Efficient Sample Preparation with Fully Programmable Valve Arrays

Abhik Kumar Khan*, Sudip Roy†, Bhargab B. Bhattacharya‡ and Sukanta Bhattacharjee*
*Indian Institute of Technology (IIT) Guwahati, India, †IIT Roorkee, India, ‡Indian Statistical Institute Kolkata, India

***Abstract*—The 2D architecture of fully-programmable valve arrays (FPVAs) is designed as a crossbar consisting of reaction chambers and microvalves, functioning as a versatile, flow-based microfluidic lab-on-chip for implementing biochemical protocols. While an FPVA enables efficient execution of various fluidic operations—such as mixing, loading, and storage, transporting fluids between chambers remains a challenging task. Furthermore, mapping a general mixing tree (representing a sequence of mixing steps) onto an FPVA is complex. It requires careful placement of reagents into specific chambers and the scheduling of subsequent mixing operations. Most sample preparation algorithms aim to generate a minimum-depth mixing tree to achieve the target mixing ratio. However, due to constraints on fluid transportation and scheduling, such a tree may not be the most practical for FPVA implementation. In this paper, we harness the power of a satisfiability solver to derive a skewed mixing tree/graph that can be efficiently mapped onto an FPVA using a single mixer. This approach localizes most fluidic operations to a small region of the crossbar. Simulation results show that, for most mixing ratios, a skewed mixing tree can be found which not only reduces fluid-transportation distance and scheduling complexities but also the number of loading cycles, reagent volumes, and waste production in sample preparation, when compared to the approach based on the minimum-depth mixing tree.**

I. Introduction

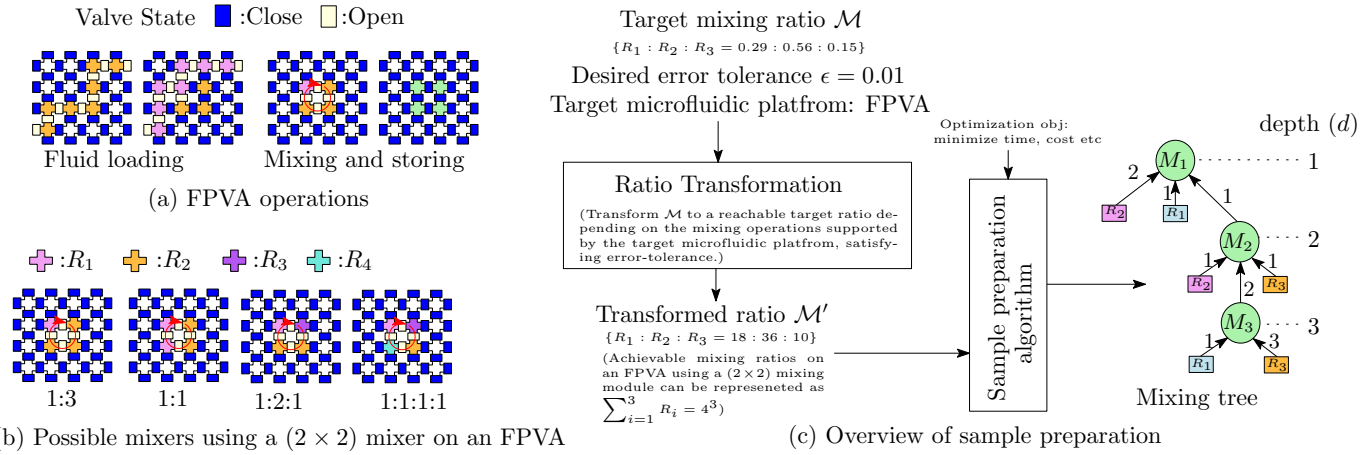
Microfluidic biochips represent a groundbreaking advancement in biomedical technology, enabling the manipulation of fluid volumes at the nano- to pico-liter scale in a programmable manner. These chips are capable of integrating multiple laboratory functions into a compact, miniaturized platform, often referred to as a Lab-on-a-Chip (LoC) [1]. By offering a faster, more cost-effective, and reliable alternative to traditional benchtop methods, LoCs have made a significant impact in areas such as point-of-care (POC) medical diagnostics [2, 3], DNA amplification [4, 5], and biomedical research [6].

These platforms are typically categorized into two main types: Digital Microfluidic Biochips (DMBs) and Flow-based Microfluidic Biochips (FMBs). DMBs manipulate individual droplets using electrical actuators on a two-dimensional array of electrodes [7], while FMBs control fluid flow within a network of microchannels via pressure-driven microvalves [8]. A notable development in FMBs is the emergence of general-purpose Fully Programmable Valve Arrays (FPVAs), which consist of a two-dimensional array of fluid chambers, each surrounded by four microvalves [9, 10]. FPVAs can perform a variety of fluidic operations—including fluid loading, mixing, and storage—by configuring the valve states (ON for closed, OFF for open) in a reconfigurable manner (Fig. 1(a)). Numerous biochemical assays, such as cell culture and surface immunoassays, have been demonstrated on FPVAs [9].

Sample preparation involves mixing two (i.e., dilution) or more (i.e., mixing) fluids in specific ratios. Programmable microfluidic platforms, such as DMBs and FMBs, deploy mixing modules to combine fluids in various proportions (Fig. 1(b) shows possible configurations of a (2×2) mixer on an FPVA). Sample preparation on biochips is carried out by executing a sequence of mixing operations, typically represented as a mixing tree [11]. This process requires the mixing of fluids (input reagents and intermediate fluids generated during the process) in different volumetric proportions (depending on the mixing ratios supported by the LoC platform), storage of intermediate fluids after subsequent mixing, and transportation of fluids between storage and mixers. One key optimization goal is to minimize the number of mixing operations (nodes in the tree), as this directly impacts sample-preparation time and resource usage. Another objective is to reduce the consumption of input fluids, which affects the overall cost. Fig. 1(c) shows an overview of sample preparation with programmable microfluidic biochips.

While FPVAs offer the flexibility to create reconfigurable mixers for fluid manipulation by adjusting valve configurations, they face a key limitation: they cannot transport intermediate fluids generated during mixing between cells, meaning they are restricted to No-Transport-Mixing (NTM [12]). An algorithm was proposed to find the scheduling and placement of mixing operations in the mixing tree without any fluid transportation between FPVA chambers [12]. Input reagents are usually assigned to their respective mixing chambers and are loaded in a specific order by pushing fluids so as to minimize the number of loading cycles [13]. The algorithm also reduces the number of valve-actuations by reducing the fluid loading cycles (K) and the length of loading paths (L). The complexity of a loading path, which is measured by its length and the number of 90-degree bends (B), impacts fluid-flow stability and chip reliability [14]. Thus, algorithms on FPVAs should aim to find simpler paths while loading fluids.

The complexity of sample-preparation algorithms on FPVA (measured in terms of (K, B, L)-parameter values) depends on the mixing tree generating the target ratio [13]. Note that for a given mixing ratio and error-tolerance, there exist multiple valid ratios which are equivalent within the error-tolerance (ref Sec. III-B for formal explanation). Existing sample-preparation algorithms (e.g., FloSPA [15]) choose a



(a) FPVA operations

(b) Possible mixers using a (2×2) mixer on an FPVA

(c) Overview of sample preparation

Fig. 1: (a) FPVA operations: loading of two reagents R_1 and R_2, mixing R_1 and R_2 in 1:3 ratio, and Storage of fluids after mixing. (b) Possible mixer configurations of a (2×2) mixer on an FPVA. (c) Overview of sample preparation with programmable microfluidic biochips.

valid mixing ratio that minimizes the depth of the mixing tree. However, the minimum-depth mixing tree may not be suitable for sample preparation with FPVA. This is mainly due to the inherent limitation of the transportation of fluids between two FPVA chambers. We have found that using a skewed mixing tree, rather than a min-depth tree, reduces automation overhead on an FPVA. A skewed mixing tree features a single primary stem, with all successor nodes acting as leaf nodes, resembling a caterpillar [16]. This structure simplifies the mixer-placement process and is well-suited for FPVAs, as it allows mixers to be stacked vertically, with each placed directly atop its predecessor. This configuration removes the need for complex placement algorithms, streamlining the execution of the sample-preparation process. To determine an equivalent mixing ratio achievable with a skewed mixing tree, we employed an SMT-based search technique. We performed an extensive simulation and found that for most of the target ratios, a skewed tree can be found with a nominal increase of the depth. Moreover, a skewed tree significantly reduces the automation overhead and performs sample preparation faster and cheaper compared to a minimum-depth mixing tree used in existing FPVA-based sample preparation.

The rest of the paper is organized as follows: Section II discusses the basics of sample preparation and its implementation on an FPVA. Section III explains the motivation behind our approach and details the proposed method. Section IV presents the simulation results, and Section V concludes the paper.

II. Background

A. Basics of Sample Preparation

Sample preparation is a process of mixing two (dilution) or more (mixture) input reagents in a given volumetric ratio. In sample preparation, a target ratio of k reagents is given as $\mathcal{M} = \{R_1 : R_2 : \cdots : R_k = x_1 : x_2 : \cdots : x_k\}$, where $0 \leq x_i \leq 1$ and $\sum_{i=1}^{k} x_i = 1$ [17, 18]. Depending on the mixer-type supported by the target microfluidic platform (e.g., a (2×2)- mixer on the FPVA can mix reagents in (1:3), (1:1), (1:2:1), and (1:1:1:1) ratios), the ratio $\mathcal{M}$ needs to be transformed into another equivalent ratio that can be realized using a sequence of mixing operations on the target microfluidic platform. For an FPVA (supporting only (2×2) mixer), the target ratio $\mathcal{M}$ is transformed as $\mathcal{M}' = \{R_1 : R_2 : \cdots : R_k = y_1 : y_2 : \cdots : y_k\}$, where $y_i \in \{0, 1, 2, \cdots, 4^d\}$ for $i = 1, 2, \cdots, k$, and $\sum_{i=1}^{k} y_i = 4^d$. Note that d is a positive integer and depends on user-given error tolerance $\epsilon \in (0, 1)$ satisfying $|\frac{y_i}{\sum_{j=1}^{k} y_j} - x_i| \leq \epsilon$ for $i = 1, 2, \cdots, k$ i.e., $\max_i\{|\frac{y_i}{\sum_{j=1}^{k} y_j} - x_i|\} \leq \epsilon$ (*a.k.a.* ratio-validity condition). The following example illustrates the ratio transformation.

Example 1. *Consider a mixing ratio* $\mathcal{M} = \{R_1 : R_2 : R_3 = 0.18 : 0.65 : 0.17\}$ *that needs to be realized on an FPVA biochip supporting* 2×2 *mixer. For the given error tolerance* $\epsilon = 0.01$*, the mixing ratio* $\mathcal{M}$ *is transformed as* $\mathcal{M}' = \{R_1 : R_2 : R_3 = 12 : 41 : 11\}$*. Notice that* d *is set to three, which is the minimum satisfying* $max\{|\frac{12}{64} - 0.18|, |\frac{41}{64} - 0.65|, |\frac{11}{64} - 0.17|\} \leq 0.01$.

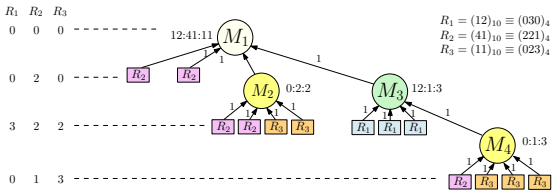


Fig. 2: Mixing tree generated with genMixing for $\{12 : 41 : 11\}$

For the transformed ratio $\mathcal{M}'$, the sample-preparation algorithm genMixing [15] represents each y_i as d-digit base-4 number i.e., $(y_i)_{10} = (a_{d-1}^i a_{d-2}^i \cdots a_1^i a_0^i)_4$. Next, these k d-digits numbers are scanned from left-to-right to construct a mixing tree in a bottom-up fashion. The value of d determines the depth of the mixing tree. For each non-zero digit a_j^i in the base-4 representation of y_i, a_j^i units of input reagent R_i are fed as input to the mixer. An internal (leaf) node in the mixing tree represents a mixing operation (input reagent), and an edge represents a unit volume of fluid shared between the nodes. Fig. 2 shows the mixing tree generated using genMixing [15] for the mixing ratio $\mathcal{M}' = \{R_1 : R_2 : R_3 = 12 : 41 : 11\}$.

FloSPA [15, 19] is a sample-preparation algorithm that

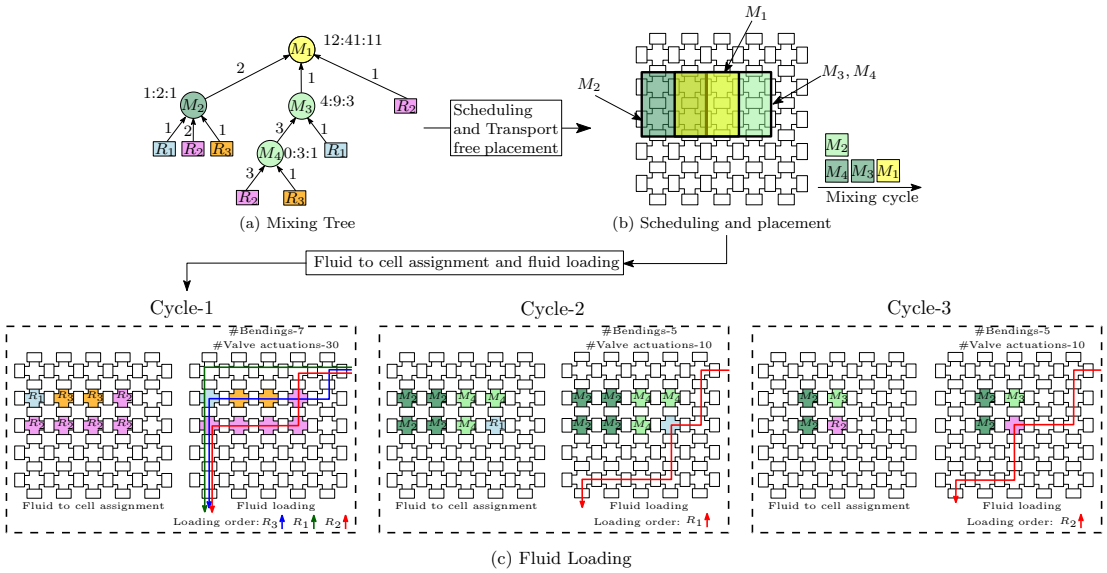


Fig. 3: Overview of sample preparation on FPVA: (a) mixing tree, (b) scheduling and placement of mixing operations. (c) For each mixing cycle, reagents are assigned to FPVA chambers, and fluid loading is performed by creating a sequence of input-to-output paths. After loading, mixing operations are performed.

supports a generalized mixing model, making it especially well-suited for flow-based biochips. It employs a rotary mixer and is capable of handling both dilution (mixing two fluids, such as a sample and a buffer) and more complex generalized mixing tasks (involving multiple fluids in specific ratios). The core optimization framework of FloSPA leverages Satisfiability Modulo Theory (SMT) to generate efficient solutions in terms of reagent cost and sample-preparation time. Fig. 3(a) shows the mixing tree generated by FloSPA for the mixing ratio $\mathcal{M}'$.

B. Design Automation of Sample Preparation on FPVA

Several steps are required to realize a mixing tree on an FPVA: mixer placement and scheduling of mixing operations, reagent assignment, and loading. Fig. 3 shows the various steps for sample preparation on an FPVA. As discussed earlier, an FPVA must adhere to No-Transport-Mixing (NTM) [12]. Fig. 3(b) shows the placement and scheduling of mixing operations for the mixing tree (Fig. 3(a)) that satisfy the constraints of NTM. The entire mixing process can be completed in three cycles. The execution of mixers M_2 and M_4 can be scheduled in parallel since they appear as disjoint subtrees of the mixing tree.

After scheduling and placing the mixing operations, the next step is to load the input fluids into the designated chambers before mixing. To do this, an input-to-output path must be created by adjusting the valve states along the path. As a result, the fluid loading time is directly proportional to the length of the path. Additionally, the amount of input reagents—essentially, the cost of sample preparation—depends on both the number of flows and the complexity of the flow-path, particularly the number of 90° bends. Since flow pressure drops significantly with an increase in 90° bends [14], reagent-usage during each loading cycle also increases. Therefore, it is important to assign reagents to chambers in a way that minimizes the number of flows (K) required for each mixing cycle.

To achieve this, a constraint satisfiability-based algorithm, called the Loading-Aware Fluid-to-Cell Assignment (LAFCA) [13], was proposed. This algorithm binds reagents to appropriate chambers in the mixers, ensuring that identical reagents can be loaded with a single flow. Once this is done, the Deterministic-Flow-from-the-Last (DFL) algorithm [13] is used to determine the optimal reagent loading path, minimizing both path-length and the number of bends. Fig. 3(c) illustrates the fluid-to-chamber assignment and fluid-loading paths for each mixing cycle. Note that in the first mixing cycle, input reagents corresponding to M_2 and M_4 are assigned to mixer chambers. Moreover, three loading cycles are required to load reagents in the order of R_3, R_1, and R_2, respectively, to minimize path length and the number of bends.

III. Proposed Method

A. Motivation of this work

The complexity of sample-preparation on an FPVA depends on the structure of the mixing tree generating the target ratio. More specifically, with the increase in the number of branch nodes (nodes having multiple mixing nodes as its children) in the mixing tree, the number of FPVA chambers used to place mixing modules (*a.k.a* footprint) increases. Consequently, the complexity of the fluid-loading path (measured in terms of path-length and the number of 90^o turns), and the number of fluid-loading cycles also increase. Therefore, a skewed mixing tree (without any branch node) is preferred for translation onto an FPVA. However, given a target ratio and a specified error-tolerance, most existing sample-preparation algorithms produce a minimum-depth mixing tree, which often is not found to be skewed.

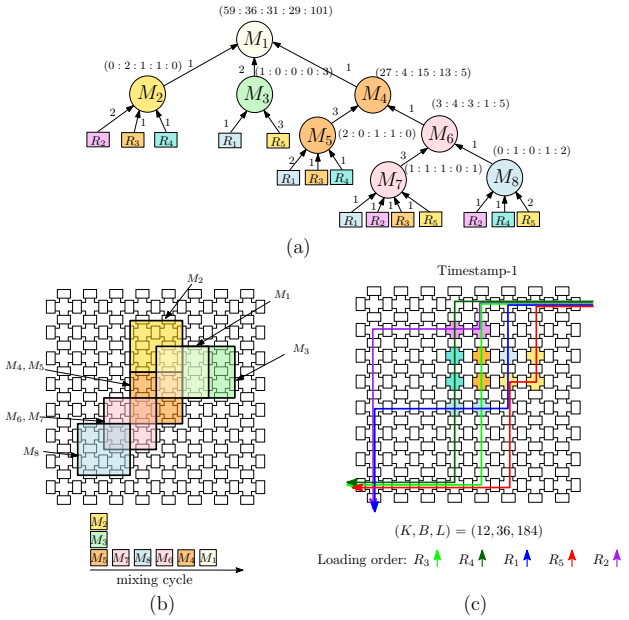


Fig. 4: (a) Mixing tree for the target ratio $\{59 : 36 : 31 : 29 : 101\}$. (b) Transport-free placement and scheduling of all mixing operations in the mixing tree. (c) Fluid to chamber assignment, loading order and the loading path for Mixing-Cycle 1. The complete procedure is illustrated in additional document Fig. 2(a-c) (Appendix I)

.

Example 2. *Consider the target mixing ratio* $\mathcal{M} = \{R_1 : R_2 : R_3 : R_4 : R_5 = 0.23 : 0.136 : 0.124 : 0.119 : 0.391\}$ *with error tolerance* $\epsilon = 0.0058$*. The existing algorithm transforms it to the mixing ratio* $M^{'} = \{R_1 : R_2 : R_3 : R_4 : R_5 = 59 : 36 : 31 : 29 : 101\}$*, by choosing minimum* d*. The mixing tree generated by FloSPA for realizing* $\mathcal{M}'$ *is shown in Fig. 4(a). Note that a skewed tree realizing the target ratio* $\mathcal{M}$ *does not exist for* $d = 4$*. Fig. 4(b) shows scheduling and placement information as mandated by NTM. The total number of chambers (i.e., footprint) required to place the mixing tree on the FPVA biochip is substantial. Note that 18 chambers are required for mixer placement. Also, in the first mixing cycle,* M_2, M_3*, and* M_5 *are scheduled (Fig. 4(c)), blocking the fluid-loading path in subsequent mixing cycles, leading to more complex fluid loading.*

Note that for $d = 4$, 51 mixing ratios are possible for the target ratio $\mathcal{M}$ given in Example 2 satisfying the ratio-validity condition. However, none of them produces a skewed mixing tree. If we increase d, there exist multiple mixing ratios $\mathcal{M}^{'}$ satisfying the ratio-validity condition. The following example shows how a nominal increase in d facilitates the quest for a skewed mixing tree for a given target ratio.

Example 3. *Consider the mixing ratio* $\mathcal{M}$ *given in Example 2. For* $d = 5$*,* $10,056$ *mixing ratios satisfy the ratio-validity condition, and for* $d = 6$*, the number of mixing ratios exceeds half a million. A skewed mixing tree can be found at* $d = 6$ *for the transformed mixing ratio* $\mathcal{M}^{''} = \{R_1 : R_2 : R_3 : R_4 : R_5 = 930 : 576 : 486 : 504 : 1600\}$ *satisfying the ratio-validity condition. Fig. 5 shows the skewed mixing tree along with the placement and scheduling information. It can*

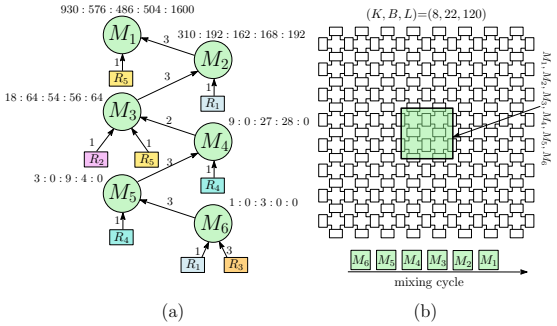


Fig. 5: (a) Skewed mixing tree found at depth 6 for the target ratio $\mathcal{M} = \{0.23 : 0.136 : 0.124 : 0.119 : 0.391\}$ and $\epsilon = 0.0058$. (b) Transport-free placement and mixing sequence of all mixers in the mixing tree generated by NTM [12]. All steps are illustrated in additional document Fig. 1(a-b) (Appendix I)

.

be observed that mixing trees for both $d = 4$ *(Fig. 4(a)) and the skewed one for* $d = 6$ *(Fig. 5(a)) require six mixing cycles. However, only four chambers are required to place all mixing operations corresponding to the skewed tree. To realize the mixing ratio based on the minimum-depth mixing tree (Fig. 4(a)), we require a total of 12 flows-instances for reagent loading, 184 valve-actuations and 36 bends in flow-paths. However, when the skewed mixing tree is used we require only 8 flows and 120 valve-actuations, also in all flow paths, there are 22 bendings.*

B. Proposed Method for Finding a Skewed Tree

In order to find a skewed mixing tree for a given target ratio, we leverage underlying caterpillar-like structure in our satisfiability modulo theory (SMT) formulation. For a given target ratio $\mathcal{M} = \{R_1 : R_2 : \cdots : R_k = x_1 : x_2 : \cdots : x_k\}$, where $x_i \in (0, 1), \sum_{i=1}^{k} x_i = 1$, and the given desired error-tolerance, the proposed SMT formulation checks whether there exists a valid target ratio $\mathcal{M} = \{R_1 : R_2 : \cdots : R_k = y_1 : y_2 : \cdots : y_k\}$ where $y_i \in \{1, 2, \cdots, 4^d - 1\}$ and $\sum_{i=1}^{k} y_i = 4^d$, at depth d, which will lead to a skewed mixing tree.

The proposed SMT-based approach constructs a skeleton tree with depth d as shown in Fig. 6(a), and then augments the tree with leaf nodes representing input reagents that are needed during mixing. Additional variables are used as required to represent intermediate mixing ratios produced at each mixing node, the usage of reagents, and possible sharing of fluids for mixing operations. The skeleton tree after augmentation is shown in Fig. 6(b). Furthermore, some constraints are added to ensure the correctness of mixing ratios generated at each mixing node. We describe the proposed SMT-based modeling as follows:

- R_j^i: (Node variables) For each mixing node M_i appearing at depth i in the skewed mixing tree, we define k integer variables $R_1^i, R_2^i, \cdots, R_k^i$, to represent the portion of input reagents $R_1, R_2, \cdots, R_k$, respectively, in the intermediate ratio generated by M_i.
- r_j^i: (Reagent variables) In the skewed mixing tree, for each mixing node M_i appearing at depth i, k reagent variables, $r_1^i, r_2^i, \cdots, r_k^i$ are used to represent the

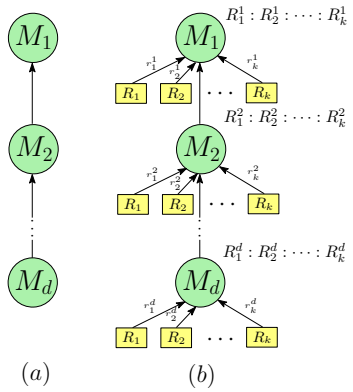


Fig. 6: (a) Skeleton tree of depth d. (b) Augmented skeleton tree.

number of chambers of the mixer to be filled with reagent $R_1, R_2, \cdots, R_k$, respectively. Note that, $r_j^i \in \{0, 1, \cdots, 4\}$, for $i = 1, 2, \cdots, d$ and $j = 1, 2, \cdots, k$.

$W_{j,i}$: (Intermediate fluid sharing variables) The mixing node appearing at depth i may use the intermediate fluid generated by the mixing node appearing at depth j. We use an integer variable $W_{j,i}$ to represent such sharing of fluids between mixing nodes. For skewed mixing tree $j = i + 1$.

To ensure correctness of mixing ratios at each mixing node, we add several constraints, as described below.

Domain constraints: Each mixing node may use at most three units of input reagents. Therefore, $r_j^i \in \{0, 1, 2, 3\}$, for $i = 1, 2, \cdots, d$ and $j = 1, 2, \cdots, k$. Similarly, $W_{i+1,i} \in \{1, 2, 3\}$ for $i = 1, 2, \cdots, d-1$.

Mixer consistency constraints: The following constraints ensure all four chambers of a (2×2) mixer in an FPVA be filled with either input reagents or intermediate fluids from the successor mixing node.

$$\begin{cases} W_{i+1,i} + \sum_{j=1}^{k} r_j^i = 4 & \text{for } i = 1, 2, \cdots, d-1 \\ \sum_{j=1}^{k} r_j^i = 4 & \text{for } i = d \end{cases}$$

Mixing ratio constraints: The following constraints are needed to calculate intermediate mixing ratios for each mixing node.

$$R_j^i = \begin{cases} 4^{d-i} \times r_j^i + W_{i+1,i} \times R_j^{i+1} & \text{if } i \in \{1, 2, \cdots, d-1\} \\ r_j^i & \text{if } i = d \end{cases}$$

Since $r_j^i \in \{0, 1, 2, 3\}$ and $W_{i+1,i} \in \{1, 2, 3\}$, we can easily eliminate the non-linearity from the equations presented above [15]. Furthermore, the following constraints ensure the validity of the mixing ratio generated at the root node.

$$\text{for } j = 1, 2, \ldots, k$$
$$\left(\frac{R_j^1}{\sum_{l=1}^{k} R_l^1} - x_j \leq \epsilon\right) \bigwedge \left(x_j - \frac{R_j^1}{\sum_{l=1}^{k} R_l^1} \leq \epsilon\right)$$

Algorithm 1 outlines the proposed SMT-based approach for identifying a skewed tree that achieves a given target ratio. Given the target ratio, we first compute a transformed ratio for minimum d that satisfies the ratio-validity condition. Next, a skeleton tree with d mixing nodes is constructed (Fig. 6(a)) which is then annotated with reagent nodes (Fig. 6(b)). An SMT instance is generated from this annotated tree by introducing variables and constraints. If the SMT instance is satisfiable, the skewed mixing tree generating the target ratio can be derived from the variable assignment. If no suitable skewed mixing tree or graph is found at the current depth, we increment d by one and continue the search. We set an upper limit of 9 for d to control the search space and limit computational time. If no skewed tree is found for $d \leq 9$, we terminate the search and use the mixing tree generated by the FloSPA [15] for sample preparation. Additionally, when searching for a skewed mixing tree, we incorporate an optimization objective to minimize the number of leaf nodes. This approach reduces the number of flows required for reagent loading, thus minimizing design-automation overheads.

Our experiments demonstrate that skewed mixing graphs (Table I) within this depth range (1 to 9) are successfully identified. Since mixing operations in a skewed mixing tree must be performed sequentially, using a deeper skewed mixing tree increases the sample-preparation time. Therefore, we restrict our search to a maximum depth of $d \leq 2 \times k$, where k is the depth of the mixing tree generated by FloSPA. Our experiments show that all benchmark mixing ratios yield a solution for $d \leq 9$.

Algorithm 1: AdaptiveMixingTree($\mathcal{M}, \epsilon$)

Input: $\mathcal{M} = \{R_1 : R : 2 : \cdots : R_k = x_1 : x_2 : \cdots : x_k\}$: mixing ratio, ϵ: error tolerance
Output: Mixing tree.

Transform $\mathcal{M}$ by choosing minimum l as $\mathcal{M}' = \{R_1 : R_2 : \cdots : R_k = y_1 : y_2 : \cdots : y_k\}$ where $y_i \in \{1, 2, \cdots, 4^{l-1}\}$ and $\sum_{i=1}^{k} y_i = 4^l$ such that $max_i\{|x_i - \frac{y_i}{4^l}|\} \leq \epsilon$;
for $(d = l; d < 10; d = d + 1)$ **do**
  T = Augmented skewed skeleton tree of depth d;
  Φ = Symbolic representation of T;
  /* Φ is the SMT instance obtained by adding variables (node, reagent, and intermediate fluid-sharing) and constraints (domain, mixer consistency, and mixing ratio) for T */
  if Φ *is Satisfiable* **then**
    // Z3 [20] is used for satisfiability checking
    return the skewed mixing tree T_s obtained from the satisfiable assignment;
return the mixing tree generated by FloSPA($\mathcal{M}'$);

C. Sharing Fluids Between Non-adjacent Mixing Nodes

So far, we have focused on the sharing of intermediate fluids between adjacent levels of the skewed mixing tree. However, sharing intermediate fluids between non-adjacent ancestors offers the potential to create a skewed mixing graph with fewer mixing operations, thereby reducing sample-preparation time. Since fluids cannot be transferred between chambers in an FPVA biochip, determining suitable sharing between non-adjacent mixing nodes is not straightforward. We explored possible ways to share intermediate fluids between non-adjacent mixing operations (for a detailed analysis, refer

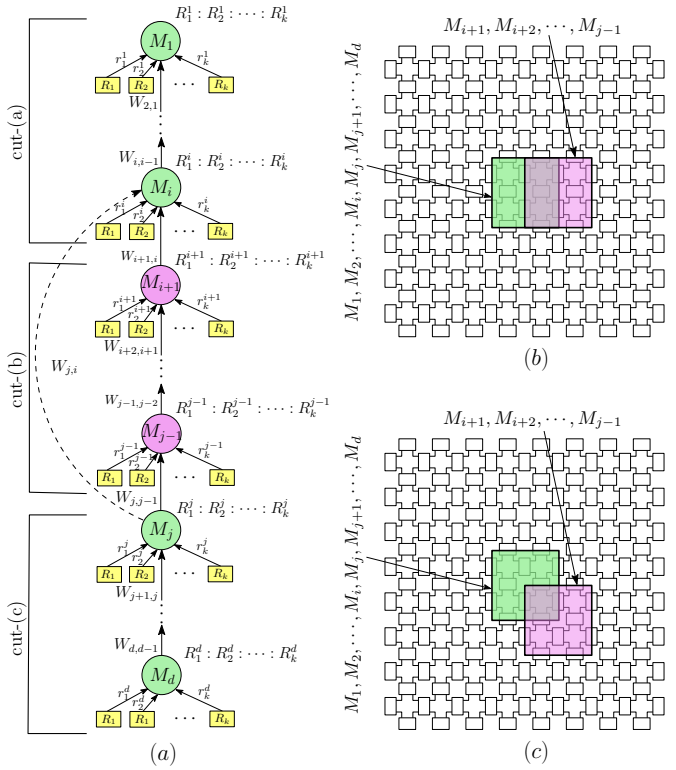


Fig. 7: (a) A skewed mixing graph with non-adjacent sharing of $W_{j,i}$ units from M_j to M_i. (b) The footprint of the skewed mixing graph when $W_{j,i} = 2$. (c) The footprint of the skewed mixing graph when $W_{j,i} = 3$.

to Appendix II in additional document). Among the various allowable fluid-sharing configurations, we focus on a specific type of sharing, referred to as admissible sharing, which is described as follows. Admissible sharing: if a mixing node M_j shares fluids with a grandparent mixing node M_i (i.e., $j > i+1$) in the skewed mixing tree, then mixing nodes $M_{i+1}, M_{i+2}, \cdots, M_{j-1}$ cannot share fluids with any of their grandparents. Note that a mixing graph satisfying admissible sharing criteria can be efficiently mapped onto an FPVA using a constant number of chambers. Moreover, for most of the target mixing ratios, a skewed mixing graph satisfying the admissible sharing condition can be found, having a nominal increase in the depth compared to the minimum-depth mixing tree.

Scheduling and Placement of a Skewed Mixing Graph Satisfying Admissible Sharing: Let us consider a skewed mixing graph shown in Fig. 7(a) where mixing node M_i uses $W_{j,i}$ units of the fluid produced at mixing node M_j, where $j > i+1$. Without loss of generality, we assume sharing between M_i and M_j only. Note that, due to the admissible sharing criteria, each sharing of fluids between more than two pairs of mixing nodes, can be handled similarly. To realize the skewed mixing graph shown in Fig. 7(a) on an FPVA, we cut it into three pieces: (a) from M_1 to M_i, (b) from M_{i+1} to M_{j-1}, and (3) from M_j to M_d. We map each cut on the FPVA separately, as illustrated in Figs. 7(b-c).

To begin with, the mixing nodes in Cut-(c) i.e., $M_d, M_{d-1}, \cdots, M_j$, are placed on a (2×2) mixer. Since $W_{j,i}$ units of M_j needed to be stored, we place mixers in Cut-(b) i.e., $M_{j-1}, M_{j-2}, \cdots, M_{i+1}$, on a different mixer. If $W_{j,i} \leq 2$, we place mixer M_{j-1} to the right as shown in Fig. 7(b). For $W_{j,i} = 3$, we place mixer M_{j-1} at the corner as shown in Fig. 7(c) so that 3 units of M_j can be stored. After the placement of mixer M_{j-1}, we place all the remaining mixers in the cut-(b) i.e., $M_{j-2}, M_{j-3}, \cdots, M_{i+1}$, on the same (2×2) mixer used for M_{j-1}. Since mixer M_i uses $W_{j,i}$ units of fluids generated by the mixer M_j (stored previously), we place the i^{th}-mixer similarly to that of M_j. Finally, we place the remaining mixers in cut-(a) $M_{i-1}, M_{i-2}, \cdots, M_1$ on the same mixer used for M_i. Note that any two mixers sharing three units of fluids should be placed in the same mixer to avoid the transportation of fluids between FPVA chambers. Since we place M_j and M_i on the same mixer, $W_{j,i} \leq 3$. Moreover, for possible nonadjacent sharing, we must ensure $W_{j,j-1} < 3$ and $W_{i+1,i} < 3$.

SMT Modeling: We incorporate necessary modifications in the SMT modeling for finding a skewed mixing tree (Sec. III-B) to enable non-adjacent sharing between mixers. We represent all possible sharing of intermediate fluids by adding directed edges (M_j, M_i), where $j > i+1$, between non-adjacent mixers M_j and M_i in the skeleton tree (Fig. 8(a)) and put additional constraints to ensure the desired sharing of fluids between non-adjacent mixing nodes. Necessary modifications in the SMT modeling are described below.

Non-adjacent fluid sharing and domain constraints: Denote by variables $W_{j,i}$, where $j > i+1$, possible sharing of fluids from M_j to M_i. Moreover, $W_{j,i} \in \{0,1,2,3\}$. The constraint $\sum_{k=1}^{i-1} W_{i,k} \leq 4$ ensures that a mixer shares at most four units of fluids with its ancestors.

Mixer consistency constraint: We need to modify mixer consistency constraints for possible non-adjacent sharing of fluids between mixing modes as follows.

$$\begin{cases} \sum_{l=i+1}^{d} W_{l,i} + \sum_{j=1}^{k} r_j^i = 4 & \text{if } i \in \{1,2,\cdots,d-1\} \\ \sum_{j=1}^{k} r_j^i = 4 & \text{if } i = d \end{cases}$$

Non-adjacent sharing constraint: The following constraints are introduced to handle admissible sharing of fluids between non-adjacent mixing nodes. For simplicity, we define indicator variables $z_{j,i}(1,\ \text{if } W_{j,i} \geq 1; 0,\ \text{otherwise})$ to identify sharing between mixers M_j and M_i. By virtue of admissible sharing, if the intermediate fluid produced in M_j is used by a mixer M_i, the mixers M_i and M_j should be placed in the same location on the FPVA to enable the sharing. These constraints enforce the admissible sharing between non-adjacent mixers.

$$z_{j,i} \implies (\sum_{k=1}^{j-2} z_{j,k} \leq 1) \wedge (\sum_{k=i+1}^{j-1} \sum_{l=1}^{k-2} z_{k,l} = 0) \wedge (W_{j,j-1} < 3) \wedge (W_{i+1,i} < 3)$$

Mixing ratio constraints: The mixing ratio at each mixing

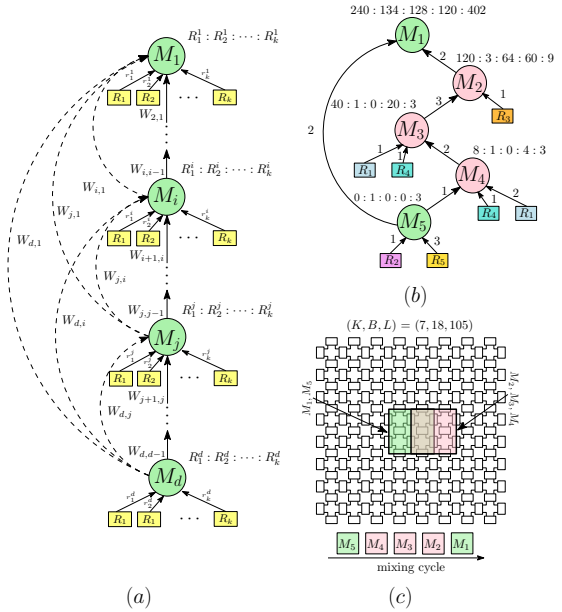


Fig. 8: (a) Augmented graph of depth d with possible sharing of intermediate fluids from a mixer. (b) Skewed mixing graph found at depth 5 for the target ratio $\mathcal{M} = \{0.23 : 0.136 : 0.124 : 0.119 : 0.391\}$ and $\epsilon = 0.0058$. (c) Transport-free placement and mixing sequence for all mixers in the mixing graph.

node is calculated as follows.

$$R_j^i = 4^{d-i} \times r_j^i + \sum_{l=i+1}^{d} W_{l,i} \times R_j^l \times 4^{l-i-1}$$
$$i = 1, 2, \cdots, d; \;\; j = 1, 2, \cdots, k$$

Example 4. *Consider the mixing ratio $\mathcal{M}$ in Example. 2 for which a skewed mixing tree having six mixing operations (i.e., $d = 6$) was found (ref. Example. 3). By enabling fluid sharing between non-adjacent mixing, we obtain a skewed graph having five mixing operations (for $d = 5$). The approximated target ratio is $\mathcal{M}''' = \{240 : 134 : 128 : 120 : 402\}$ which satisfies the ratio-validity condition. The skewed graph is shown in Fig. 8(b), and the corresponding scheduling and placement of the skewed tree is shown in Fig. 8(c). Since the number of mixing nodes is reduced in the skewed mixing graph compared to the skewed mixing tree, sample-preparation time is reduced. We observe that the footprint is increased by two. However, the overhead is reduced as the parameters of the skewed mixing tree (Fig. 5(a)) are reduced to $(K, B, L = 7, 18, 105)$ for the skewed mixing graph (Fig. 8(b)).*

IV. Simulation Results

The proposed method is implemented and compared with existing algorithms [12, 13, 15] for FPVAs. To evaluate the performance of the proposed sample preparation approach with skewed mixing trees/graphs, we performed simulations using a large set of randomly generated target ratios by varying the number of input reagents (k) as 3, 4, and 5. For a fixed number of input reagents, we have considered 1000 different mixing ratios. For each target mixing ratio, we run FloSPA to generate a mixing tree, and the proposed algorithm to generate a skewed mixing tree/graph, for error tolerance (ϵ) 0.01 and 0.001. Note that for $\epsilon = 0.01$, each ratio component of the target and the transformed mixing ratio, is correct up to one decimal point. Similarly, for $\epsilon = 0.001$, the ratio components are correct up to two decimal places. In our experiments, we have considered $\epsilon = 0.01$ and 0.001, which provide sufficient accuracy in target-concentrations for practical cases. Finally, the mixing trees/graphs are mapped onto the target FPVA of size 10×10. We report the number of mixing operations (n_m), total reagent usage (n_r), waste (n_w), and the distribution of tree-heights. Moreover, mixing trees generated with FloSPA and the proposed algorithm are mapped to a 10×10 FPVA to compare overheads which are measured in terms of parameters K, B, L, and the footprint (the total number of FPVA chambers used for mixing). Specifically, K represents the total number of fluid flows, B is the total number of 90° bends in all flow paths, and L length of flow-paths. These parameters are used to optimize the efficiency and reliability of fluid loading on FPVAs as they are indicative of fluid consumption, valve actuations, and the stability of fluid-flow. Lower values of K and L result in reduced fluid usage and fewer valve actuations, respectively, thereby improving efficiency and reliability. Meanwhile, minimizing B helps reduce flow-rate fluctuations and pressure on chamber walls, which maintains operational accuracy and minimizes errors in fluidic operations.

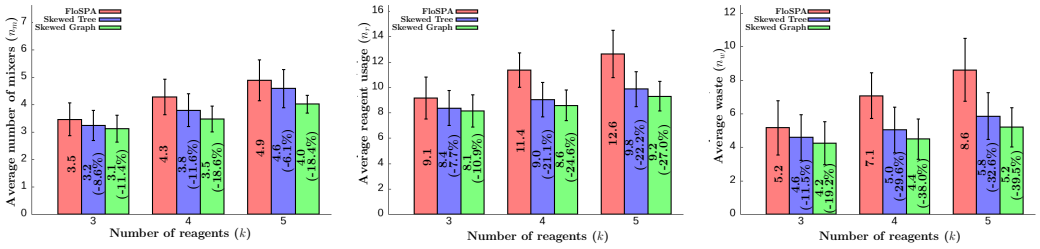
TABLE I: Output tree-height distribution

ϵ	k	Method	Tree-Height (d)								No Tree
			2	3	4	5	6	7	8	9	
0.01	3	FloSPA	56	932	12	0	0	0	0	0	N/A
		Skewed tree	47	658	289	6	0	0	0	0	0
		Skewed graph	50	746	204	0	0	0	0	0	0
	4	FloSPA	9	981	10	0	0	0	0	0	N/A
		Skewed tree	3	205	664	121	7	0	0	0	0
		Skewed graph	3	289	707	1	0	0	0	0	0
	5	FloSPA	3	969	28	0	0	0	0	0	N/A
		Skewed tree	2	29	348	510	97	0	0	0	14
		Skewed graph	2	41	895	62	0	0	0	0	0
0.001	3	FloSPA	2	7	245	746	0	0	0	0	N/A
		Skewed tree	1	4	110	559	324	2	0	0	0
		Skewed graph	1	6	191	788	14	0	0	0	0
	4	FloSPA	0	1	118	881	0	0	0	0	N/A
		Skewed tree	0	1	8	83	521	352	28	4	3
		Skewed graph	0	1	21	421	544	6	5	2	0
	5	FloSPA	0	0	54	946	0	0	0	0	N/A
		Skewed tree	0	0	1	6	102	316	340	152	83
		Skewed graph	0	0	2	50	528	228	131	61	0

ϵ: Error tolerance; k: Number of reagents in the mixing ratio.

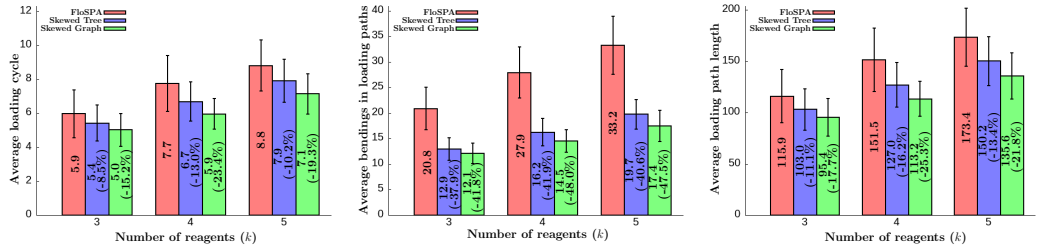
A. Comparing Tree Parameters

Given a mixing ratio and the value of error tolerance, while FloSPA outputs a minimum-depth mixing tree, the proposed method finds a skewed mixing tree/graph by increasing the depth, incrementally, if necessary. Table I shows the distribution of tree-depth for different error-tolerance values (ϵ). Our experiments show that skewed mixing graphs (Table I) are identified within a depth (d) between 2 and 9. Since deeper skewed mixing trees increase sample preparation time,



(a) (b) (c)

Fig. 9: Average value of mixing tree/graph parameters over 1000 mixing ratios for different the number of reagents. Error tolerance (ϵ) value was set to 0.01. (a) Number of mixers (n_m). (b) Total reagent usage (n_r). (c) Total waste (n_w).



(a) (b) (c)

Fig. 10: Overhead is shown in terms of the average number of (a) loading cycle (K), (b) bending (B), and (c) the average length of loading path (L). Average values are computed for 1000 mixing ratios by varying the number of reagents. Error-tolerance (ϵ) value was set to 0.01.

we restrict the search to $d \leq 2k$, where k is the depth of the FloSPA-generated tree. All benchmark mixing ratios yield solutions for $d \leq 9$.

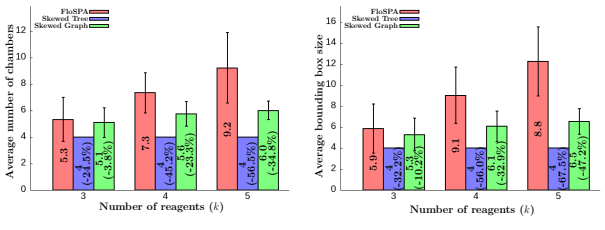
Notice that when $\epsilon = 0.01$, out of 1000 mixing ratios, each having four input reagents, i.e., $k = 4$, FloSPA outputs a mixing tree of depth 2, 3, and 4 for 9, 981, and 10 ratios, respectively. The proposed method also finds a skewed mixing tree for all ratios for which 3, 205, 664, 121, and 7 mixing ratios have their corresponding skewed mixing tree depth 2, 3, 4, 5, and 6, respectively. Furthermore, the proposed method for identifying a skewed mixing graph—by enabling the sharing of intermediate fluids between non-adjacent mixing nodes—can identify 3, 289, 707, and 1 mixing ratio(s) at depths 2, 3, 4, and 5, respectively. This demonstrates that the proposed approach can find a skewed mixing tree/graph with only a nominal increase in depth compared to the minimum-depth mixing tree generated by FloSPA. However, a skewed mixing graph tends to have a lower depth than a skewed mixing tree.

Figs. 9 (a-c) show the mean and standard deviation (error-bar) for the parameters n_m, n_r, and n_w, respectively, for $\epsilon = 0.01$. The bar plots indicate that even though the average values of n_m, n_r, and n_w increase with the number of input reagents, the proposed method achieves significant reductions compared to FloSPA. Moreover, these values are further reduced in the skewed mixing graph. Note that the above parameters are interdependent. The minimum-depth mixing tree generated by FloSPA often has multiple branches, leading to an increase in the number of mixing operations. With more mixings, the number of reagents (n_r) required to realize the mixing ratio on the biochip increases. Consequently, waste (n_w) also increases. In contrast, the skewed mixing tree simplifies the design by minimizing the number of branches, often resulting in fewer mixing operations. This directly reduces the number of input reagents (n_r) required and, subsequently, decreases waste (n_w). The skewed mixing graph improves upon this further by enabling more intermediate-fluid sharing among nodes, which further reduces the number of mix-operations compared to the skewed mixing tree. As a result, both reagent usage (n_r) and waste (n_w) are minimized in the skewed mixing graph. A similar reduction in the average values of parameters can be observed for $\epsilon = 0.001$ in Figs. 6(a-c) in additional document (Appendix III).

B. Comparing Implementation Overheads

For a comparative study, we map both the mixing trees obtained by FloSPA [15] and the proposed method onto a 10×10 FPVA chip to assess the implementation overhead. While placing a mixing tree on the FPVA, we adhere to NTM [12]-constraints for the mixing tree generated by FloSPA, followed by fluid assignment and loading algorithms (LAFCA+DFL [13]). Given a target ratio, we use the combination of FloSPA+NTM+LAFCA+DFL (*a.k.a* baseline) to complete the task of sample preparation. Implementation overheads are quantified using the value of (K, B, L) parameters to realize the target ratio on the FPVA. We have also mapped the skewed mixing tree/graph obtained by the proposed method on the same FPVA, and the values of K, B, and L are noted. Since a skewed mixing trees enable vertical stacking of mixers, with each mixer placed directly atop its predecessor, mapping of the skewed mixing tree/graph does not need any sophisticated procedure. Figs. 10(a-c) show the mean and standard deviation (error-bar) for K, B, and L, respectively, for $\epsilon = 0.01$. From

the bar plots (Figs. 10(a-c)), one can observe considerable improvements in regard to (K, B, L)-parameters when skewed tree/graphs are used for sample preparation on an FPVA. Further improvements can be observed when the number of input reagents increases.



(a) Reaction chambers used (b) Rectangular chip area

Fig. 11: Average value of the (a) footprint (total number of FPVA chambers used for mixer placement) and (b) number of FPVA chambers in the minimum-area isothetic rectangle covering the footprint. Average values are computed for 1000 mixing ratios by varying the number of reagents. Error-tolerance (ϵ) value was set to 0.01.

To evaluate area overhead, we report the footprint (the total number of chambers on the FPVA used for mixer placement) and the bounding box footprint (the minimum-area isothetic rectangle that covers all mixing regions) of the mixers. It is important to note that the bounding box area reflects the irregularity of the mixer footprint. Figs. 11(a-b) show the mean and standard deviation (represented by error bars) for the footprint and bounding box footprint, respectively, for $\epsilon = 0.01$. Since a single mixer is required to implement any skewed mixing tree, the footprint size is four. However, when a skewed mixing graph is used for sample preparation, the footprint may increase. Despite this, both the footprint and bounding-box are are upper-bounded by 7 and 9 units, respectively.

V. Conclusions

In this paper, we have proposed an SMT-based algorithm for producing a skewed mixing tree/graph that can be conveniently mapped to an FPVA satisfying the constraints of No-Transport-Mixing (NTM), during sample preparation. We relax the requirement of having a minimum-depth mixing tree as imposed by existing sample-preparation algorithms, and transform the given mixing ratio to an equivalent one by suitably choosing the depth of mixing tree, if needed, so that a skewed mixing tree/graph can be found. Simulation results show that such a tree/graph can be obtained for most of the mixing ratios by a nominal increase of the depth compared to the minimum-depth mixing tree. Moreover, the usage of a skewed mixing tree/graph reduces reagent volumes, waste production, and fluid-handling load.

References

[1] J. Park, D. H. Han, and J. K. Park, "Towards practical sample preparation in point-of-care testing: user-friendly microfluidic devices," *Lab on a chip*, vol. 20, pp. 1191–1203, 2020.

[2] S.-M. Yang, S. Lv, W. Zhang, and Y. Cui, "Microfluidic point-of-care (POC) devices in early diagnosis: A review of opportunities and challenges," *Sensors*, vol. 22, pp. 1620:1–33, 2022.

[3] W. C. Tai, Y. C. Chang, D. Chou, and L. M. Fu, "Lab-on-paper devices for diagnosis of human diseases using urine samples-a review," *Biosensors*, vol. 11, p. 260, 2021.

[4] S. W. Dutse and N. A. Yusof, "Microfluidics-based lab-on-chip systems in DNA-based biosensing: An overview," *Lab Chip*, vol. 11, pp. 5754–5768, 2011.

[5] S. Park, Y. Zhang, S. Lin, T.-H. Wang, and S. Yang, "Advances in microfluidic pcr for point-of-care infectious disease diagnostics," *Biotechnology Advances*, vol. 29, no. 6, pp. 830–839, 2011.

[6] N. Convery and N. Gadegaard, "30 years of microfluidics," *Micro and Nano Engineering*, vol. 2, pp. 76–91, 2019.

[7] K. Chakrabarty and F. Su, *Digital Microfluidic Biochips: Synthesis, Testing, and Reconfiguration Techniques*. CRC Press, 2018.

[8] T. Thorsen, S. J. Maerkl, and S. R. Quake, "Microfluidic large-scale integration," *Science*, vol. 298, no. 5593, pp. 580–584, 2002.

[9] L. M. Fidalgo and S. J. Maerkl, "A software-programmable microfluidic device for automated biology," *Lab Chip*, vol. 11, pp. 1612–1619, 2011.

[10] X. Huang, H. Cai, W. Guo, G. Liu, T.-Y. Ho, K. Chakrabarty, and U. Schlichtmann, "Control-logic synthesis of fully programmable valve array using reinforcement learning," *IEEE Trans. on CAD*, vol. 43, no. 1, pp. 277–290, 2024.

[11] S. Bhattacharjee, B. B. Bhattacharya, and K. Chakrabarty, *Algorithms for Sample Preparation with Microfluidic Lab-on-Chip*. River Publisher, 2019.

[12] G. Choudhary, S. Pal, D. Kundu, S. Bhattacharjee, S. Yamashita, B. Li, U. Schlichtmann, and S. Roy, "Transport-free module binding for sample preparation using microfluidic fully programmable valve arrays," in *Proc. DATE*, 2020, pp. 1335–1338.

[13] D. Kundu, J. Giri, S. Maruyama, S. Roy, and S. Yamashita, "Fluid-to-cell assignment and fluid loading on programmable microfluidic devices for bioprotocol execution," *Integration*, vol. 78, pp. 95–109, 2021.

[14] P. Spedding, E. Benard, and G. Mcnally, "Fluid flow through 90 degree bends," *Asia-Pacific Journal of Chemical Engineering*, 2008.

[15] S. Bhattacharjee, S. Poddar, S. Roy, J.-D. Huang, and B. B. Bhattacharya, "Dilution and mixing algorithms for flow-based microfluidic biochips," *IEEE Trans. on CAD*, vol. 36, no. 4, pp. 614–627, 2017.

[16] F. Harary and A. J. Schwenk, "The number of caterpillars," *Discrete Mathematics*, vol. 6, no. 4, pp. 359–365, 1973.

[17] W. Thies, J. P. Urbanski, T. Thorsen, and S. P. Amarasinghe, "Abstraction layers for scalable microfluidic biocomputing," *Natural Computing*, vol. 7, no. 2, pp. 255–275, 2008.

[18] S. Bhattacharjee, Y.-L. Chen, J.-D. Huang, and B. B. Bhattacharya, "Concentration-resilient mixture preparation with digital microfluidic lab-on-chip," *ACM Trans. Embed. Comput. Syst.*, vol. 17, no. 2, pp. 49:1–49:12, 2018.

[19] S. Bhattacharjee, R. Wille, J. Huang, and B. B. Bhattacharya, "Storage-aware algorithms for dilution and mixture preparation with flow-based lab-on-chip," *IEEE Trans. on CAD*, vol. 39, no. 4, pp. 816–829, 2020.

[20] L. M. de Moura and N. Bjørner, "Z3: An efficient SMT solver," in *Proc. of TACAS*, 2008, pp. 337–340, [Z3 is available at https://github.com/Z3Prover/z3].

Abhik Kumar Khan is currently working as a Software Engineer at Bank of America, Hyderabad. He earned his M.Tech degree in Computer Science and Engineering from Indian Institute of Technology, Guwahati, in 2024. He received his B.Tech degree in Computer Science and Engineering from the Institute of Engineering and Management, Kolkata, in 2020. His research interests include microfluidics, automation, and machine learning.

Sudip Roy (Member, IEEE) received the B.Sc. degree (Hons.) in physics and the B.Tech. degree in computer science and engineering from the University of Calcutta, Kolkata, India, in 2001 and 2004, respectively, and the M.S. (by research) and Ph.D. degrees in computer science and engineering from the Indian Institute of Technology (IIT) Kharagpur, Kharagpur, India, in 2009 and 2014, respectively.

He is currently an Associate Professor with the Department of Computer Science and Engineering, Indian Institute of Technology (IIT) Roorkee, Roorkee, India. His research areas include electronic design automation for digital systems and microfluidic biochips, optimization techniques, and information and communication technologies for disaster risk reduction.

Bhargab B. Bhattacharya (F' 07) is currently on the faculty of Ashoka University, Sonipat, Haryana, India. Prior to that he worked at the Indian Statistical Institute, Kolkata, at Indian Institute of Technology Kharagpur, and at National Institute of Technology Rourkela. He also served as Visiting Professor at the University of Nebraska-Lincoln, and Duke University, USA, at the University of Potsdam, Germany, and at Tsinghua University, Beijing, China. He received the B.Sc. degree in physics from the Presidency College, Kolkata, B.Tech. and M.Tech. degrees in Radiophysics and Electronics, and the PhD degree in Computer Science, all from the University of Calcutta, India, in 1971, 1974, 1976, and in 1986 respectively.

Dr. Bhattacharya's research interests include VLSI design and test, microfluidic lab-on-chips, image analysis, and geometric algorithms. He has published more than 400 technical articles, authored/edited five books, and he holds one Indian and ten US Patents.

Sukanta Bhattacharjee received the B.Sc. degree (Hons.) in computer science and the B.Tech. degree in computer science and engineering from the University of Calcutta, Kolkata, India, in 2006 and 2009, respectively, and the M.Tech. and Ph.D. degrees in computer science from the Indian Statistical Institute, Kolkata, in 2012 and 2017, respectively.

He is working as an Assistant Professor with the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati, India. His research interests include design automation algorithms, microfluidics, security, and machine learning.