3M-DeSyn: Design Synthesis for Multi-Layer 3D-Printed Microfluidics with Timing and Volumetric Control

Yushen Zhang
yushen.zhang@tum.de
Technical University of Munich
Munich, Germany

Dragan Rašeta
dragan.raseta@tum.de
Technical University of Munich
Munich, Germany

Tsun-Ming Tseng
tsun-ming.tseng@tum.de
Technical University of Munich
Munich, Germany

Ulf Schlichtmann
ulf.schlichtmann@tum.de
Technical University of Munich
Munich, Germany

ABSTRACT

3D printing has revolutionized microfluidic device fabrication, enabling rapid prototyping and intricate geometries. However, designing lab-on-a-chip systems remains challenging. In microfluidic devices, precise control over fluid behavior is crucial, requiring careful attention to both timing and volume. Current state-of-the-art design automation tools for microfluidics have limitations, particularly in addressing the specific challenges of 3D-printed microfluidics and user-defined timing and volumetric constraints, and no design synthesis tool exists targeting these domains. We present 3M-DeSyn, a novel design synthesis method for 3D-printed microfluidics that incorporates timing and volumetric constraints and outputs print-ready 3D modeling files. It automates the design process, allowing users to specify schematics and desired flow control parameters. The underlying methodology is based on mathematical modeling of fluidic behavior and utilizes constraint optimization programming to find optimized solutions. Experimental results show significant improvements in design time while enabling rapid development of custom microfluidic systems.

CCS CONCEPTS

• **Hardware** → **Timing analysis**; **Software tools for EDA**; **Physical synthesis**; *Placement*; *Wire routing*; **Biology-related information processing**; Analysis and design of emerging devices and systems.

KEYWORDS

Design Synthesis, 3D Printing, Microfluidics, Lab-on-a-Chip, Biochip, Design Automation, Timing Control, Volume Control

ACM Reference Format:
Yushen Zhang, Dragan Rašeta, Tsun-Ming Tseng, and Ulf Schlichtmann. 2025. 3M-DeSyn: Design Synthesis for Multi-Layer 3D-Printed Microfluidics with Timing and Volumetric Control. In *30th Asia and South Pacific Design Automation Conference (ASPDAC '25), January 20–23, 2025, Tokyo, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3658617.3697787

1 INTRODUCTION

Microfluidic devices enable the miniaturization and automation of complex laboratory processes, revolutionizing various fields, from point-of-care diagnostics and drug delivery to cell biology and chemical synthesis [8]. Conventional microfluidic fabrication techniques like photolithography and polydimethylsiloxane (PDMS) glass bonding often involve complex cleanroom processes. In recent years, three-dimensional printing (3D printing) has emerged as a compelling alternative, offering rapid prototyping capabilities and enabling the creation of intricate geometries that are not possible with conventional methods [9, 10, 16].

Designing microfluidic devices is a challenging task that requires specialized knowledge in fluid dynamics, mechanical design, and manufacturing. Microfluidic devices rely on precise manipulation of fluids at the microscale. This manipulation often necessitates careful consideration of both timing and volume constraints during the design process [4, 7, 12, 15]. Timing constraints are crucial for various applications, such as drug delivery and chemical synthesis. Precise control over the timing of fluid flow ensures accurate mixing of reagents, synchronization of reactions, and proper sequencing of operations. For instance, in enzymatic reactions, substrates and enzymes might need to meet at a defined time difference to optimize reaction yield [7]. Volumetric constraints are critical for applications that require precise metering or dispensing of fluid volumes, such as drug delivery, chemical analysis, and cell culture [2, 3, 6]. Achieving specific reaction times or mixing ratios in microfluidic applications frequently requires careful design of microfluidic channels, considering factors like channel dimensions, flow rates, and actuation mechanisms.

In particular, unlike conventional PDMS microfluidic devices such as microfluidic large-scale integration (mLSI) chips, on which the fluid delivery can be controlled with a series of valves and pumps, 3D-printed microfluidics mostly consist of flow layers only, making it difficult to integrate those pneumatic structures for timing and volumetric control. While researchers have reported pneumatic valves for 3D printing [1, 10, 14], their applicability remains limited due to the need for customized printing materials and printers. Thus, correctly designing the geometric structure of the microfluidic chip features, such as length, cross-section, and curvature on fluid flow

ASPDAC '25, January 20–23, 2025, Tokyo, Japan

ACM ISBN 979-8-4007-0635-6/25/01
https://doi.org/10.1145/3658617.3697787

rates, timing, and volume, remains the primary way to ensure the timing and volume constraints required for individual experiments.

The traditional design process, often reliant on manual computer-aided design (CAD) modeling, can be time-consuming and prone to errors. The labor-intensive design tasks include architecture research, manual layout design, fabrication process optimization, etc. The design complexity further increases for 3D-printed multi-layer devices, which require designers to master 3D-capable modeling software and fabrication skills.

Several microfluidic design automation tools have been developed in recent years to assist the design process of microfluidic devices. For example, 3DμF [11] is an interactive design automation tool that streamlines the design process of planar chips. However, its functionality is limited to 2D layouts and doesn't encompass design synthesis capabilities. Cloud Columba [13] is a design synthesis tool for continuous-flow mLSI, also targeting 2D chip layouts with dedicated flow and control layers, excluding 3D-printed microfluidics. Flui3d [17] is a state-of-the-art interactive design automation tool for 3D-printed microfluidics specifically targeting 3D structured multilayered chip layouts. Despite its significant contribution to automating 3D-printed microfluidics design, the absence of integrated design synthesis functions necessitates a manual design effort.

In general, despite significant advances, current design automation tools lack the functionality to incorporate user-defined timing and volumetric requirements, which are crucial aspects of microfluidic device design. As a result, there is a pressing need for a design synthesis tool that not only automates the placement of components and routing of channels but also optimizes their geometric parameters to satisfy the timing and volumetric constraints within a three-dimensional structure.

In this paper, we present a novel design synthesis method for 3D-printed microfluidics: *3M-DeSyn*, which automatically synthesizes print-ready 3D modeling files incorporating user-specified timing and volumetric constraints, such as the desired timing and volume ratios and input flow rates. We mathematically model the fluidic behavior and use constraint optimization programming (COP) to minimize the chip size. The remainder of this paper is organized as follows: Section 2 introduces 3D-printing microfluidic fabrication and details the timing and volumetric constraints. Section 3 presents the problem formulation. Section 4 describes the overview of the proposed synthesis method. Section 5 details the proposed synthesis method. Section 6 presents and discusses the experimental results, and a conclusion is drawn in Section 7.

2 BACKGROUND

2.1 3D-Printed Microfluidics

3D-printed microfluidic chips are fabricated using additive manufacturing technologies such as PolyJet, Fused Deposition Modeling (FDM), or Stereolithography (SLA). These chips typically consist of flow layers only, without the need for additional bonding or assembly steps. Among the various 3D printing technologies employed for microfluidics fabrication, SLA has emerged as the most popular choice due to its outstanding precision and resolution.

In the SLA process, a light source is used to selectively cure a photosensitive resin layer by layer, enabling the creation of intricate microfluidic structures with high resolution. The high precision and resolution afforded by SLA make it possible to print multi-layer microfluidic chips, allowing for the vertical stacking of microfluidic components. This 3D chip architecture enables the integration of complex fluidic networks within a compact footprint, further minimizing the overall size of the microfluidic devices.

2.2 Timing and Volumetric Constraints

In contrast to conventional microfluidic chips that employ pneumatic valves and pumps for precise fluid control, 3D-printed microfluidic chips primarily rely on the exact design of internal geometries and fluidic pathways to achieve the desired timing and volume control of fluid delivery. This design-driven approach arises due to the typical absence of a dedicated control layer in 3D-printed chips, necessitating careful optimization of channel dimensions and network architecture.

A microfluidic application with timing constraints demands that fluids introduced through designated inlets with given flow rates reach their destinations at designated times. Specifically, a timing constraint is defined by (1) given inlet(s), (2) respective flow rate(s), (3) destination, and (4) time or time shift. Figure 1 (a) illustrates two timing constraints Tr_1 and Tr_2. Tr_1 specifies that fluids originating from inlets I_1 and I_2 (with equal flow rates, $Q_1 = Q_2 = 150\mu l/m$) must arrive at chamber C_1 simultaneously; and Tr_2 specifies that fluids from all three inlets, I_1, I_2, and I_3 (with a lower flow rate $Q_3 = 100\mu l/m$), must arrive at chamber C_2 simultaneously.

A microfluidic application with volumetric constraints demands that fluids introduced through designated inlets reach their destinations with defined volumetric values or ratios. Specifically, a volumetric constraint is defined by (1) given inlet(s), (2) respective flow rate(s), (3) destination, and (4) volume or volume ratio. Figure 1 (b) illustrates a volumetric constraint Vr_1, which specifies that fluids originating from inlets I_1 and I_2 (with $Q_1 = 200\mu l/m$ and $Q_2 = 300\mu l/m$) must fill the chamber C_1 with a 2:3 ratio.

3 PROBLEM FORMULATION

3M-DeSyn takes a schematic description of the required design as input. Optional constraints such as the maximum and minimum number of layers, designation of component position, timing and volumetric requirements, and flow rates at the inlets can be provided to further refine the desired output. Simultaneous entry of the fluids at the inlets and steady flow rates are assumed. Specifically, we address the following problem:

Input:

- A schematic description of the required design, including component definitions and inter-relationships.
- Optional dimensional requirements, such as margins, minimum channel width, or number of layers.
- Optional positional requirements, such as a component on a specific layer.
- Optional timing requirements, including flow rates at the corresponding inlets.
- Optional volumetric requirements, including flow rates at the corresponding inlets.

Objective: Find the optimized multilayered placement of components and suitable routing paths and dimensions for the channels,

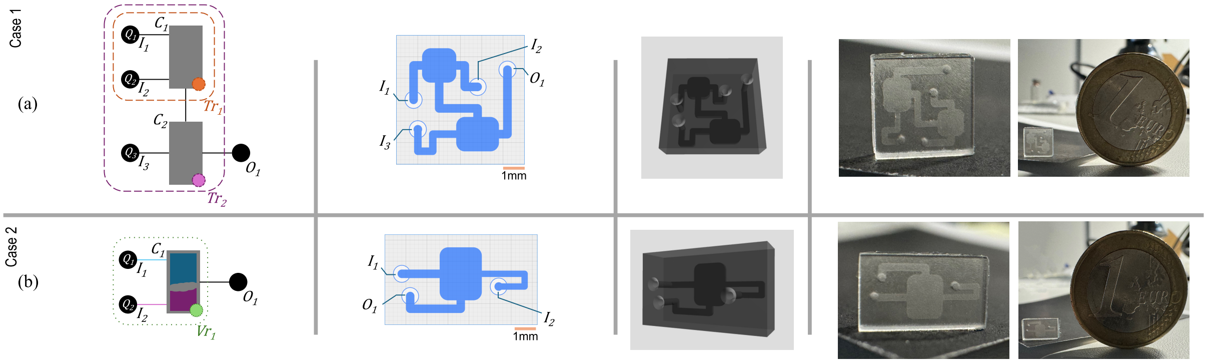

Figure 1: Microfluidic applications with (a) timing and (b) volumetric constraints. The left column depicts the schematics. The dashed lines and dotted lines represent the timing (Tr) and volumetric (Vr) constraints, respectively. The colored dots indicate the destinations. The two center columns depict the physical layouts and the 3D-printable files synthesized by 3M-DeSyn. The right column shows the photos of the 3D-printed chips alongside a one-euro coin for size comparison.

considering the given timing and volumetric requirements, to minimize the overall chip size.

Output: A 3D modeling file that represents the optimized solution, incorporating the component placement and channel routing.

4 OVERVIEW OF PROPOSED METHOD

Figure 2 illustrates the overall synthesis flow of 3M-DeSyn. Given an input, if timing and/or volumetric constraints are specified, a graph-based representation of the chip schematic is constructed, where components are modeled as nodes and their inter-relationships as edges. An analysis is then performed on the graph to identify the flow paths that are relevant to the timing and/or volumetric constraints. Based on the analysis, we mathematically model the required constraints, which, together with the mathematical constraints for the placement and routing problem, construct a comprehensive optimization model, aiming to minimize the chip’s overall dimensions and enhance efficiency in terms of fluid flow (hydraulic resistance) while ensuring other attributes, such as channel widths and lengths, are refined based on the specified timing and volumetric constraints. We then solve the model with an optimizer and interpret the chip layout as an STL 3D modeling file, which is a standard file format for 3D printing.

5 MULTI-LAYER PHYSICAL SYNTHESIS

5.1 Graph Creation and Pathfinding Analysis

If any timing and/or volumetric constraints are given, a graph representation of the chip schematic is constructed to differentiate the flow paths taken by different fluids from their inlets to the shared destination. The overall pathfinding procedure for handling timing and volumetric constraints, presented in Algorithm 1, leverages the principles of Breadth-First Search (BFS) with modifications tailored to this specific application. The inputs to the procedure are inlets, their corresponding flow rates, a destination component, and volume offsets. Volume offsets reflect the timing and volumetric constraints and are calculated based on the destination component’s volume and the desired volume ratio of input fluids or the required time shift for arrival. If the fluids from all inlets must arrive simultaneously (time shift of 0) or the volume ratio is 1:1, all volume offsets will be 0.

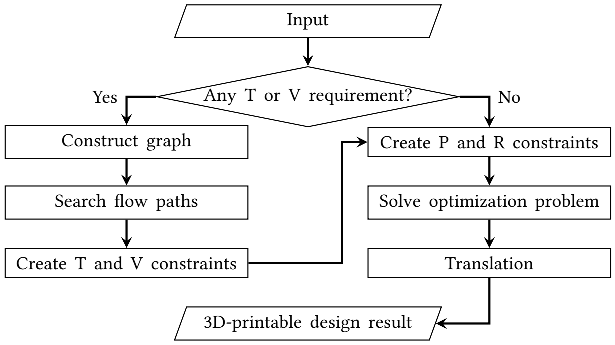
Input
Any T or V requirement?
Yes
No
Construct graph
Search flow paths
Create T and V constraints
Create P and R constraints
Solve optimization problem
Translation
3D-printable design result

Figure 2: Overview of proposed 3M-DeSyn method. T - timing, V - volumetric, P - placement, R - routing.

We first initialize the data structures. An empty queue is created to denote the queue of nodes for exploration. For each node in the array of inlet nodes, empty lists are created to store volumes and flow rates. These lists track the flow volume and the corresponding flow rates experienced by each node throughout the search.

Next, we prepare the inlet nodes for exploration by appending the volume offset to the corresponding volume list for each node. The node’s flow rate is appended to its corresponding flow rate list. All nodes from the inlet node array are then enqueued.

The main loop iterates through the queue, exploring possible paths from inlet nodes to the destination node. An adjacent list is retrieved for the current node, containing all unvisited nodes connected to it. To ensure that only one flow path exits the node and to check for any unvisited adjacent nodes, the current node may be re-enqueued back into the queue if there is more than one adjacent node. The loop then continues to the next iteration. The current node is marked as visited. If multiple flows from different inlets converge at this node, corresponding COP equality constraints are created based on the current volumes and flow rates, and the

Algorithm 1 Algorithm to search the flow paths from the inlet nodes to the destination node to create COP constraints

```
Input: An array of inlet nodes $I[]$, their corresponding flow rates $F[]$, their volume offsets $V[]$, and a destination node $d$;
Output: Corresponding COP constraints;
$Q \leftarrow$ new empty queue
for $i \in I$ do
    $i.volumes.append(V[i])$
    $i.flowRates.append(F[i])$
end for
$Q$.enqueue($I$)
while $Q \neq \emptyset$ do
    $current \leftarrow Q.dequeue()$
    $adjacent \leftarrow$ list of all unvisited nodes adjacent to $current$
    if $|adjacent| > 1 \land Q \neq \emptyset$ then
        $Q$.enqueue($current$)
        continue
    end if
    mark $current$ as visited
    if $|current.volumes| > 1$ then
        Create corresponding COP constraints
        continue
    end if
    if $current = d$ then
        break
    end if
    $nextVolume \leftarrow sum(current.volumes) + current.calculateVolume()$
    $nextFlowrate \leftarrow sum(current.volumes)$
    for $n \in adjacent$ do
        $n.volumes.append(nextVolume)$
        $n.flowRates.append(nextFlowrate)$
        $Q$.enqueue($n$)
    end for
end while
```

procedure continues to the next iteration. If the current node is the destination node, the loop terminates as the path has been found. The updated volume and flow rate for the current node are then calculated. The procedure iterates through adjacent nodes of the current node, updating their volume and flow rate information and enqueuing them into the queue. This process effectively extends the search to unvisited neighbors, marking them for exploration in subsequent iterations. The loop continues as long as the queue is not empty, systematically exploring all possible paths from the inlet nodes to the destination node. After exploring the graph, constraints are established for timing and volumetric requirements at any node with multiple inlet connections.

5.2 Multi-layer Chip, Component, and Channel

A 3D-printed microfluidic chip typically comprises multiple layers, each containing various components interconnected by channels. Vertical channels, called vias, enable fluid flow between layers. Each component has a specific size defined by a bounding box. We represent the multilayered layout of a 3D-printed microfluidic chip as a collection of 2D coordinate planes. Each plane corresponds to a distinct layer of the chip.

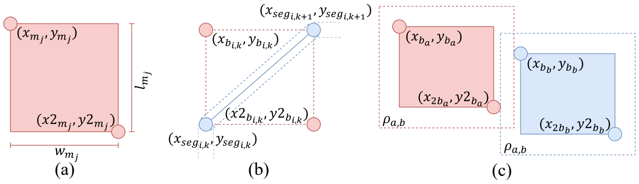

Figure 3: Illustration of (a) a component, (b) a channel segment and its corresponding bounding box, and (c) the bounding boxes of components.

As shown in Figure 3 (a), a component m_j ($j \in N$) is defined by its top-left corner coordinates (x_{m_j}, y_{m_j}), width (w_{m_j}), length (l_{m_j}), and layer (ly_{m_j}). Additionally, four binary orientation variables ($d^{0}_{m_j}$, $d^{90}_{m_j}$, $d^{180}_{m_j}$, and $d^{270}_{m_j}$) specify the module's orientation, with only one variable being true at a time. These variables allow for rotations in 0°, 90°, 180°, and 270°. The bottom-right corner coordinates ($x2_{m_j}$ and $y2_{m_j}$) can therefore be modeled as follows:

$$x2_{m_j} = x_{m_j} + (d^{0}_{m_j} + d^{180}_{m_j}) \cdot w_{m_j} + (d^{90}_{m_j} + d^{270}_{m_j}) \cdot l_{m_j}, \quad (1)$$

$$y2_{m_j} = y_{m_j} + (d^{0}_{m_j} + d^{180}_{m_j}) \cdot l_{m_j} + (d^{90}_{m_j} + d^{270}_{m_j}) \cdot w_{m_j}. \quad (2)$$

The set of all components in a design is denoted by M.

Channels are modeled as polylines connecting two components. Each channel c_i ($i \in N$) has a specified width (w_{c_i}). The channel path is defined by a sequence of points ($seg_{i,0}$, $seg_{i,1}$, ..., $seg_{i,n+1}$), where $n \in N$ and $n \geq 2$ represents the number of segments in the polyline, as illustrated in Figure 3 (b). This number is set in advance, and any unnecessary segments are reduced to a length of zero during optimization.

A channel segment $(seg_{i,k})$ has four orientation options: horizontal, diagonal, vertical, or inter-layer, defined by a set of four binary variables ($d^{0}_{seg_{i,k}}$, $d^{45}_{seg_{i,k}}$, $d^{90}_{seg_{i,k}}$, and $d^{via}_{seg_{i,k}}$), among which only one can be true at any given time. Analogous to components, each channel segment is defined by its top-left corner coordinates $(x_{seg_{i,k}}, y_{seg_{i,k}})$, width $(w_{seg_{i,k}})$, length $(l_{seg_{i,k}})$, layer $(ly_{seg_{i,k}})$ and the bottom-right corner coordinates $(x2_{seg_{i,k}}, y2_{seg_{i,k}})$. The set containing all channels in a design is denoted by C.

The chip itself is denoted as $chip$, with variables representing its physical dimensions: width (w_{chip}), length (l_{chip}), and number of layers (nl_{chip}). These dimensions are used to constrain the placement of modules and channels within the chip boundaries.

All components ($m_j \in M$) are confined within the chip dimensions by the following constraints:

$$x2_{m_j} + \rho_c \leq w_{chip}, \quad y2_{m_j} + \rho_c \leq l_{chip}, \quad ly_{m_j} \leq nl_{chip}. \quad (3)$$

Here, ρ_c is a user-defined constant representing the minimum clearance between components/channels and the chip boundary.

Similar constraints confine individual channel segments $seg_{i,k}$ for all channels $c_i \in C$:

$$x_{seg_{i,k}} + w_c/2 + \rho_c \leq w_{chip}, \quad y_{seg_{i,k}} + w_c/2 + \rho_c \leq l_{chip}, \quad (4)$$

$$ly_{seg_{i,k}} \leq nl_{chip}. \quad (5)$$

These constraints, along with the model's objective function that will be described later, ensure that the resulting chip design is as compact as possible. Chip volume is an important metric as smaller chips translate to reduced fabrication costs and time.

5.3 Non-Overlapping Constraints

To achieve proper layout optimization and prevent overlapping components or channels, a bounding box approach is employed. Each feature (component or channel segment) is enclosed by a rectangle, defining its minimum clearance from other features.

A channel segment ($seg_{i,k}$) with two endpoints ($x_{seg_{i,k}}, y_{seg_{i,k}}$) and ($x_{seg_{i,k+1}}, y_{seg_{i,k+1}}$) is considered as the diagonal of its bounding box $b_{i,k}$, as illustrated in Figure 3 (b). The bounding box is defined by its top-left corner ($x_{b_{i,k}}, y_{b_{i,k}}$) and bottom-right corner ($x2_{b_{i,k}}, y2_{b_{i,k}}$), which can be modeled as follows:

$$x_{b_{i,k}} = min(x_{seg_{i,k}}, x_{seg_{i,k+1}}), \quad y_{b_{i,k}} = min(y_{seg_{i,k}}, y_{seg_{i,k+1}}), \tag{6}$$

$$x2_{b_{i,k}} = max(x_{seg_{i,k}}, x_{seg_{i,k+1}}), \; y2_{b_{i,k}} = max(y_{seg_{i,k}}, y_{seg_{i,k+1}}). \tag{7}$$

The layer of the bounding box, denoted by $ly_{b_{i,k}}$ is simply inherited from the channel segment's layer:

$$ly_{b_{i,k}} = ly_{seg_{i,k}}. \tag{8}$$

For brevity, the notation b_j is also used to represent a component m_j's bounding box. Here, the definition mirrors the channel segment case, with coordinates derived from the component's corner points and layer.

Six distinct relative positions ensure no overlap between two bounding boxes b_a and b_b:

$$x2_{b_a} + \rho_{a,b} \leq x_{b_b} + q_1 M, \quad y2_{b_a} + \rho_{a,b} \leq y_{b_b} + q_2 M, \tag{9}$$

$$x2_{b_b} + \rho_{a,b} \leq x_{b_a} + q_3 M, \quad y2_{b_b} + \rho_{a,b} \leq y_{b_a} + q_4 M, \tag{10}$$

$$ly_{b_a} + 1 \leq ly_{b_b} + q_5 M, \quad ly_{b_b} + 1 \leq ly_{b_a} + q_6 M, \tag{11}$$

$$q_1 + q_2 + q_3 + q_4 + q_5 + q_6 = 5. \tag{12}$$

Here, q_1 to q_6 are auxiliary binary variables, M is a very large constant, and $\rho_{a,b}$ represents the minimum spacing required between bounding boxes b_a and b_b, as depicted in Figure 3 (c). The final constraint (Eq. 12) ensures that at least one of the binary variables (q_1 to q_6) is equal to 0. This guarantees that the relative positions of b_a and b_b satisfy at least one of the six conditions listed above (Eq. 9 - Eq. 11). The constraints Eq. 11 can be omitted if a bounding box must not overlap any other components on any layer.

5.4 Timing and Volumetric Control

The relation between the flow rate (Q), volume (V), and time to fill the volume (T), is defined by:

$$T = \frac{V}{Q}. \tag{13}$$

When incorporating timing or volume constraints, calculating the volume of each feature is crucial. Components have user-defined dimensions and static positions, making volume calculation straightforward. The total volume v_i of a channel c_i is the sum of the volumes of its individual segments. The volume of a segment ($seg_{i,k}$) is simply the product of the length $l_{seg_{i,k}}$, width $w_{seg_{i,k}}$, and the user-defined layer height h_{layer}.

The length of a segment is further defined by the distance between the two endpoints of the segment. As previously explained, a segment has four orientation options represented by four binary variables. The following constraints enforce these orientations:

$$x2_{b_{i,k}} - x_{b_{i,k}} \leq (d^{0}_{seg_{i,k}} + d^{45}_{seg_{i,k}})M, \tag{14}$$

$$y2_{b_{i,k}} - y_{b_{i,k}} \leq (d^{45}_{seg_{i,k}} + d^{90}_{seg_{i,k}})M, \tag{15}$$

$$|x2_{b_{i,k}} - x_{b_{i,k}} - y2_{b_{i,k}} + y_{b_{i,k}}| \leq (d^{0}_{seg_{i,k}} + d^{90}_{seg_{i,k}})M, \tag{16}$$

$$|ly_{seg_{i,k}} - ly_{seg_{i,k+1}}| \leq d^{via}_{seg_{i,k}} M, \tag{17}$$

$$d^{0}_{seg_{i,k}} + d^{45}_{seg_{i,k}} + d^{90}_{seg_{i,k}} + d^{via}_{seg_{i,k}} = 1. \tag{18}$$

Here, M is a very large constant. The final constraint (Eq. 18) guarantees that exactly one of the binary orientation variables is true for each segment. The other four constraints limit the segment's coordinates and layer based on the chosen orientation, where Eq. 14 constrains horizontal/diagonal x-axis extent, Eq. 15 constrains diagonal/vertical y-axis extent, Eq. 16 constrains horizontal/vertical segment length using Manhattan distance, and Eq. 17 constrains via connections between layers.

With these constraints in place, calculating the segment length becomes straightforward:

$$\begin{aligned} l_{seg_{i,k}} = (x2_{b_{i,k}} - x_{b_{i,k}}) \cdot d^{0}_{seg_{i,k}} + (y2_{b_{i,k}} - y_{b_{i,k}}) \cdot \\ d^{90}_{seg_{i,k}} + \sqrt{2} \cdot (x2_{b_{i,k}} - x_{b_{i,k}}) \cdot d^{45}_{seg_{i,k}}. \end{aligned} \tag{19}$$

This equation considers the segment's x and y extents based on the chosen orientation and accounts for the additional distance traveled in the diagonal case using the square root of 2.

Building upon our flow paths analysis from Section 5.1, an equality constraint can be imposed. For a node with two converging flow paths, the nodes and edges belonging to one path form set A, while those of the other path form set B. The equality constraint ensures that the ratio of volume to flow rate remains consistent across both paths. This is mathematically expressed as:

$$|\frac{\sum_{a \in A} v_a}{Q_A} - \frac{\sum_{b \in B} v_b}{Q_B}| \leq \epsilon \tag{20}$$

Here, Q_A and Q_B represent the flow rates associated with sets A and B, respectively, and ϵ represents the acceptable error tolerance in the volume-to-flow rate ratio. Corresponding volume offsets can be added to each fraction within the constraint if a timing shift or certain volume ratio is desired. For example, a desired timing shift can be implemented by adding a volume offset based on the flow rate and the time difference (with Eq. 13). Similarly, a specific volume ratio can be achieved by adjusting the volume offsets (as fractions of the destination component's volume) accordingly.

5.5 Optimization Formulation

As discussed earlier, a key objective in 3D-printed microfluidic chip design is minimizing chip volume. Smaller chips translate to lower material costs and reduced fabrication time. Furthermore, minimizing the total routed channel length within the chip directly benefits overall chip performance. Shorter channels lead to lower hydraulic resistance in the flow paths, consequently reducing the energy consumption of the external fluid pumping system.

Therefore, the overall constrained optimization model for this design synthesis problem can be defined as:

$$\begin{aligned} \text{Minimize: } & w_{chip} \cdot l_{chip} \cdot nl_{chip} + \\ & \sum_{i=1,\ldots,m} \sum_{k=1,\ldots,n_i} l_{seg_{i,k}} + \\ & |ly_{seg_{i,k}} - ly_{seg_{i,k+1}}| \cdot (h_{layer} + \Delta_{layer}), \\ \text{Subject to: } & (1) - (12) \text{ and } (14) - (20). \end{aligned}$$

Here, h_{layer} is a user-defined constant representing the height of the feature layer, Δ_{layer} is a user-defined distance between two layers, m is the number of channels, and n_i is the number of segments of the channel c_i.

6 EXPERIMENTAL RESULTS

We implemented the 3M-DeSyn method in Java and evaluated its performance on an Apple Silicon M1 8-core (4 performance + 4

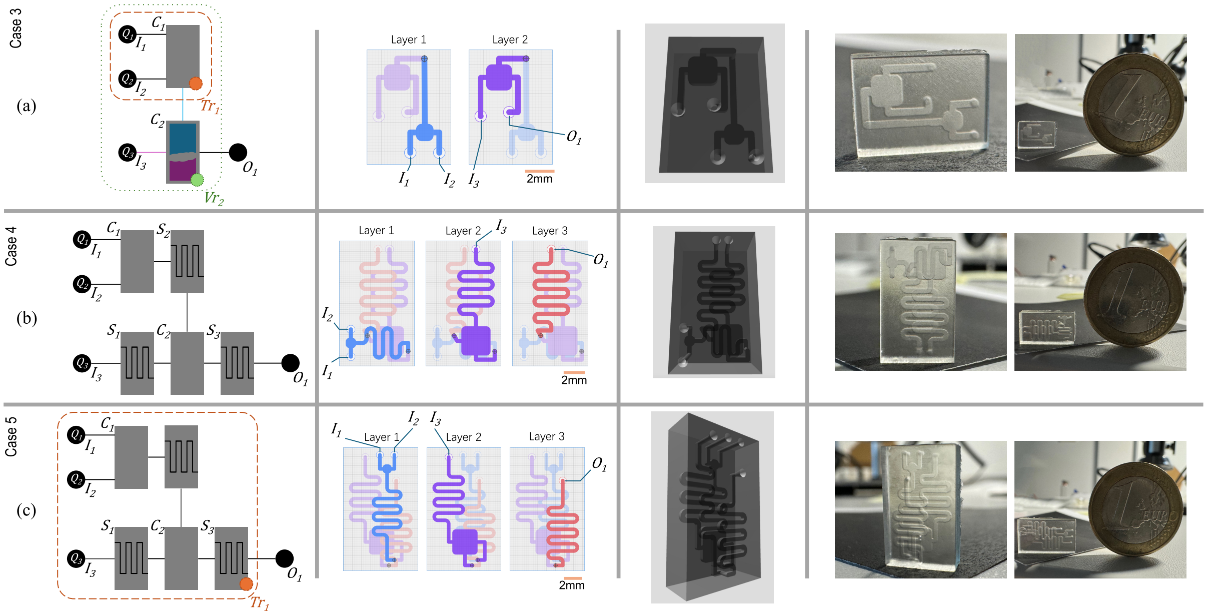

Figure 4: Three test cases for the 3M-DeSyn microfluidic design synthesis method. The left column shows the schematic representations of each test case. I, O, C and S stand for inlet, outlet, chamber and serpentine channel, respectively. The dashed lines and dotted lines represent timing (Tr) and volumetric (Vr) constraints, respectively. The component with a corresponding colored dot indicates the destination component of Tr or Vr, where the fluids must arrive either within a specified time frame or according to a specified volume ratio. The center-left and center-right columns depict the synthesized physical layouts and the corresponding 3D-printable file representations, respectively. The right column shows photos of the 3D-printed microfluidic chips alongside a one-euro coin for size comparison.

efficiency) MacBook Pro with 16 GB of memory. The constrained optimization model was solved using Gurobi Optimizer [5]. To demonstrate the effectiveness of the 3M-DeSyn method, we designed five test cases with varying requirements, including timing, volumetric, and multi-layer constraints. We compared the performance of our method in these test cases with a baseline method.

6.1 Baseline Method

Currently, there is a lack of design automation tools capable of synthesizing layouts for multilayered, 3D-printed microfluidics, nor do existing tools consider timing and volumetric constraints within the design process. While Flui3d represents a state-of-the-art design automation tool for 3D-printed microfluidics, it primarily focuses on accelerating the design process using a predefined 3D component library. Consequently, for our baseline method, we employed a manual design approach utilizing Flui3d's functionalities for component placement and channel routing within a multilayered structure. As with conventional microfluidic chip design workflows, any timing and volumetric constraints must be considered and calculated manually. These calculations often involve factors such as channel dimensions, component positioning, and other relevant parameters.

Since the design workflow of microfluidic devices for 3D printing may vary from engineer to engineer, we abstract the design workflow that employs Flui3d into several distinct stages: (1) review and understand the schematics, (2) assess the timing and volumetric requirements, (3) research the underlying architecture, including deciding how many layers to design, (4) create an initial multi-layer layout (preliminary component placement), (5) route channels in alignment with the schematics (preliminary channel routing), (6) determine channel dimensions based on the preliminary layout, timing, and volumetric requirements, and (7) adjust placement and routing (placement and routing). The stages from (4) to (7) may need to be iterated multiple times until the final design meets all the requirements. To estimate the required time T (in minutes) to design each test case, we developed metrics for each design stage:

(1) $T_{schem} = 0.5 \cdot \#_{comp} + 0.5 \cdot \#_{interrel}$,
(2) $T_{timing\&volume} = 2(\#_{treq} + \#_{vreq}) + 3(\#_{treq} \cdot \#_{vreq}) \cdot \#_{comp} \cdot \#_{interrel}$,
(3) $T_{architecture} = 5(\#_{comp} + \#_{interrel})$,
(4) $T_{pre-placement} = 5 \cdot \#_{comp}$,
(5) $T_{pre-routing} = 5 \cdot \#_{interrel}$,
(6) $T_{dimcalc} = 2(\#_{comp} + \#_{interrel}) + 2(\#_{treq} + \#_{vreq}) \cdot \#_{comp} \cdot \#_{interrel}$,
(7) $T_{p\&r} = 2(\#_{comp} + \#_{interrel} + \#_{treq} + \#_{vreq})$,

where $\#_{comp}$ and $\#_{interrel}$ denote the number of components and inter-relationships, respectively, and $\#_{treq}$ and $\#_{vreq}$ denote the number of timing and volumetric requirements, respectively.

To estimate the total time required to design a 3D-printed microfluidic device with the help of Flui3d, we assume each case requires three iterations of stages (4) to (7). Thus, we estimate the total manual design time $T_{mandesign}$ by

$$\begin{aligned} T_{mandesign} = {} & T_{schem} + T_{timing\&volume} + T_{architecture} + \\ & 3(T_{pre-placement} + T_{pre-routing} + T_{dimcalc} + T_{p\&r}). \end{aligned} \tag{21}$$

Equation 21 provides an estimation of the time required for the manual design that we use as the baseline. In reality, the time and design iterations required for a chip design are usually higher.

Table 1: Experimental results of our proposed method. $\#_{comp}$, $\#_{interrel}$, $\#_{layer}$, $\#_{treq}$, $\#_{vreq}$ and $\#_{chbnd}$ denote the number of components, inter-relationships, layers, timing requirements, volumetric requirements, and channel bends, respectively. L_{ch} denotes the total length of routed channels. $T_{mandesign}$ and $T_{3M-DeSyn}$ denote the time required to design the test case employing manual design and 3M-DeSyn, respectively.

Case	$\#_{comp}$	$\#_{interrel}$	$\#_{layer}$	Area	$\#_{treq}$	$\#_{vreq}$	$\#_{chbnd}$	L_{ch}	$T_{mandesign}$	$T_{3M-DeSyn}$	Result
1	7	5	1	$6000\mu m \times 6000\mu m$	2	0	9	$14413\mu m$	13h46m00s	00h00m13s	Fig. 1 (a)
2	4	3	1	$4250\mu m \times 7232\mu m$	0	1	4	$9063\mu m$	05h07m30s	00h00m02s	Fig. 1 (b)
3	7	5	2	$5775\mu m \times 7900\mu m$	1	1	8	$17023\mu m$	20h54m00s	00h00m35s	Fig. 4 (a)
4	9	8	3	$7200\mu m \times 12000\mu m$	0	0	10	$12466\mu m$	08h07m30s	00h02m16s	Fig. 4 (b)
5	9	8	3	$8200\mu m \times 13653\mu m$	1	1	12	$22108\mu m$	13h51m30s	00h03m40s	Fig. 4 (c)

6.2 Comparison: 3M-DeSyn to Baseline

We compared the performance of our method to the baseline design method. The leftmost column of Figures 1 and 4 depict the schematics of the five test cases with their corresponding requirements. Table 1 summarizes the results and includes the number of components to be placed, channels to be routed, the number of layers, the estimated size of the manually routed area required for each layer in the baseline method, and the number of timing and volumetric constraints. The 3M-DeSyn method generates the optimized design layouts based on given constraints. The center two columns of Figures 1 and 4 depict the resulting layouts and 3D-printable files for the five test cases.

To further demonstrate the practicality of the 3M-DeSyn method, we printed the output files using a hobby 3D printer – Anycubic Photon D2, employing a hobby resin – Anycubic Plant-Based + Clear. The printed results for all five test cases are shown in the rightmost column in Figures 1 and 4.

It is worth noting that 3M-DeSyn offers significant speed up in terms of design time compared to the baseline method. Whereas baseline methods may require hours to design microfluidic chips that meet challenging design constraints, our method produces comparable results in minutes or even within seconds.

7 CONCLUSION

Microfluidic devices have revolutionized various scientific and engineering fields by enabling miniaturization and automation of complex processes. Emerging 3D printing offers a compelling alternative fabrication method, providing rapid prototyping capabilities and enabling intricate geometries unattainable with traditional methods. However, the design of microfluidic lab-on-a-chip systems remains a time-consuming and error-prone process, often reliant on specialized knowledge and manual CAD modeling. Existing design automation tools fall short in this regard, primarily focusing on 2D layouts or lacking integrated design synthesis capabilities.

Timing and volume constraints are crucial aspects to consider when designing microfluidic devices, as they directly impact factors like reaction times and mixing ratios. Unfortunately, none of the existing design automation tools consider these aspects.

This work addresses these limitations by introducing the 3M-DeSyn method, a novel design synthesis method for 3D-printed microfluidics. The 3M-DeSyn method automates the design process by directly synthesizing microfluidic layouts, considering critical timing and volumetric requirements for multilayered 3D-printed microfluidics. As demonstrated through the test cases, 3M-DeSyn significantly reduces design time and facilitates the creation of printable microfluidic layouts that adhere to design requirements.

REFERENCES

[1] Anthony K Au, Nirveek Bhattacharjee, Lisa F Horowitz, Tim C Chang, and Albert Folch. 2015. 3D-printed microfluidic automation. *Lab on a Chip* 15, 8 (2015), 1934–1941.

[2] Simon F Berlanda, Maximilian Breitfeld, Claudius L Dietsche, and Petra S Dittrich. 2020. Recent advances in microfluidic technology for bioanalysis and diagnostics. *Analytical chemistry* 93, 1 (2020), 311–331.

[3] Ursula Bilitewski, Meike Genrich, Sabine Kadow, and Gaber Mersal. 2003. *Biochemical analysis with microfluidic systems.* Vol. 377. Springer. 556–569 pages.

[4] Krishnendu Chakrabarty and Fei Su. 2018. *Digital microfluidic biochips: synthesis, testing, and reconfiguration techniques.* CRC press.

[5] Gurobi Optimization, LLC. 2023. Gurobi Optimizer Reference Manual. https://www.gurobi.com

[6] Skarphedinn Halldorsson, Edinson Lucumi, Rafael Gómez-Sjöberg, and Ronan MT Fleming. 2015. Advantages and challenges of microfluidic cell culture in polydimethylsiloxane devices. *Biosensors and Bioelectronics* 63 (2015), 218–231.

[7] Zuoyan Han, Wentao Li, Yanyi Huang, and Bo Zheng. 2009. Measuring rapid enzymatic kinetics by electrochemical method in droplet-based microfluidic devices with pneumatic valves. *Analytical chemistry* 81, 14 (2009), 5840–5845.

[8] David Holmes and Shady Gawad. 2010. The application of microfluidics in biology. *Microengineering in biotechnology* (2010), 55–80.

[9] Anna V Nielsen, Michael J Beauchamp, Gregory P Nordin, and Adam T Woolley. 2020. 3D printed microfluidics. *Annual Review of Analytical Chemistry* 13 (2020), 45–65.

[10] Jose L Sanchez Noriega, Nicholas A Chartrand, Jonard Corpuz Valdoz, Collin G Cribbs, Dallin A Jacobs, Daniel Poulson, Matthew S Viglione, Adam T Woolley, Pam M Van Ry, Kenneth A Christensen, et al. 2021. Spatially and optically tailored 3D printing for highly miniaturized and integrated microfluidics. *Nature communications* 12, 1 (2021), 5509.

[11] Radhakrishna Sanka, Joshua Lippai, Dinithi Samarasekera, Sarah Nemsick, and Douglas Densmore. 2019. 3DμF-interactive design environment for continuous flow microfluidic devices. *Scientific reports* 9, 1 (2019), 1–10.

[12] Helen Song, Joshua D Tice, and Rustem F Ismagilov. 2003. A microfluidic system for controlling reaction networks in time. *Angewandte Chemie* 115, 7 (2003), 792–796.

[13] Tsun-Ming Tseng, Mengchu Li, Yushen Zhang, Tsung-Yi Ho, and Ulf Schlichtmann. 2019. Cloud columba: Accessible design automation platform for production and inspiration. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD).* IEEE, 1–6.

[14] Jingyi Wang, Carlton McMullen, Ping Yao, Niandong Jiao, Min Kim, Jin-Woo Kim, Lianqing Liu, and Steve Tung. 2017. 3D-printed peristaltic microfluidic systems fabricated from thermoplastic elastomer. *Microfluidics and Nanofluidics* 21 (2017), 1–13.

[15] Kevin Ward and Z Hugh Fan. 2015. Mixing in microfluidic devices and enhancement methods. *Journal of Micromechanics and Microengineering* 25, 9 (2015), 094001.

[16] Alireza Ahmadian Yazdi, Adam Popma, William Wong, Tammy Nguyen, Yayue Pan, and Jie Xu. 2016. 3D printing: an emerging tool for novel microfluidics and lab-on-a-chip applications. *Microfluidics and Nanofluidics* 20 (2016), 1–18.

[17] Yushen Zhang, Mengchu Li, Tsun-Ming Tseng, and Ulf Schlichtmann. 2024. Open-source interactive design platform for 3D-printed microfluidic devices. *Communications Engineering* 3, 1 (2024), 71.