resistance will be equal to their sum:

$$R_T = R_{H,1} + R_{H,2} + ... + R_{H,N} \tag{1.3}$$

Similarly, parallel connected N fluid resistors have an equivalent resistance

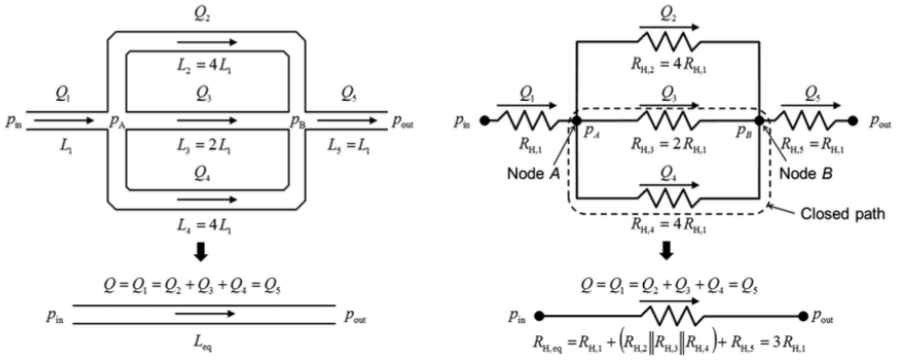

Figure 1.4: A simple microfluidic network and its equivalent electrical model [1]

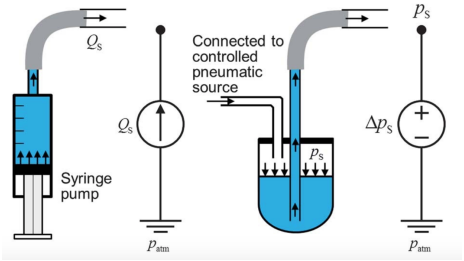

Figure 1.5: Electrical models of syringe and peristaltic pumps [1]

of:

$$\frac{1}{R_T} = \frac{1}{R_{H,1}} + \frac{1}{R_{H,2}} + ... + \frac{1}{R_{H,N}} \tag{1.4}$$

An example of an microfluidic - electrical model is given in figure 1.4.

Most pressure-driven microfluidic devices are operated with pressure sources or with syringe / peristaltic pumps. Electrical models of syringe and peristaltic pumps are given in figure 1.5.

A popular example for the illustration of the hydraulic-electric analogy is the microfluidic maze, which was proposed by Whitesides and colleagues in 2003 [2]. Pressure-driven flow in the networks could be used to identify global pathways of least fluid resistance. At each node, fluids follow flow division analogous to electric current division by taking the path of least resistance (figure 1.6).

Among the many direct applications for the hydraulic-electric analogy are

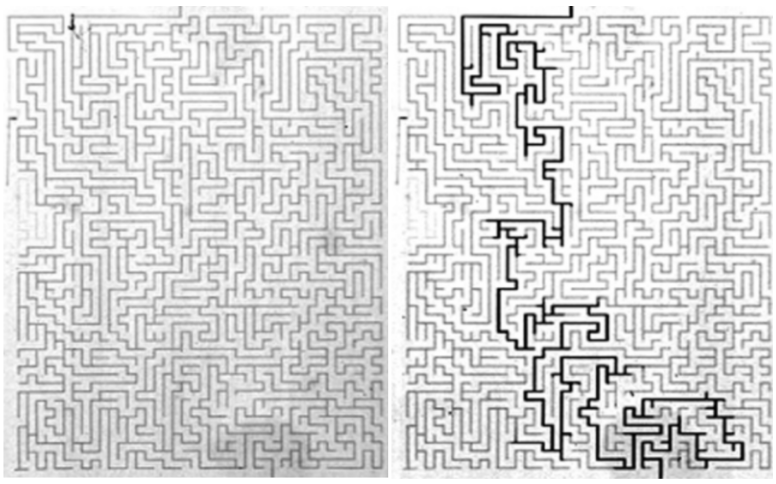

Figure 1.6: A one-solution maze filled with a "fluidic solution" [2]

dilution microfluidic networks, fluid mixing channels, microfluidic proportional / pyramidal / serial networks for gradient generation, combinatorial mixing, flow distribution , shear force generators, hydraulic filtering, and hydraulic tapping.

Dilutions and gradient generation are essential to many research area. Since Whitesides and colleagues proposed in 1999 to utilize microfluidic technology for gradient generation [35], various designs have emerged, showing promising potential as a powerful and versatile technology to accurately control tempo-spatial conditions [36]. Gradients of diffusible signaling molecules play important roles in numerous processes, both biological and chemical. For example, Walker and colleagues used microfluidic gradient generator to show that biochemical gradient signaling such as growth factors and hormones can result in directed cell migration and chemotaxis [37]. Lee and colleagues have shown a dendritic growth mediated by concentration gradients of growth factors, such as epidermal growth factor (EGF), fibroblast growth factor 2 (FGF2) and platelet-derived growth factor (PDGF) [38]. Predominantly, in biological developmental processes, the notion of morphogen gradient is intimately associated with paradigms on the determination of the cells location, differentiation and fate [39]. Microfluidic gradients formation is also in used extensively in liquid chromatography [40] and toxicity screening [41], where the influence of different

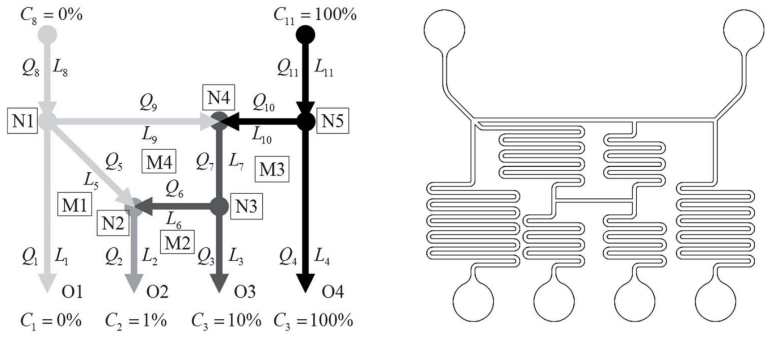

Figure 1.7: Illustration of a typical serial dilution microfluidic network generating a 10-fold log profile of a sample concentration [1]

drug concentrations on cell function is analyzed to derive dose-response curves [42]. Hence, there is a great interest in platforms which can maintain a variable, stable tempo-spatial gradients [43].

An illustration of a typical serial dilution microfluidic network generating a 10-fold log profile of a sample concentration is given in figure 1.7. A serial network of microchannels mimics a serial dilution comprising a series of stepwise dilutions to reduce the concentration of sample into a broad range. In conventional manual serial dilutions, a sample solution is serially diluted wherein the sample diluted from a previous step is processed sequentially through successive dilution steps. A 10-fold logarithmic gradient is obtained by the same 9 : 1 volumetric mixing ratio of the buffer and the sample diluted from the previous step in each step [1].

A pyramidal network suggested by the Whitesides and colleagues in 2001 [3], became one of the most popular ways of generating concentration gradients in microfluidic circuits. In a pyramidal network, fluid is to splited, combined, and mixed in a controlled manner (figure 1.8). While keeping the number of inlets low, the splitting in the pyramidal network increase the number of streams carrying different concentrations and being brought together into a broad channel. As a result, complex gradients can be approximated with step-gradients made of a large number of small steps [3].

The pyramidal architecture can generate more complicated gradient struc-

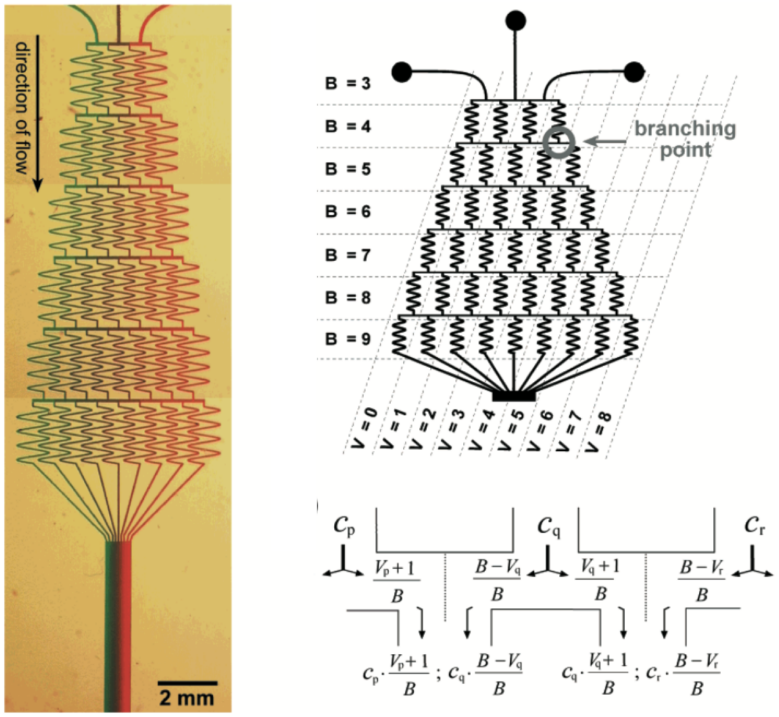

Figure 1.8: Gradient generation in microfluidic a pyramidal network [3]

tures. One example is the periodic gradients, which is shown in figure 1.9

Another interesting example is the use of microfluidic flow distribution to design an array of clamps for the immobilazation and imaging of *C.elegans* [44] (figure 1.10). The array is designed such that, on average, one worm is sorted into each clamp. This design enables researchers to immobilize over 100 worms in less than 15 minutes, allowing them to investigate physiology and behavior in large populations of c.elegans.

1.2 Optimization Algorithms

In general, optimization is the selection of the most suited element with regard to a cost function criterion, from some set of available alternatives. In the simplest case, an optimization problem consists of maximizing or minimizing a cost function by systematically choosing input values from within an allowed

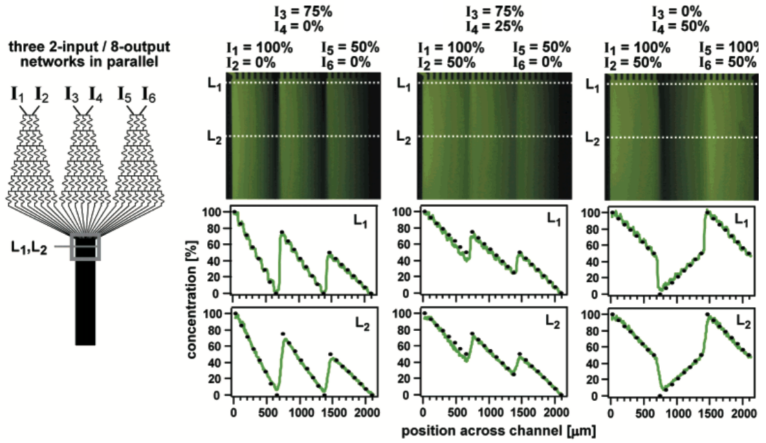

Figure 1.9: Periodic gradient generation in microfluidic a pyramidal network [3]

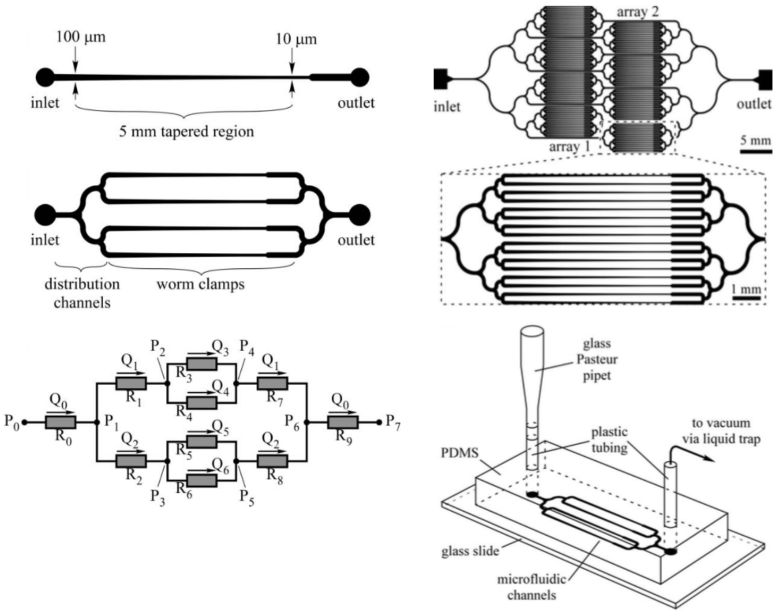

Figure 1.10: Design of a microfluidic array of worm clamps [3]

set and computing the value of the function.

In this work, we utilized four optimization algorithms: hill climbing, iterative local search, simulated annealing, and beam search. *Hill climbing* search is a loop that continually moves in the direction of increasing value - that is, uphill. It terminates when it reaches a peak where no neighbor has a higher value. Hill climbing does not look ahead beyond the immediate neighbors of the current state. A pseudo code for hill-climbing search is shown in 1.2.1.

Algorithm 1.2.1: HILL CLIMBING(*Problem*)

```
current ← MAKE-NODE(problem.Initial − State)
repeat
  { neighbor ← a highest-valued successor of current
    if neighbor.VALUE < current.VALUE
      then return (current.STATE)
    current ← neighbor
until
```

Unfortunately, hill climbing often gets stuck in local maxima, ridges, and plateaux (figure 1.11, A).

A variant of hill climbing is *iterative local search* (figure 1.11, B), which conducts a series of hill-climbing searches from randomly generated initial states. A pseudo code for iterative local search is shown in 1.2.2.

Algorithm 1.2.2: ITERATIVE LOCAL SEARCH(*Problem*)

```
localmaximum ← HillClimbing(problem)
repeat
  { candidate ← pertubated local minimum
    local maximum candidate ← HILLCLIMBING(problem))
    if local maximum candidate < local maximum
      then return to the candidate of local maximum
      else continue with the new local maximum candidate
until iteration bound is reached
```

Another variant is *Beam search*, which explores a graph by expanding the most fitted candidate in a limited set (beam diameter). Beam search keeps track of k states rather than just one. A pseudo code for beam search is shown in 1.2.3.

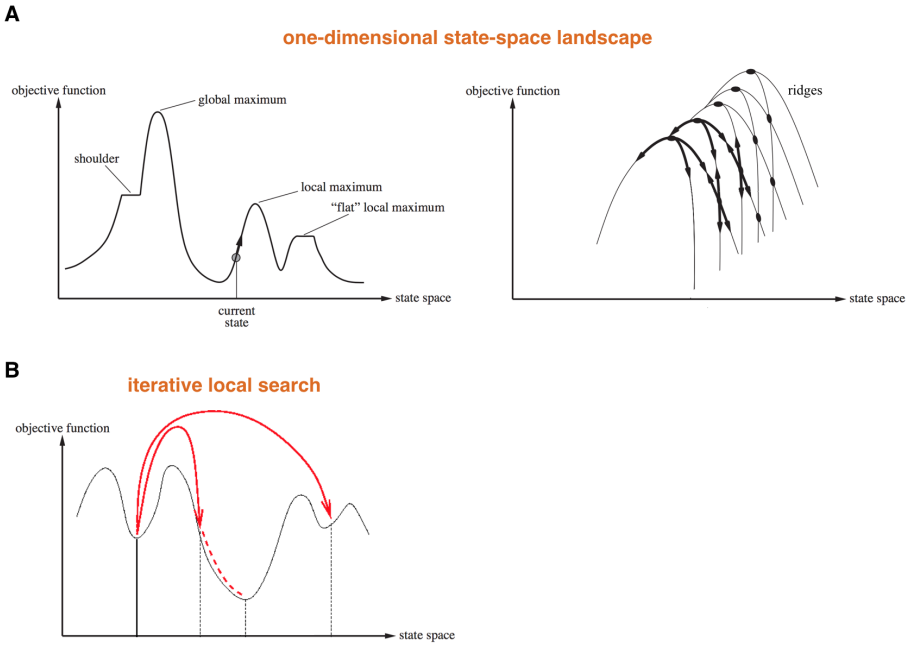

Figure 1.11: A one-dimensional state-space landscape in which elevation corresponds to the objective function

Algorithm 1.2.3: BEAM SEARCH(*Problem*)

localmaximum[*k*] ← HillClimbing(problem)
repeat
  space ← pertubate local maximum *i*, 'beam diameter' *d* times
  for each *candidate* ∈ *space*
    do candidate solution ← HILLCLIMBING(*problem*))
  localmaximum[*k*] ← *k* most fitted candidates
until iteration bound is reached

Hill-climbing algorithms that never makes down-hill moves toward states with lower value (or higher cost) is guaranteed to be incomplete. In contrast, a purely random walk - that is moving to a successor chosen uniformly at random from the set of successors is complete but inefficient. Therefore, it seems reasonable to try to combine hill climbing with a random walk in some way that yields both efficiency and completeness. *Simulated annealing* is such an algorithm. The innermost loop of the simulated-annealing algorithm is quite like hill climbing. Instead of picking the best move, however, it picks a random

move. If the move improves the situation, it is always accepted. Otherwise, the algorithm accepts the move with some probability. The probability decreases exponentially with the badness of the move, i.e. the amount of dF by which the evaluation is worsened. The probability also decreases as the temperature T goes down: bad moves are more likely to be allowed at the start when T is high, and they become more unlikely as T decreases. A pseudocode for simulated-annealing is shown in 1.2.4.

Algorithm 1.2.4: SIMULATED ANNEALING($Problem, Schedule$)

comment: Schedule maps time to 'Temperature'
$current \leftarrow$ MAKE-NODE($problem.Initial - State$)
for $t \leftarrow 1$ **to** ∞
do
- $T \leftarrow$ SCHEDULE(t)
- **if** $T = 0$
  - **then return** (c)$current$
- $next \leftarrow$ a randomly selected successor of $current$
- $dF \leftarrow next$.Value - $current$.Value
- **if** $dF > 0$
  - **then** $current \leftarrow next$
  - **else** $current \leftarrow next$with probability $e^{\frac{dF}{T}}$

Optimization algorithms are widely used in various fields of computer science and engineering. Specifically, using optimization for circuits layout has always been an important concept in the electronic industry, as will be described below.

1.3 Physical layout of printed electrical circuit boards [54]

Printed circuit boards (PCBs) are the backbones of almost every electronic device, and therefore, their design and manufacturing are vital to industrial production processes. Tasks in PCB design include logical design and physical layout. During logic design, components and interconnections are assigned and in the physical layout phase, the geometric positions of the components and their physical connections are decided. The final step is the industrial production.

During physical layout, the two major issues here are *component placement* and *wire routing*. In this work, the same major issues holds.

A PCB is built as a stack of layer pairs. Each pair starts as an insulating sheet with copper deposited on one or both sides. The copper sides of the sheet are first etched with different wiring patterns. Then, layer connecting holes (vias) with a conductive path between two or more layers are drilled. electrical components are usually attached to the board by pins soldering into a via pattern on the board. An example of components and connection layers in PCBs is shown in figure 1.12. The first task of the physical design is to decide on the size of the board and on the number of layers.

Most of the articles dealing explicitly with PCB design describe a PCB design issue and a heuristic for its solution. The utilization of heuristics is not surprising, as the problems encountered in physical PCB layout are usually NP-hard in theory and computationally expensive in practice.

Many algorithms and models for the design of PCBs have been proposed in the past two decays. For example, physical constraints on component placement in PCBs include repeated trials of random placements [45], cluster development [46], knowledge based systems [47], local search algorithms [48, 49], as well as combinations of these approaches [50].

1.4 Backtracked and Filtered Clustering

Another important aspect in this work is clustering. Specifically, we utilized backtracking search and forward filtering. Briefly, given a graph $G = (V, E)$ in which nodes $V = (v_1, v_2, ..., v_n)$ represent entities and edges $E = (< v_1, v_2 > , ..., < v_{k-1}, v_k >)$ represent constraints between entities (figure 1.13, A), the clustering algorithm should group nodes into minimum number of clusters $n_1, ..., n_c$, such that the number of constraints in each group is minimal. Therefore, similarly to the graph coloring problem, connected nodes should be assigned to a different layer. The challenge of realizing this design can be referred to as a

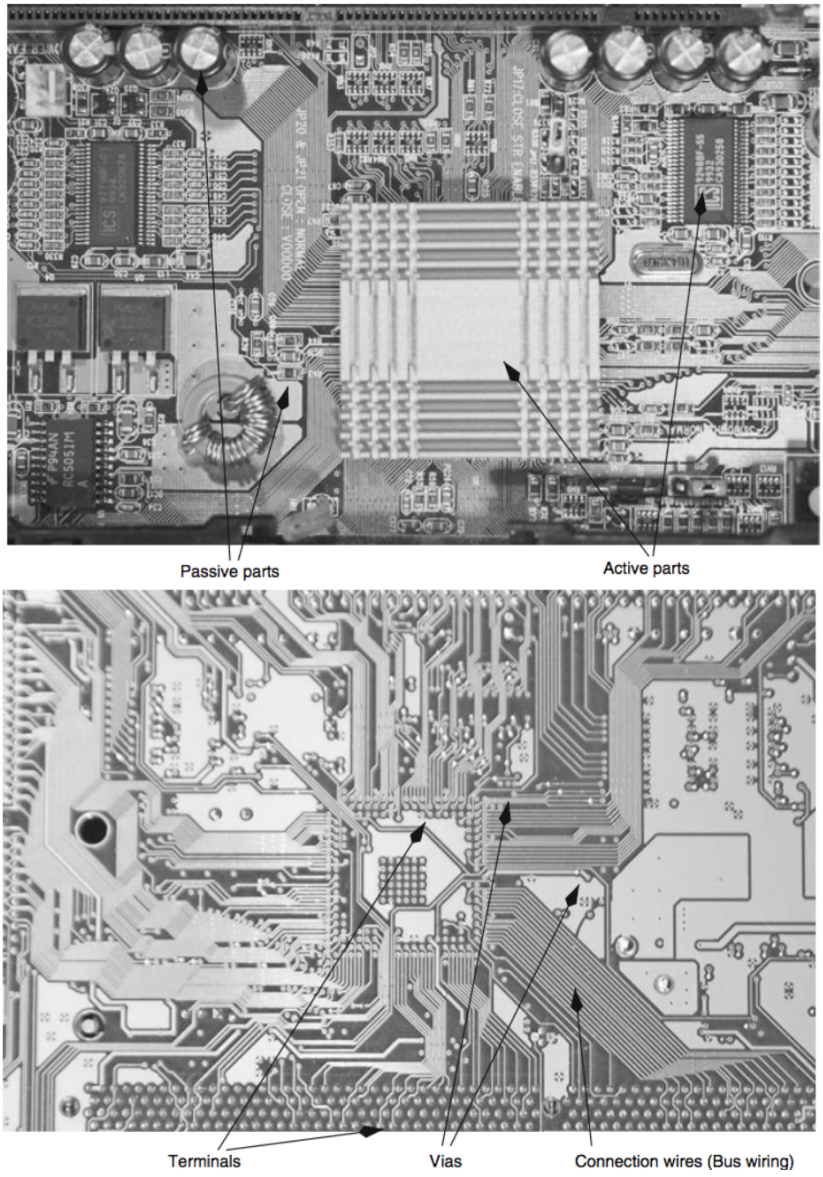

Figure 1.12: Components and connection layers in PCB's [54]

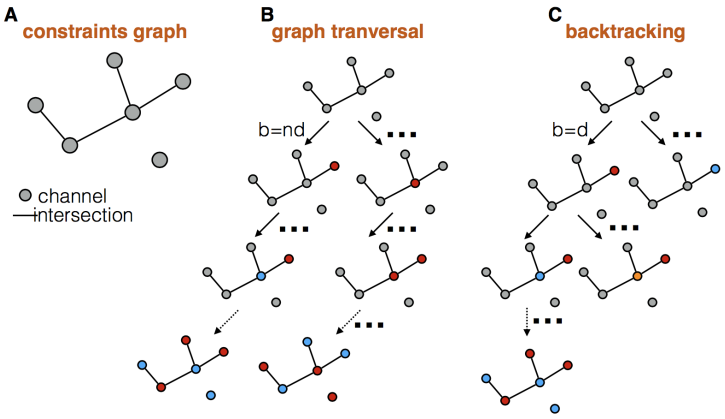

Figure 1.13: Clustering using backtracking and forward filtering

Constraint Satisfaction Problem (CSP). Building a search tree and using uniform search algorithms such as Depth First Search (DFS) or Breath First Search (BFS) results in a commutative search space with $n! \cdot d$ possible assignments, where n is the number of channels and d is the number of layers (both equals to the number of nodes in the worst case of full connectivity where all channels intersect and each channel is designated to a different layer). This searching approach is commutative since the order of assignments does not matter. However, by considering a single variable at each node during traversal, search can be optimized. This approach is often referred to in the literature as a backtracking search (figure 1.13, B-C).

A backtracking search improve DFS in two ways: (1) Fixed ordering - a layer is assigned to a single channel at each step (assigning channel x to layer w and channel y to layer z, is equivalent to assigning channel x to layer z and channel y to layer w); (2) Forward filtering - check constraints as you go through assignments - this approach considers only values, which do not conflict with previous assignments (incremental goal test). CSPs can be relaxed by the elimination of large portions of the search space at once by identifying variable/value combinations that violate the constraints [51]. We note that while forward checking keep arcs consistent (for each pair of nodes, any consistent assignment to one can be extended to the other), there are other algorithms

that maintain higher degrees of consistency by considering longer consistency paths and by propagating constraints when assignments are made. Here, since in most microfluidic designs, the number of channels is small (tens to hundreds), we consider backtracking and forward filtering to be sufficient.

2 Method

"The quality of connections is the key to quality per se." Charles Eames

Here we present an algorithm which, based on fabrication-related constraint propagation and optimization protocol, suggests a design for the proposed model with minimal footprint. Algorithm schematic is illustrated in figure 2.1.

Our algorithm accepts an electrical model as an input parameter, and produces 3D design, suitable for 3D printing. Our algorithmic approach is divided into 7 components: (1) construction of snakes geometries, one for each resistor in the electrical model, in correspondence to physical and fabrication constraints; (2) discretization of the hydraulic layout to a rectangular grid, in correspondence to the dimensions of the channels, as they were defined in (1); (3) random assignment of resistors to the grid; (4) layout optimization by the rearrangement of channels locations; (5) channels clustering into non-intersecting groups, where each group defines a new connection layer; (6) generation of a layered layout, where the bottom layer consists the hydraulic channels and the upper layers defined by the connection layouts; (7) generation of the design in vector graphics.

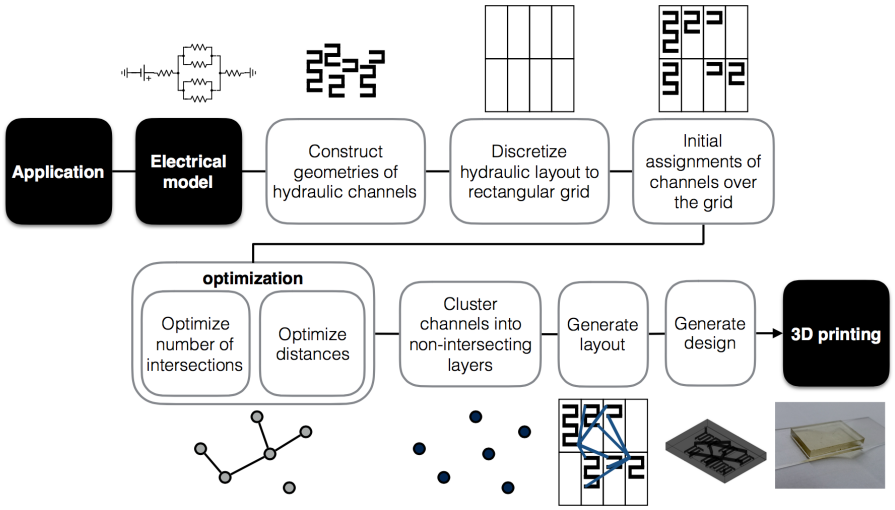

Figure 2.1: Algorithm schematic

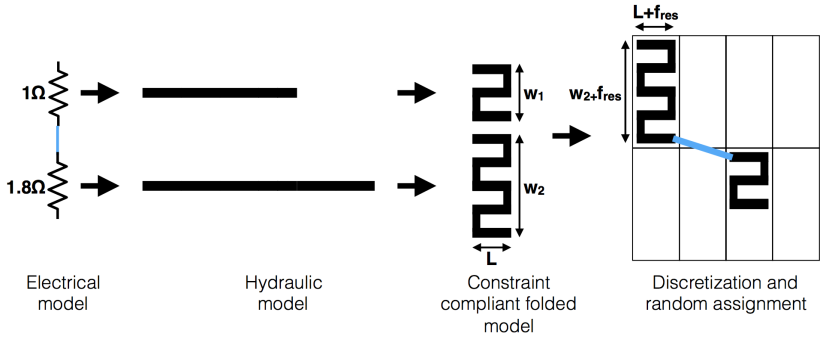

Figure 2.2: An illustration of the discretization and initial assignment process

2.1 Discretization

The discretization process is illustrated in figure 2.2.

For microfluidic networks, the computerized geometrical realizer should follow several design rules: (1) fabrication resolution; (2) aspect ratio of structures; and (3) distance between networks inlets / outlets. When using 3D printing, the fabrication resolution is largely depending on the printing medium and printer. Flow structure dimensions are generally limited to an aspect ratio of 1/10 (height/width). Structures with lower aspect rations are prone to col-

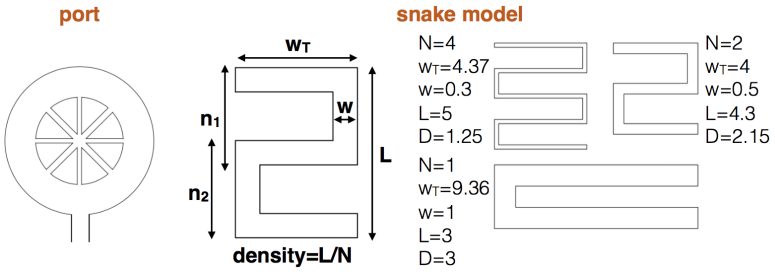

Figure 2.3: A visualization of snake-based channel geometry

lapse. If a feature is wider than this design rule permits, support posts must be added. Standard inlet/output settings should support standard channels punchers, which are usually characterized with a punch holes of 20 gauge (approximately 0.66mm in diameter). Distance between ports should be at least $2mm$ [52].

Each of the networks channels, which was predefined in the electrical model, can be geometrically realized using endless possibilities of shapes. Here we chose a parameterized geometry of an alternate snake. The snake geometry is gained by iteratively folding a channel in a constant interval until all design rules are met. We have characterized the snake model with 5 parameters: w for the channel width, w_T for the snake width, L for the snake length, N for the number of snake's bends and D for density $\frac{L}{N}$ (figure 2.3).

To calculate the transversal length of the perfused fluid through the snake geometry, the principles of fluid dynamics at the microscale should be considered. A fluid in motion comes to a complete stop at the surface due to viscous effects, defining a no-slip boundary. The layer that sticks to the surface slows the adjacent fluid layer (again, due to viscous forces), which slows the next fluid layer, and so on, causing the development of a velocity profile that defines a boundary layer. This velocity profile results in the fact that most of the fluid is flowing through the middle of the channel (where the velocity is maximized) [53]. Considering a parabolic profile of flow, the fluid transverse length T in the

snake can be calculated with:

$$T = (N + 1) \cdot w + L - 3w \tag{2.1}$$

The solution space of any given design is enormous. For example, a circuit layout with only 12 channels can be spatially organized in a minimum $4.8 \cdot 10^8$ different arrangements. It is therefore clear that an optimization process is compulsory. Our approach would be to start with a base line solution and then iteratively change and improve it following an optimization protocol. Our base line solution will draw the mechanical layout bottom to top, defining the channels in a basic snake formation on a predefined grid, and laying on top of it connecting layers that preserve the hydraulic resistance ratio between the connected ports.

Our suggestion for a base line solution is to divide the canvas to a 2D grid of rectangles, which will be defined using the properties of the geometries being generated and the specified constraints. For example, from the fabrication resolution, the width w of the channel is derived. From the aspect ratio constraint, the allowable snake width w_T is derived. The base line solution will prolong and bend the snake (in intervals of $N + 2$) until the specified resistance (or length) is satisfied, setting the snake's length L. The rectangle's dimensions that block the snake are $(L + f_{res}) \cdot (w + f_{res})$, where f_{res} is the fabrication resolution. To allow trivial displacements of the channels over the grid, the base line solution can create an homogeneous grid, where the width and length of the rectangle is derived from the longest channel in the model.

After the grid is defined, the base line solution randomly assign the channels to the grid. Following channels assignment, the connection between them is derived from user specifications, and connection layers are generated and aligned on top of the flow layer. Non-intersecting channels are clustered into layers as will be described later.

Each connection between two snakes is defined by a straight line between the

source and target (figure 2.2). These connections will be fabricated in a separate layer, forming bridges between the snakes. The main problem is that these bridges might intersect each other, and therefore must be placed in different layers. This means that the baseline solution might be realized with many layers, which is less effective and more costly to fabricate. For this reason we turn this challenge into an optimization problem, which has two steps: (1) placement optimization, which assign the snakes to the grid so that a minimum number of intersections between connections are formed; (2) among the solutions with a minimal number of layers, choose the one in which the total length of the connecting channels is minimal. Once an optimized layout is proposed, our algorithm clusters the connections into a minimal number of groups so that each cluster does not have intersecting channels. Each cluster is then realized as a layer during fabrication.

2.2 Optimization

The discretization process is illustrated in figure 2.4.

Our optimization approach aim at reducing the number of layers by changing locations of the flow segments in the flow layer and then clustering the non-intersecting connecting channels into minimal number of groups. Snakes are therefore displaced and shift locations following an optimization protocol that aims at minimizing a cost function. This problem is often referred to in the literature as quadratic assignment [54].

Here we propose a two-step cost estimation. The parameter for the first cost estimation f_a is the number of intersections m of all connecting channels: $f_a = m$. Since connecting bridges cannot intersect in the same layer, we minimize the number of layers by minimizing the number of intersections. Next, among the suggestions that our algorithm finds to have a minimal number of intersections, we are optimizing the length of the connecting channels. We define the second cost estimation f_b as: $f_b = \sum_{i=1}^{k} l_i$, where k is the number of

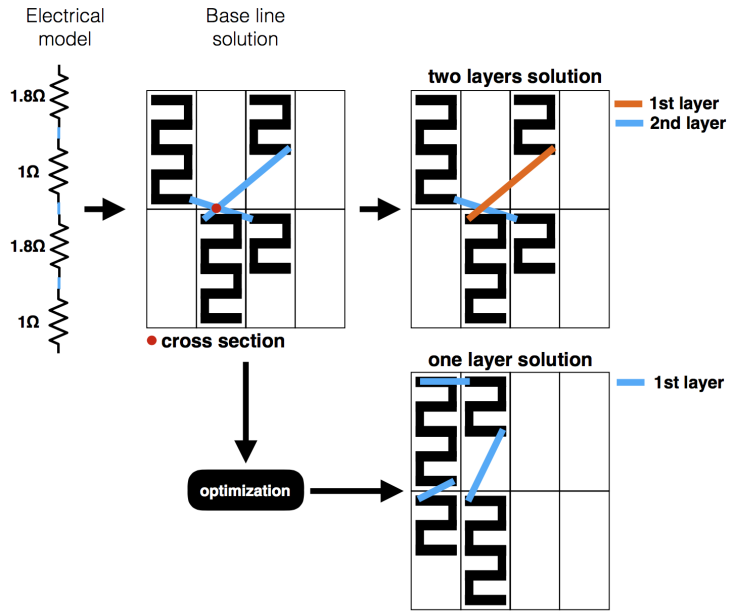

Figure 2.4: An illustration of the optimization process

connecting channels, and l_i is the length of connecting channel l. This is aimed at choosing a design in which closely connecting channels are packed together. For the optimization of both cost functions we chose to apply four different protocols: hill climbing, iterative local search, simulated annealing, and beam search [51] as they were described in section 1.2. In each step of an optimization cycle, resistors' locations are changed and the number of intersections, and energy is recalculated. Intersections were calculated using:

$$x = \frac{((p_1x \cdot p_2y - p_1y \cdot p_2x) \cdot (p_3x - p_4x)) - ((p_3x \cdot p_4y - p_3y \cdot p_4x) \cdot (p_1x - p_2x))}{((p_1x - p_2x) \cdot (p_3y - p_4y) - (p_1y - p_2y) \cdot (p_3x - p_4x))}$$
$$y = \frac{((p_1x \cdot p_2y - p_1y \cdot p_2x) \cdot (p_3y - p_4y)) - ((p_3x \cdot p_4y - p_3y \cdot p_4x) \cdot (p_1y - p_2y))}{((p_1x - p_2x) \cdot (p_3y - p_4y) - (p_1y - p_2y) \cdot (p_3x - p_4x))}$$

where p_1 is the point that define one end of the connecting channel (exit point of a channel), and p_2 is the second end of the connecting channel (entry point of a second channel). Together they define a connecting line between two channels. Similarly, p_3 and p_4 define another connecting line. x, y are the

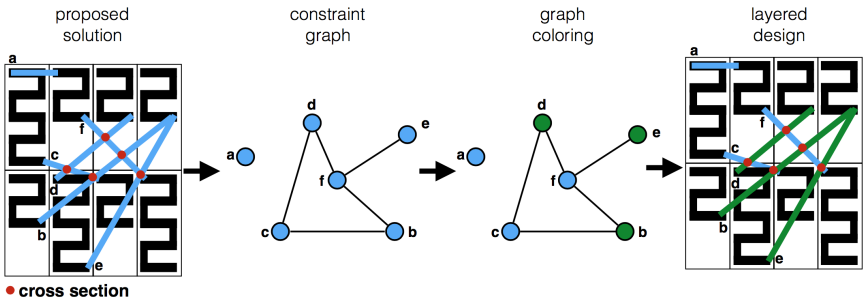

Figure 2.5: An illustration of the clustering process

computed coordinates of the intersection point of these two lines. Following the calculation of the intersection point, it remains to be checked if the intersection point is bounded between p_1, p_2, p_3, p_4. This is accomplished by comparing end coordinates. Length is calculated using the fluid transverse length in the snake, as it was defined in equation 2.1.

2.3 Clustering

Illustration of the clustering process is given in figure 2.5.

Once we find a solution that minimizes the number of intersections, we cluster the connecting bridges into layers in which no intersections exists. This problem can be reduced to the graph coloring problem by constructing a virtual graph $G = (V, E)$. Each connection, or bridge between snakes, defines a node in V. Each crossing between two bridges defines an edge in E. A colouring in G, induces a clustering of our original set of bridges to groups, where each group does not contain intersection. This is because if two nodes have the same color ithey do not have a connecting edge in E, and therefore their bridges do not contain an intersection. Finding the minimum number of colors is known as the graph coloring problem and is NP-complete[55].

The challenge of realizing this design can be referred to as a Constraint Satisfaction Problem (CSP) and it can be efficiently addressed using backtracking search and forward filtering. CSPs can be relaxed by the elimination of large

portions of the search space at once by identifying variable/value combinations that violate the constraints [51]. We used fixed ordering and constraints propagation with each assignment of a node to a layer. Our algorithm considers only values, which do not conflict with previous assignments (incremental goal test).

Specifically, we utilized forward filtering and backtracking as they were described in section 1.4. First, we assign a random color c to one of the nodes t in the constraint graph G. Afterwards, we perform forward filtering, going through all nodes, which are connected to t, adding c to their forbbiden colors assignment list. Once a color is assigned to a node, it is added to a colors priority list. When moving to another node for color assignment, first, colors in the priority colors list are considered. If all prioriterized colors are forbiden for this node, a new color is assigned (and hence, a new layer is introduced to the design). We note, that in this version of the algorithm, once a prioriterized color was not forbidden for an assignment, it is assigned to the node, without checking assignments of other prioreterized colors. The implications of alternative possible assignments are not considered, and therefore this proposed greedy approach is not optimal. However, for the applicativity of the algorithm to microfluidic circuits, which are very sparsed (channels are rarely fully connected), this algorithm produces good results, as they will be presented later. The algorithm terminates when graph tranversal is completed and all nodes are assigned to a layer. In our implementation, graph tranversal was made using Breadth First Search (BFS) [55].

2.4 Fabrication

Once the layered design of the network is concluded, it is converted to a vector-graphic drawing and then translated to an AutoCAD script, which can be invoked directly from program's interface. In this work, we have created a framework for automatic generation of AutoCAD script from a predefined channels' layout. The script is introduced to AutoCAD to create an STL (STereo-

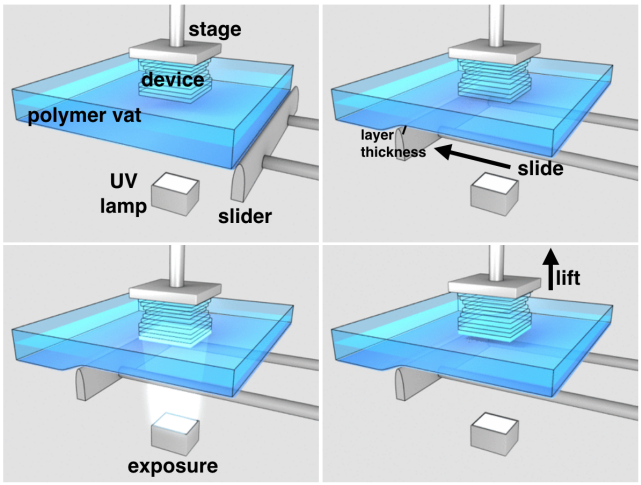

Figure 2.6: Slide and seperate 3D printing

Lithography) file, which describes 3D geometrical surfaces without color or texture attributes. STL can then be sent for 3D printing.

We can use the generated 3D computer model to fabricate two types of devices. The first one is the actual device and the second is a mold. The mold can be used to generate a device by polymer casting, curing, mold degradation. Briefly, devices are printed using Asiga Pico 2 (Asiga, Australia) with a commercial resin (plasclear), while using a glass slide as the build platform. Printing technology is termed Slide-And-Separate (SAS), and based on upside-down stereolithography system. Stereolithography is an additive manufacturing process that works by focusing an ultraviolet (UV) laser on to a vat of photopolymer resin. With the help of computer aided manufacturing or computer aided design software (CAM/CAD), the UV laser is used to draw a pre-programmed design or shape on to the surface of the photopolymer vat. The Asiga has an up side down elevator apparatus. The elevator platform ascends a distance equal to the thickness of a single layer of the design in out case, it was set to $50um$) into the photopolymer vat. Then, a resin-filled blade sweeps across a cross section of the layer, re-coating it with fresh material. The subsequent layer is traced, joining the previous layer. Illustration is shown in figure 2.6.

3

Software

Program architecture is composed of 7 packages, each responsible for a different aspect of the program. Simplified package, class diagrams and descriptions are given below. Package view of the software architecture in shown in figure 3.1.

Class diagrams of the frame, fundamentals and geometry packages are shown in figure 3.2. The fundamental entities being used by the different entities in the framework, such as the enumerated dimension and direction, are defined in the fundamental package. The fundamental class 'physical element' is abstractly defining fabricated entities. Both channel segment and snake (which is defined as a list of channel segments) extends the physical element class. Visualization of the graph layout is dealt within the frame package, using Java's swing library.

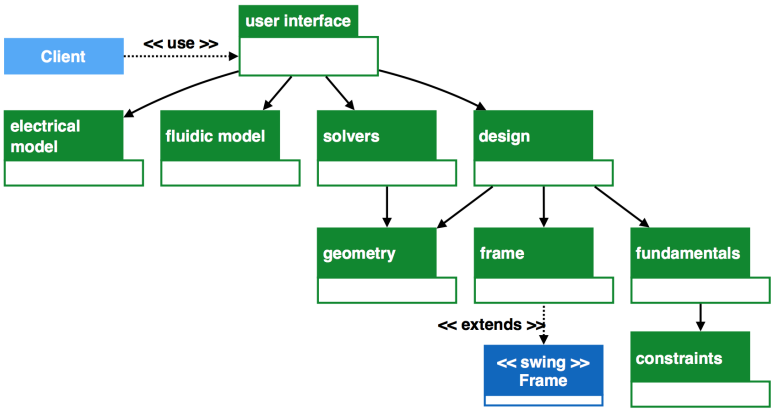

Figure 3.1: Package view of the software architecture

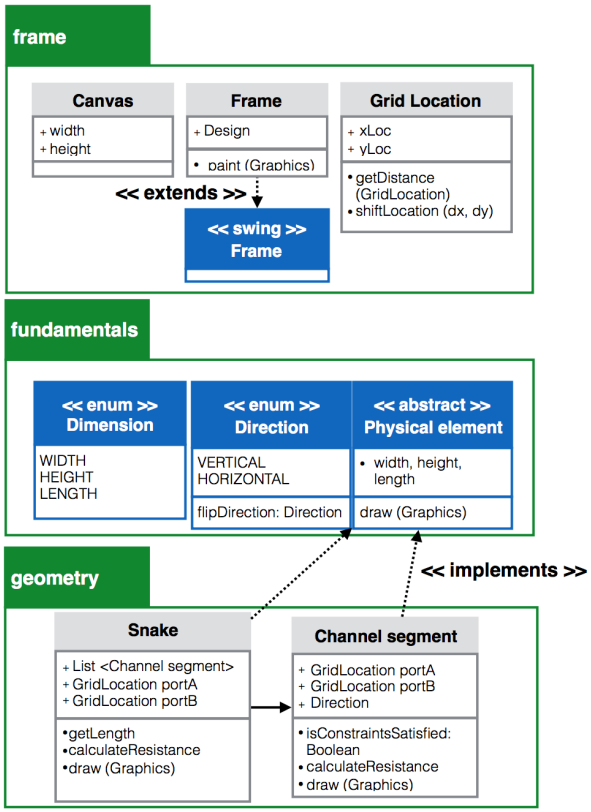

Figure 3.2: Class diagrams of the frame, fundamentals and geometry packages

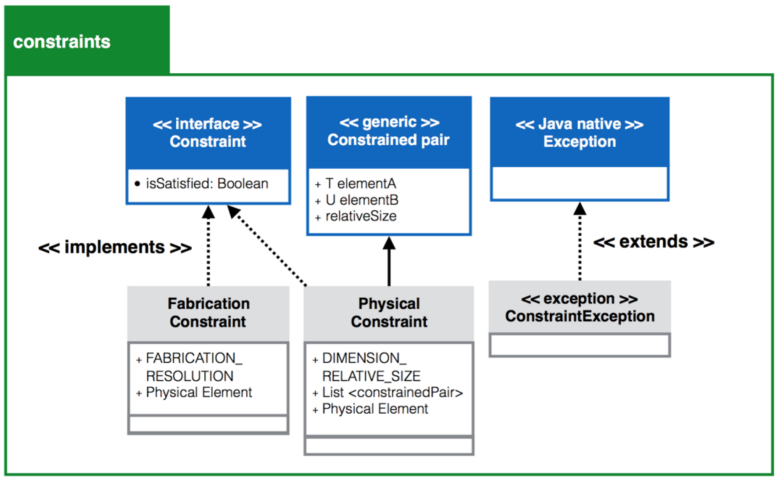

Figure 3.3: Class diagram of the constraints package

Class diagram of the constraints package is shown in figure 3.3. The constraints package is responsible for the constraints as they were specified in the text. These were divided into two main categories: fabricational and physical constraints. Both fabricational and physical constraints classes implement the constraints interface. The physical constraints class defines a series of dependencies between the generated dimensions of geometries.

Class diagrams of the design, solvers, hydraulic and electric circuits packages are shown in figure 4.4. The main responsibility of our framework is to convert an abstract description of an electrical circuit to a real design of a hydraulic design. Any solver has to implement this functionality - a responsibility, which is defined in the solver interface. Electrical circuit is described as a series of connectors, each connected to a resistor which can in turn be connected to a series (list) of resistors. The design class holds the canvas (the 2D grid on which the hydraulic circuit is defined) and the geometries with their location on the grid.

To illustrate the usage of the software, the main functional unit is given below:

```
1 public class Main {
2   final static String SCENE_FILE_PATH = System.getProperty("user.
```

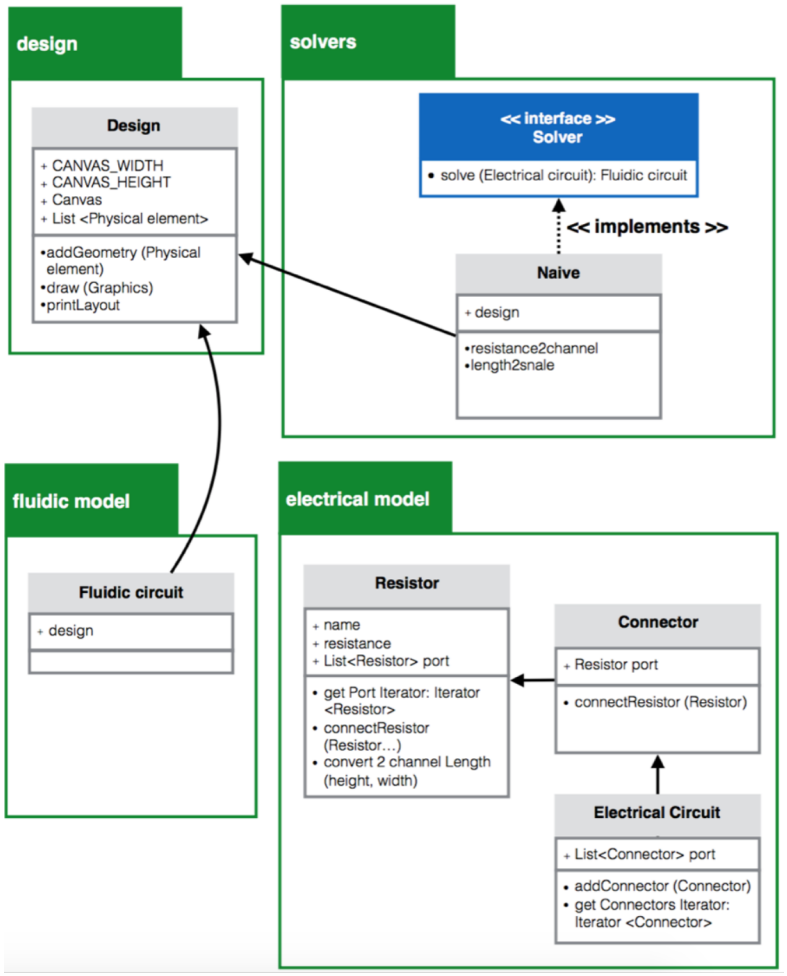

Figure 3.4: Class diagrams of the design, solvers, hydraulic and electric circuits packages

```
    dir");
  public static void main(String[] args) {
    // Get electrical model from parsed XML
    String layoutFileName = "/src/layouts/" + "layout5" + ".xml";
    File sceneFile = new File (SCENE_FILE_PATH + layoutFileName);
    String layoutXMLDesc = loadLayoutFromFile(sceneFile);
    LayoutXMLParser parser = new LayoutXMLParser();
    ElectricalCircuit circuit = parser.parse(layoutXMLDesc);
    // Solving the electrical model
    Solver CL = new Naive(circuit);
    FluidicCircuit FC = CL.solve();
    FC.generatePanels();
    // Cretaing an AutoCAD script of the design
    AutoCAD_generation CAD = new AutoCAD_generation(FC);
    CAD.generate();
  }
```

A graphical display was also developed, to allow the designer to view the resulted design without a CAD environment (figure 3.5). The display features a tabbed frame, in which each tab represents a layer, where the last tab displays the layers together (connections and channels were colored differently).

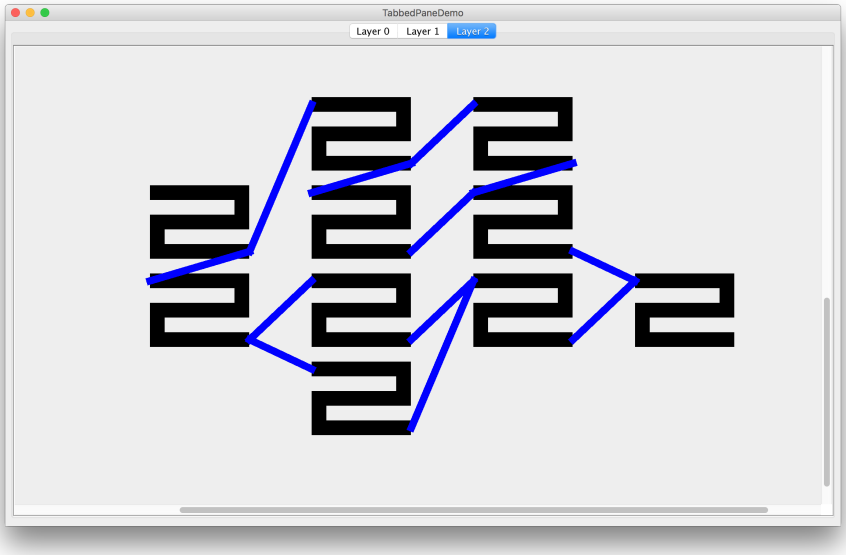

Figure 3.5: Graphical display

4

Results

4.1 Optimization

To demonstrate the capabilities of our approach we have created 5 microfluidic layouts, with 6-64 channels and 8-150 connections (using XML files). We have implemented a base line solver and four optimization algorithms: beam search, hill climbing, iterative local search and simulated annealing; as well as a filtered backtracking algorithm for layer clustering. Functionality was encapsulated in a software package that can parse an XML file and produces three outputs: an AutoCAD script file that consists of the proposed design, a log report, which describe the convergence of the algorithm toward the proposed solution and a

user interface, in which the user can observe a schematic of the proposed design (no need for external software). Visualization of the electrical description of the first and simplest layout, the proposed design and the printed device are shown in figure 2.1.

In our software package, the electrical model is described using an XML, which is parsed and build into the framework. XML description of the first and simplest electrical model is given below:

```
<layout resFactor="10 17">
    <R1 name="A" resistance="500" connectedTo="B C D E" />
    <R2 name="B" resistance="300" connectedTo="F" />
    <R3 name="C" resistance="300" connectedTo="F" />
    <R4 name="D" resistance="300" connectedTo="F" />
    <R5 name="E" resistance="300" connectedTo="F" />
    <R6 name="F" resistance="500" connectedTo="" />
    <portA name="conA" connectedTo="A" />
</layout>
\label{xml:xmlModel}
```

A schematics of the proposed designs for an electrical layout with 64 resistors, varying resistance values and 150 connections is shown in figure 4.1. The base line not optimized solution, produces a design with an energy of 62,579 (total length, arbitrary units) and 755 intersection points. This solution was clustered to a final design with 15 layers (figure 4.1, A). Time to solution was almost immediate. The Hill Climbing algorithm, with randomly chosen initial conditions, and an iteration bound of 2,500, produced a design with an energy of 27178 and 76 intersections. This solution was clustered to a final design with 5 layers (figure 4.1, B). The Beam Search (BS) algorithm, with a beam diameter of 10 and a search bound of 2500, produced a design with an energy of 27,576 and 64 intersections. This solution was clustered to a final design with 4 layers (figure 4.1, C). Iterative Local Search (ITL), with 200 local iterations of hill climbing searches (each conducts 2,500 iterations), produced a design with an energy of 23996 and 21 intersections. This solution was clustered to a final design with 4 layers (figure 4.1, D). Simulated Annealing (SA), with an iter-

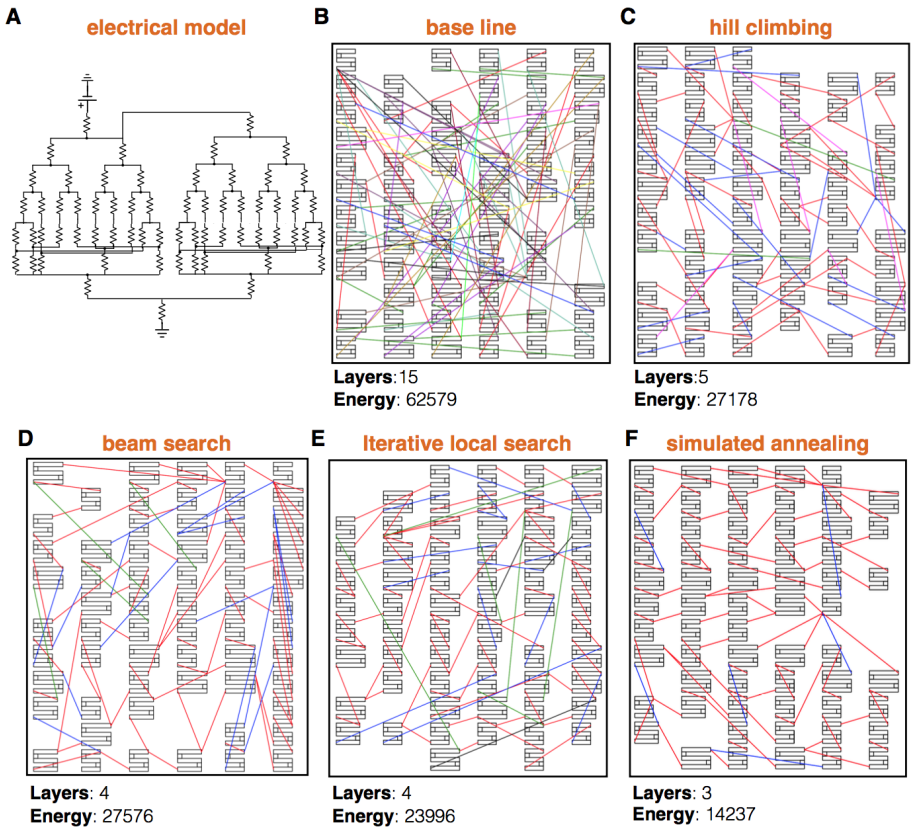

Figure 4.1: Schematics of all proposed designs for an electrical layout with 64 resistors with varying values and 150 connection. Colors indicate designation to a layer

ative bound of 500,000 and an exponentially dropping temperature, produced a design with an energy of 14,237 and only 4 intersections. This solution was clustered to a final design with 3 layers (figure 4.1, E). BS, ITL and SA worked on their solutions for a total of 4-6 minutes each.

In a similar fashion, optimization was made on additional four designs. Results for a model with 6 resistors and 8 interconnections are shown in figure 4.2, A. Results for a model with 8 resistors and 12 interconnections are shown in figure 4.2, B.

Results for a model with 10 resistors and 16 interconnections are shown in figure 4.3. Results for a model with 31 resistors and 39 interconnections are

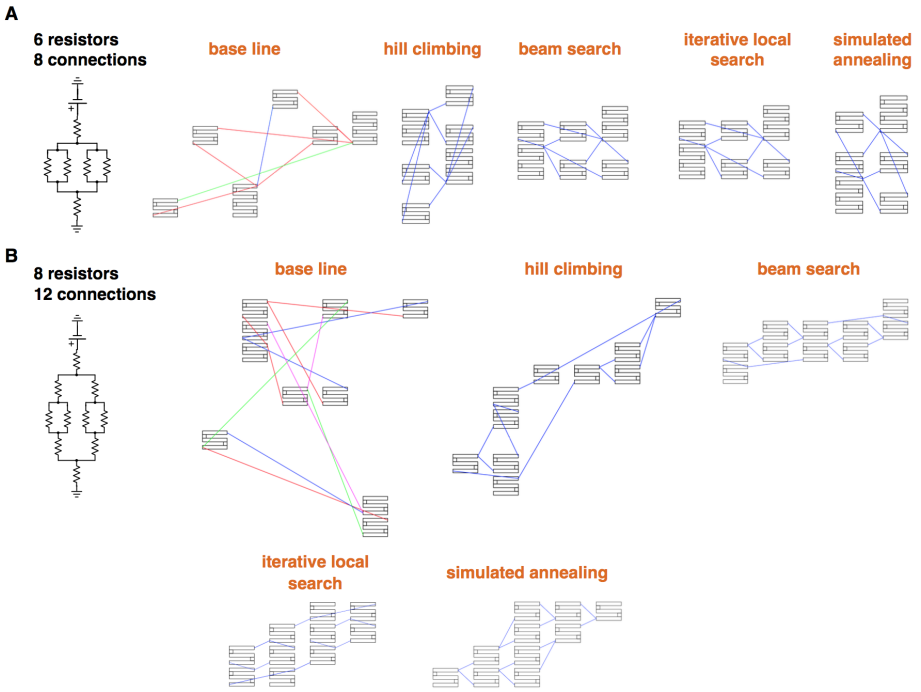

Figure 4.2: Results for electrical models with 6/8 resistors and 8/12 connections

shown in figure 4.3. A comparison between the performance of the optimization algorithms for each design is given in table 4.5.

4.2 Fabrication

The proposed design was 3D printed (figure 4.6) using Asica Pico 2, 3D printer. The Asiga printer has a build size of $51x32x76mm$, a x, y axis resolution of $39um$ and a z axis resolution of $1um$. Print speed was set to $40\frac{mm}{h}$. Light source was $385nm$ UV LED light. Since the material system of the Asiga printer was open, we printed the devices in a class 4 clean room.

This proposed design paradigm is aimed at providing a layout of a microfluidic circuit with a predefined set of resistance. This allow the circuit designer to set a specific fluid resistance profile that result with a specific flow rate distribution. To verify the gained velocity distribution in the proposed design we

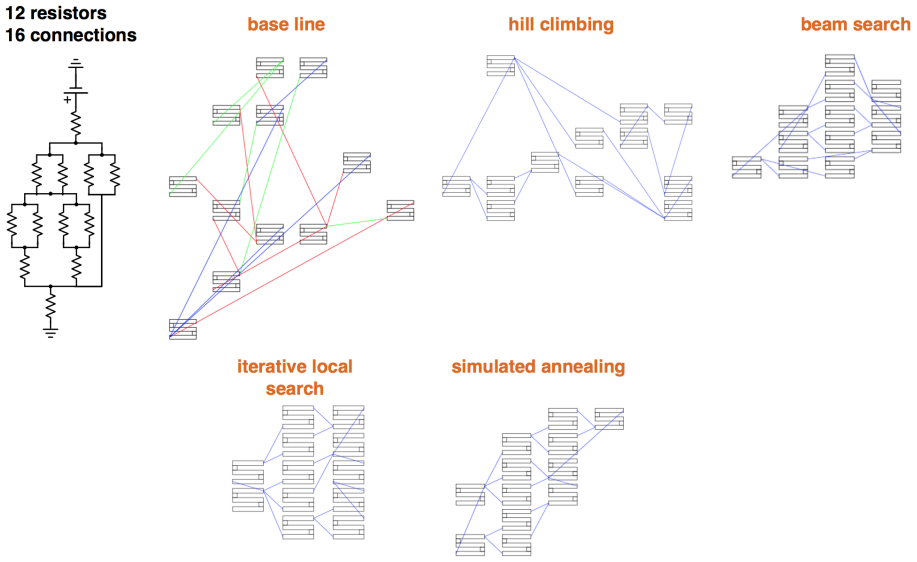

Figure 4.3: Results for electrical models with 10 resistors and 16 connections

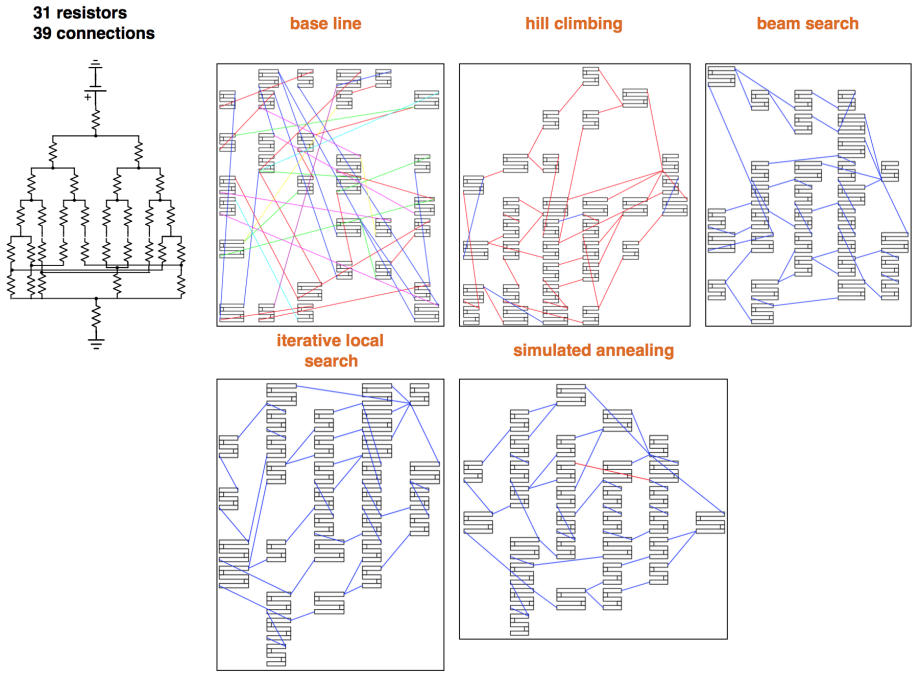

Figure 4.4: Results for electrical models with 31 resistors and 39 connections

Resisting channels	connections	base line				hill climbing			
		intersections	layers	Energy	Time (s)	intersections	layers	Energy	Time (s)
6	8	4	4	4154	<1	0	2	2309	1
8	12	22	5	9116	<1	0	2	3395	1
10	16	23	4	11716	<1	0	2	6172	1
31	39	149	8	29350	<1	4	3	10316	2
64	150	624	15	62579	<1	83	5	27178	2

Resisting channels	connections	beam search				iterative local search				simulated anealling			
		intersections	layers	Energy	Time (s)	intersections	layers	Energy	Time (s)	intersections	layers	Energy	Time (s)
6	8	0	2	1521	7	0	2	1521	16	0	2	1482	21
8	12	0	2	1734	11	0	2	1854	22	0	2	1385	22
10	16	0	2	2928	10	0	2	2269	28	0	2	2469	32
31	39	0	2	8095	42	0	2	7703	95	1	3	7121	98
64	150	38	4	27576	294	20	4	23996	356	19	3	14237	348

Figure 4.5: Comparison between the performance of the optimization algorithms

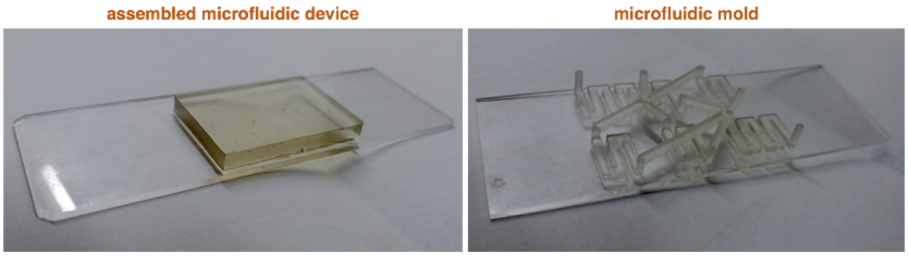

Figure 4.6: 3D printed microfluidic device and mold

performed Streamlines Image Analysis [27]. SIV utilizes long-exposure images to capture multiple particle streaklines in fully developed laminar flow, to derived velocity fields. Results show an excellent correspondence with R^2 of 0.94.

5

Discussion

Microfluidics is an exponentially growing technology, as it is attested the tremendous attention the field has attracted in the past decade. Current design paradigm for microfluidic circuits relies on intricate design-evaluation iterations, manual vector-graphics drawings and hand-operated fabrication. This design paradigm is often results with non-optimized layouts, a low satisfaction degree of physical and fabricational constraints and a hindrance to the process of the design-simulation cycle. Moreover, current microfluidic devices are predominantly fabricated using soft lithography in clean rooms. Soft lithography a is well established technique in the field and it allows for the fabrication of complex microfluidic structures such as mechanical valves and electrodes. However,

lithography is manually operated and therefore results in significant investment in time. 3D printing of hydraulic circuits aims to relax this manufacturing bottleneck, providing hand-free fabrication of devices in increasing resolution and complexity. Circuit layout optimization has been an important concept in the electronic industry since the introduction of PCBs. This approach however, was never utilized for microfluidic design. In a similar fashion to PCB and VLSI layout design, in this work we have provided advanced automated microfluidic geometric realizer for 3D printing. Such a system can dramatically diminish the concept-to-production time of microfluidic-based systems.

Our approach however is limited to a layered microfluidic design in which the hydraulic resistors are placed in one layer and the connecting channels in others. Future development may aim to unify the connection and resistors layers into a single layer, allowing fabrication with a single layer printing or straight forward lithography. Another advancement can be the incorporation of mechanical valves into the hydraulic layout. More profound advancement can be made by completing the automation of the design by systematizing the generation of the electrical model, given the specification of the microfluidic application.

To conclude, given an electrical model, our algorithm can significantly reduce the design time of spatially optimized microfluidic circuits, allowing a seamless computer-aided transition from concept to production.

6
Bibliography

[1] K. W. Oh, K. Lee, B. Ahn, and E. P. Furlani, "Design of pressure-driven microfluidic networks using electric circuit analogy," *Lab on a Chip*, vol. 12, no. 3, pp. 515–545, 2012.

[2] M. J. Fuerstman, P. Deschatelets, R. Kane, A. Schwartz, P. J. Kenis, J. M. Deutch, and G. M. Whitesides, "Solving mazes using microfluidic networks," *Langmuir*, vol. 19, no. 11, pp. 4714–4722, 2003.

[3] S. K. Dertinger, D. T. Chiu, N. L. Jeon, and G. M. Whitesides, "Generation of gradients having complex shapes using microfluidic networks," *Analytical Chemistry*, vol. 73, no. 6, pp. 1240–1246, 2001.

[4] G. M. Whitesides, "The origins and the future of microfluidics," *Nature*, vol. 442, no. 7101, pp. 368–373, 2006.

[5] G. Whitesides and A. Stroock, "Flexible methods for microfluidics [j]," *Phys Today*, vol. 54, no. 6, pp. 42–48, 2001.

[6] S. N. Bhatia, M. L. Yarmush, and M. Toner, "Controlling cell interactions by micropatterning in co-cultures: Hepatocytes and 3t3 fibroblasts," *Journal of Biomedical Materials Research*, vol. 34, no. 2, pp. 189–199, 1997.

[7] Y. K. Cheung, B. M. Gillette, M. Zhong, S. Ramcharan, and S. K. Sia, "Direct patterning of composite biocompatible microstructures using microfluidics," *Lab Chip*, vol. 7, pp. 574–579, 2007.

[8] X. Mu, W. Zheng, J. Sun, W. Zhang, and X. Jiang, "Microfluidics for manipulating cells," *Small*, vol. 9, no. 1, pp. 9–21, 2013.

[9] P. Emil, F. Jiang, S. Axel, R. Stephen, R. Clive, and A. French, "High-throughput multi-antigen microfluidic fluorescence immunoassays," *BioTechniques*, vol. 40, no. 1, p. 8590, 2006.

[10] S. Veronica, D. Antje, K. Tomer, R. Stephen, and C. Joseph, "Microfluidic single-cell real-time pcr for comparative analysis of gene expression patterns," *Nature Protocols*, vol. 7, p. 829838, 2012.

[11] T. Das and S. Chakraborty, "Biomicrofluidics: Recent trends and future challenges," *Sadhana*, vol. 34, no. 4, pp. 573–590, 2009.

[12] C. T. Ho, R. Z. Lin, W. Y. Chang, H. Y. Chang, and C. H. Liu, "Rapid heterogeneous liver-cell on-chip patterning via the enhanced field-induced dielectrophoresis trap," *Lab Chip*, vol. 6, pp. 724–734, 2006.

[13] T. Kalisky, P. Blainey, and S. R. Quake, "Genomic analysis at the single-cell level," *Annual Review of Genetics*, vol. 45, no. 1, pp. 431–445, 2011. PMID: 21942365.

[14] F. K. Balagadd, L. You, C. L. Hansen, F. H. Arnold, and S. R. Quake, "Long-term monitoring of bacteria undergoing programmed population control in a microchemostat," *Science*, vol. 309, no. 5731, pp. 137–140, 2005.

[15] A. R. Wheeler, W. R. Throndset, R. J. Whelan, A. M. Leach, R. N. Zare, Y. H. Liao, K. Farrell, I. D. Manger, and A. Daridon, "Microfluidic device for single-cell analysis," *Analytical Chemistry*, vol. 75, no. 14, pp. 3581–3586, 2003.

[16] J. Voldman, M. L. Gray, M. Toner, and M. A. Schmidt, "A microfabrication-based dynamic array cytometer," *Analytical Chemistry*, vol. 74, no. 16, pp. 3984–3990, 2002.

[17] E. Oosterbroek, *Modeling, Design and Realization of Microfluidic Component*. PhD thesis, University of Twente, Enschede, 1999.

[18] A. Chaudhari, T. Ghoshal, M. T. Shaw, C. Cummins, D. Borah, J. D. Holmes, and M. A. Morris, "Formation of sub-7 nm feature size ps-b-p4vp block copolymer structures by solvent vapour process," *Proc. SPIE*, vol. 9051, pp. 1–10, 2014.

[19] M. Jackson, *Microfabrication and Nanomanufacturing*. CRC Press, 2005.

[20] K. Pilnam, W. Keon, C. Min, H. Sung, M. Sun, and Y. Kahp, "Soft lithography for microfluidics: a review," *Biochip Journal*, vol. 2, pp. 1–11, 2008.

[21] P. Yao, G. Schneider, D. Prather, E. Wetzel, and D. O'Brien, "Fabrication of three-dimensional photonic crystals with multilayer photolithography," *Opt. Express*, vol. 13, pp. 2370–2376, Apr 2005.

[22] P. J. Kitson, M. H. Rosnes, V. Sans, V. Dragone, and L. Cronin, "Configurable 3d-printed millifluidic and microfluidic 'lab on a chip' reactionware devices," *Lab Chip*, vol. 12, pp. 3267–3271, 2012.

[23] B. C. Gross, J. L. Erkal, S. Y. Lockwood, C. Chen, and D. M. Spence, "Evaluation of 3d printing and its potential impact on biotechnology and the chemical sciences," *Analytical Chemistry*, vol. 86, no. 7, pp. 3240–3253, 2014.

[24] K. C. Bhargava, B. Thompson, and N. Malmstadt, "Discrete elements for 3d microfluidics," *Proceedings of the National Academy of Sciences*, vol. 111, no. 42, pp. 15013–15018, 2014.

[25] Y. Nahmias and S. Bhatia, *Microdevices in Biology and Medicine*. Artech House, 2009.

[26] D. R. Mott, P. B. Howell Jr, J. P. Golden, C. R. Kaplan, F. S. Ligler, and E. S. Oran, "Toolbox for the design of optimized microfluidic components," *Lab on a Chip*, vol. 6, no. 4, pp. 540–549, 2006.

[27] E. Keinan, E. Ezra, and Y. Nahmias, "Frame rate free image velocimetry for microfluidic devices," *Applied physics letters*, vol. 103, no. 6, p. 063507, 2013.

[28] J. R. Anderson, D. T. Chiu, H. Wu, O. Schueller, and G. M. Whitesides, "Fabrication of microfluidic systems in poly (dimethylsiloxane)," *Electrophoresis*, vol. 21, no. 1, pp. 27–40, 2000.

[29] L. G. Valiant, "Universality considerations in vlsi circuits," *IEEE Transactions on Computers*, vol. 100, no. 2, pp. 135–140, 1981.

[30] K. Inoue, K. Shimada, and K. Chilaka, "Solid model reconstruction of wireframe cad models based on topological embeddings of planar graphs," *Journal of Mechanical Design*, vol. 125, no. 3, pp. 434–442, 2003.

[31] P. J. Kitson, M. H. Rosnes, V. Sans, V. Dragone, and L. Cronin, "Configurable 3d-printed millifluidic and microfluidic lab on a chipreactionware devices," *Lab on a Chip*, vol. 12, no. 18, pp. 3267–3271, 2012.

[32] B. C. Gross, J. L. Erkal, S. Y. Lockwood, C. Chen, and D. M. Spence, "Evaluation of 3d printing and its potential impact on biotechnology and the chemical sciences," *Analytical chemistry*, vol. 86, no. 7, pp. 3240–3253, 2014.

[33] K. C. Bhargava, B. Thompson, and N. Malmstadt, "Discrete elements for 3d microfluidics," *Proceedings of the National Academy of Sciences*, vol. 111, no. 42, pp. 15013–15018, 2014.

[34] N. Bhattacharjee, A. Urrios, S. Kang, and A. Folch, "The upcoming 3d-printing revolution in microfluidics," *Lab on a Chip*, vol. 16, no. 10, pp. 1720–1742, 2016.

[35] S. Takayama, J. C. McDonald, E. Ostuni, M. N. Liang, P. J. Kenis, R. F. Ismagilov, and G. M. Whitesides, "Patterning cells and their environments using multiple laminar fluid flows in capillary networks," *Proceedings of the National Academy of Sciences*, vol. 96, no. 10, pp. 5545–5548, 1999.

[36] D. B. Weibel and G. M. Whitesides, "Applications of microfluidics in chemical biology," *Current opinion in chemical biology*, vol. 10, no. 6, pp. 584–591, 2006.

[37] G. M. Walker, J. Sai, A. Richmond, M. Stremler, C. Y. Chung, and J. P. Wikswo, "Effects of flow and diffusion on chemotaxis studies in a microfabricated gradient generator," *Lab on a Chip*, vol. 5, no. 6, pp. 611–618, 2005.

[38] K. Lee, E. A. Silva, and D. J. Mooney, "Growth factor delivery-based tissue engineering: general approaches and a review of recent developments," *Journal of the Royal Society Interface*, vol. 8, no. 55, pp. 153–170, 2011.

[39] J. Gurdon and P.-Y. Bourillot, "Morphogen gradient interpretation," *Nature*, vol. 413, no. 6858, pp. 797–803, 2001.

[40] R. A. Brennen, H. Yin, and K. P. Killeen, "Microfluidic gradient formation for nanoflow chip lc," *Analytical chemistry*, vol. 79, no. 24, pp. 9302–9309, 2007.

[41] A. Tirella, M. Marano, F. Vozzi, and A. Ahluwalia, "A microfluidic gradient maker for toxicity testing of bupivacaine and lidocaine," *Toxicology in vitro*, vol. 22, no. 8, pp. 1957–1964, 2008.

[42] J. Pihl, J. Sinclair, E. Sahlin, M. Karlsson, F. Petterson, J. Olofsson, and O. Orwar, "Microfluidic gradient-generating device for pharmacological profiling," *Analytical chemistry*, vol. 77, no. 13, pp. 3897–3903, 2005.

[43] S. Kim, H. J. Kim, and N. L. Jeon, "Biological applications of microfluidic gradient devices," *Integrative Biology*, vol. 2, no. 11-12, pp. 584–603, 2010.

[44] S. E. Hulme, S. S. Shevkoplyas, J. Apfeld, W. Fontana, and G. M. Whitesides, "A microfabricated array of clamps for immobilizing and imaging c. elegans," *Lab on a Chip*, vol. 7, no. 11, pp. 1515–1523, 2007.

[45] W. Magnuson, "A comparison of constructive placement algorithms," in *Region Six Conference Record, 1977. IEEE 1977*, pp. 28–32, IEEE, 1977.

[46] S. Areibi and Z. Yang, "Effective memetic algorithms for vlsi design= genetic algorithms+ local search+ multi-level clustering," *Evolutionary Computation*, vol. 12, no. 3, pp. 327–353, 2004.

[47] T. Pannérec, "Knowledge-based automatic components placement for single-layer pcb layout," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 669–675, Springer, 2003.

[48] C. Sechen, *VLSI placement and global routing using simulated annealing*, vol. 54. Springer Science & Business Media, 2012.

[49] S. M. Sait, M. Faheemuddin, M. R. Minhas, and S. Sanaullah, "Multiobjective vlsi cell placement using distributed genetic algorithm," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pp. 1585–1586, ACM, 2005.

[50] L. Zhang, R. Raut, Y. Jiang, U. Kleine, and Y. Kim, "A hybrid evolutionary analogue module placement algorithm for integrated circuit layout designs," *International journal of circuit theory and applications*, vol. 33, no. 6, pp. 487–501, 2005.

[51] S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards, *Artificial intelligence: a modern approach*, vol. 2. Prentice hall Upper Saddle River, 2003.

[52] J. Melin and S. R. Quake, "Microfluidic large-scale integration: the evolution of design rules for biological automation," *Annu. Rev. Biophys. Biomol. Struct.*, vol. 36, pp. 213–231, 2007.

[53] A. C. Yunus and J. M. Cimbala, "Fluid mechanics: fundamentals and applications," *International Edition, McGraw Hill Publication*, pp. 185–201, 2006.

[54] N. Abboud, M. Grötschel, and T. Koch, "Mathematical methods for physical layout of printed circuit boards: an overview," *OR Spectrum*, vol. 30, no. 3, pp. 453–468, 2008.

[55] T. H. Cormen, *Introduction to algorithms*. MIT press, 2009.

תקציר

התקנים מיקרופלואידיים הינם "מערכות על שבב" (lab on a chip) ויישומם חולש על עולם המחקר מזה הביולוגי-מולקולרי ועד לזה הקוואנטי-פיסיקלי. תכנון וארכיטקטורה של התקנים מיקרופלואידיים מתחיל עם אילוצי מערכת ומסתיים בהסקתו של גרף טופולוגי המתאר ארכיטקטורת רשת מכאנית דו-מימדית. אופן תכנון זה מסתמך פעמים רבות על מודלים נומריים ומידע נסיוני ועל כן אינו מאפשר תכנון מונחה פרמטרים, דורש משאבים חישוביים ניכרים ואינו אפקטיבי כלכלית. יתר על כן, פרדיגמת הייצור של התקנים מיקרופלואידיים נסמכת בעיקרה על ליתוגרפיה ועל כן תלויה בהתערבות ידנית.

במחקר זה ניצלנו אנלוגיה תיאורטית שבין מעגלים הידראולים למעגלים חשמליים על מנת להגדיר רשת מיקרופלואידית מופשטת. תכננו אלגוריתם שבהתבסס על אילוצים פיזיקליים ופרוטוקולי אופטימיזציה, מציע תכנון למעגל מכני בעל גודל מינימאלי. לבסוף יצרנו מנגנון תכנון אוטומטי המייצר את התכנון המוצע באמצעות גרפיקה ווקטורית באופן כזה שיתאים להדפסה תלת-מימדית.

יש בגישה המוצעת במחקר פוטנציאל למזער את זמן התכנון של מעגלים מיקרופלואידיים, באופן זה היאפשר מעבר ישיר מרעיון לייצור.

עבודה זו בוצעה בהנחיית פרופ' אריק שמיר מבית הספר אפי ארזי למדעי המחשב,
המרכז הבינתחומי, הרצליה.

המרכז הבינתחומי הרצליה
בית הספר אפי ארזי למדעי המחשב
התוכנית לתואר שני (M.Sc) - מסלול מחקרי

תכנון מונחה מחשב של מעגלי התנגדות הידראולים להדפסה תלת מימדית

מאת
אלישי עזרא צור

עבודת תזה המוגשת כחלק מהדרישות לשם קבלת תואר מוסמך M.Sc במסלול המחקרי בבית ספר אפי ארזי למדעי המחשב, המרכז הבינתחומי הרצליה

ינואר 2017