Routing-Aware Resource Allocation for Mixture Preparation in Digital Microfluidic Biochips*

Sudip Roy[†], Partha P. Chakrabarti[†], Srijan Kumar[†], Bhargab B. Bhattacharya[‡], and Krishnendu Chakrabarty[§]
[†]*Indian Institute of Technology Kharagpur, India. E-mail: {sudipr, ppchak, srijankumar}@cse.iitkgp.ernet.in*
[‡]*Indian Statistical Institute Kolkata, India. E-mail: bhargab@isical.ac.in*
[§]*Duke University, Durham, USA. E-mail: krish@ee.duke.edu*

***Abstract*—On-chip mixing of several biochemical fluids with a specified ratio of concentration factors is a challenging problem in automating biochemical laboratory protocols on a digital microfluidic biochip. The performance of a mixing algorithm depends on resource allocation, e.g., the placement of mixer modules, storage units, boundary reservoirs or dispensers on the chip floor. With a limited number of resources, mixing of a large number of fluids may be slowed down because of the stalls arising out of fluidic constraints during droplet transportation. In this paper, we propose a routing-aware resource allocation technique which can be adopted with two basic mixing algorithms. Simulation results show that on an average, the proposed scheme can reduce the number of droplet crossing paths by 75.4% or by 89.7%, depending on the underlying basic algorithm used for mixture preparation.**

I. Introduction

Digital microfluidic (DMF) biochips offer a potential platform for implementing a wide range of biochemical laboratory protocols [1]. Recently, several design automation methods have been proposed for architectural synthesis (i.e., operation scheduling, resource selection and binding) and physical design (i.e., module placement and droplet routing) of DMF biochips [1–3]. For a real-life bioprotocol, sample and mixture preparation steps play an important role in high-throughput applications. Many CAD algorithms and schemes for automatic mixture preparation have been reported [4–10]. Hsieh et al. [11] presented a design methodology with dynamic error recovery for architectural and layout synthesis of a sample preparation biochip.

In a mixture preparation biochip, the on-chip mixing modules, storage units (some additional electrodes used to store intermediate droplets), and the fluid reservoirs at the chip boundary are regarded as *resources*, which are non-sharable and indivisible. In this paper, we demonstrate that reagent allocation to the boundary reservoirs plays an important role in determining the time to route droplets to the mixers; this in turn, strongly affects the overall time for mixture preparation. Given a desired (or target) ratio, a mixing tree can be constructed by running an existing algorithm [4, 6, 9, 10].

*This work of S. Roy was supported by Microsoft Corporation and Microsoft Research India under the Microsoft Research India PhD Fellowship Award. The work of B. B. Bhattacharya was supported by a special grant to Nanotechnology Research Triangle provided by Indian Statistical Institute. The work of K. Chakrabarty was supported in part by the US National Science Foundation under grant no. CCF-0914895.

We model the various droplet transportation routes from fluid reservoirs to mixers with a bipartite graph where the nodes are drawn as two disjoint linearly ordered sets, and the edges are drawn as straight-lines; next, the total number of edge intersections (i.e., crossing droplet pathways) is minimized by reallocating the fluid reservoirs. This leads to a suitable placement of reservoirs and mixers for an application-specific biochip, which reduces the path crossovers as well as the total length of droplet transportation routes. This, in turn, reduces the total number of stalls (in terms of clock cycles) required by the droplets to avoid unintended mixing. Simulation results show that on an average (for varying number of available on-chip mixers), our scheme can reduce the number of crossovers in the droplet routing paths by 75.4% and 89.7%, respectively for the mixing trees obtained by two existing mixing algorithms, namely *MM* [4] and *RMA* [6].

The remainder of the paper is organized as follows. Section II describes the related prior work and the problem formulation. The proposed method is presented in Section III and the simulation results are shown in Section IV. Finally, we conclude the paper in Section V.

II. Automatic Mixture Preparation

The *mixture preparation* is a preprocessing step for mixing two or more fluids in a given ratio. In the (1:1)-mixing model, two equal-volume droplets of biochemical fluids with concentration factors (CFs) C_1 and C_2 are mixed and a subsequent balanced splitting is performed to obtain two unit-volume resultant droplets of $CF = \frac{C_1+C_2}{2}$. A *mixing tree* is a binary tree whose leaf nodes indicate fluid inputs and the non-leaf nodes denote (1:1) mix-split steps between two input (or intermediate) fluid droplets. One (1:1) mixing and a consecutive balanced splitting are together referred to as a *mix-split cycle*. The root of a mixing tree represents the target mixture and the level-order traversal of a mixing tree provides the sequence of such mix-split cycles. After each mix-split cycle, two unit-volume intermediate droplets are produced, one of which is discarded as a *waste droplet* and the other one is used in the next cycle as an intermediate droplet. If the depth of the mixing tree is d, the maximum error in CF of any constituent fluid x_i in the target mixture can be expressed as $\frac{1}{2^{d+1}}$. In general, the target mixture with a desired ratio a_1:a_2:...:a_N of N different fluids $x_1, x_2, \ldots, x_N$ is represented by an algebraic expression $\frac{a_1x_1+a_2x_2+\ldots+a_Nx_N}{L}$, where $a_1, a_2, \ldots, a_N$ are non-zero positive integers and L $(= \sum_{i=1}^{N} a_i = 2^d)$ is the sum of ratio components. Hence, the CF of fluid x_i in the target mixture is $\frac{a_i}{L}$, such that $\sum_{i=1}^{N} \frac{a_i}{L} = 1$.

Let T_{ms} be the total number of mix-split steps (i.e., non-leaf nodes in the mixing tree) and W be the total number of waste droplets (discarded intermediate droplets other than two target droplets) generated during the execution of a mixing tree. A scheduled mixing tree provides the sequence of mix-split cycles with timing assignment and mixer allocation to the non-leaf nodes. Let M_{lb} be the minimum number of mixers required for minimum-time completion of mixing tree execution (i.e., d time cycles) as computed in [12]. The parameters T_{ms}, W and M_{lb} are used to characterize the mixing tree obtained by a mixing algorithm for a given target ratio.

A. Prior Work and Motivation

In the literature, there exist three mixing algorithms to determine different mixing trees from a target ratio such as, ***MM*** [4], ***RMA*** [6] and ***RSM*** [9]. Among various ways to schedule a mixing tree, we adopt a scheduling scheme for optimal-time completion of mixture preparation, referred as *OSM* (Optimal-Scheduling-With-M-Mixers) [12]. For a target ratio 7:14:11 of three biofluids x_1, x_2 and x_3, Fig. 1(a), (b) and (c) present the *MM*-tree, *RMA*-tree and *RSM*-tree, respectively, all scheduled with two mixers M_1 and M_2 (since $M_{lb} = 2$). The timing assignments (t) to non-leaf nodes in a mixing tree are marked by 1 to 5 (in 'red'). In a scheduled mixing tree, if the timing difference between two non-leaf nodes of a directed edge is more than one, then a storage unit is required during that running time cycle. We can estimate an overhead (U) of storage unit requirements as the sum of the storage units required during all the time cycles. For the scheduled *MM*-tree of Fig. 1(a), $U = \{0 + 1 + (1 + 1) + 1\} = 4$.

Without loss of generality, we assume a layout with two on-chip mixers and three boundary reservoirs for a mixture preparation biochip as shown in Fig. 1. The reservoirs are located from left-to-right for loading fluids x_1, x_2 and x_3, respectively, and two mixers (M_1 and M_2) are placed on-chip ordered from left-to-right. Here, the loading of a reservoir R_i with an input fluid x_i is denoted by $x_i \rightarrow R_i$. If storage units are required for a scheduled mixing tree, they may also be included in the layout. By *resource allocation* we mean loading of input fluids to boundary reservoirs and assignment of the non-leaf nodes of a mixing tree to the on-chip mixers. As shown in Fig. 1, the droplet transportation routes can be listed from the scheduled mixing tree for resource allocation in the layout. For $t = 1$, the droplet

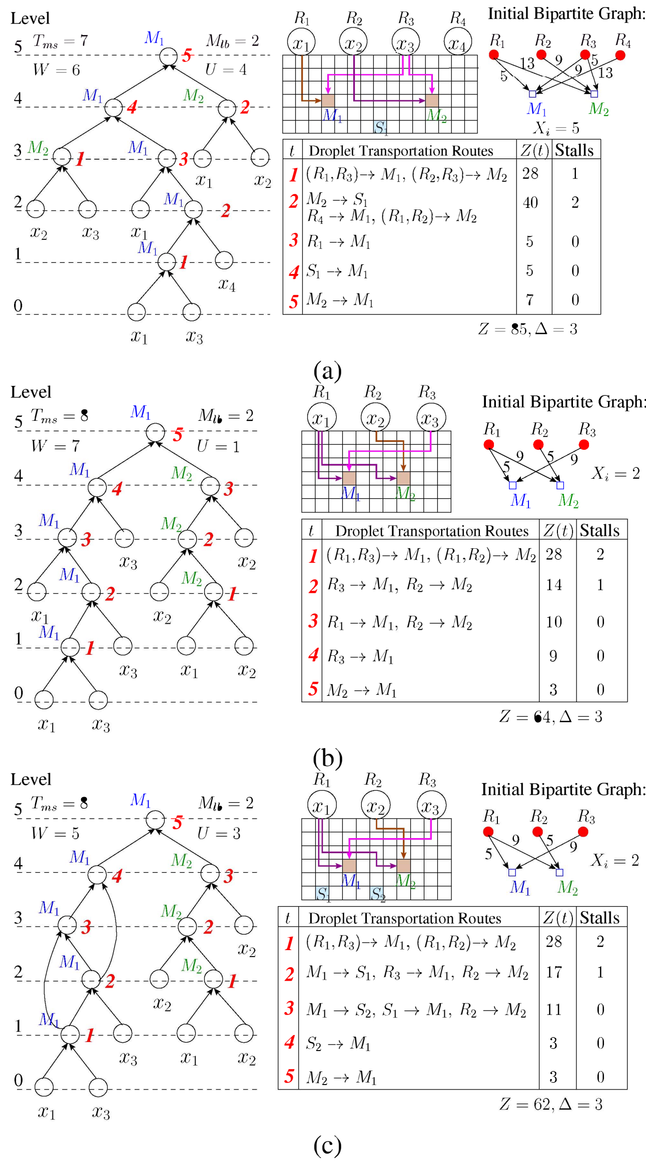

Figure 1. Scheduled mixing trees obtained by (a) *MM* [4] for a ratio 13:12:5:2, (b) by *RMA* [6] and (c) *RSM* [9], for a ratio 7:14:11. (Respective layouts, bipartite graphs and droplet transportation routes are also shown).

routing paths are shown in the figure of layout and the number of stalls required during this transportation can be justified by the number of crossing between two such paths. For any t, the number of used cells (electrodes) during the droplet transportations are denoted by $Z(t)$. Hence, the total length of droplet transportation routes can be computed as $Z = \sum_t Z(t)$ and the total number of stalls required (i.e., Δ) can be estimated as the number of crossovers required to pass at the same time by two different input droplets.

For a scheduled mixing tree, the initial resource allocation is modeled as an initial directed bipartite graph (edge weight indicates routing length) as depicted in Fig. 1. Let X_i be the number of edge-crossings in the initial bipartite graph (for the example mixing tree shown in Fig. 1, $X_i = 2$). We assume that all the mix-split steps of a mixing tree require

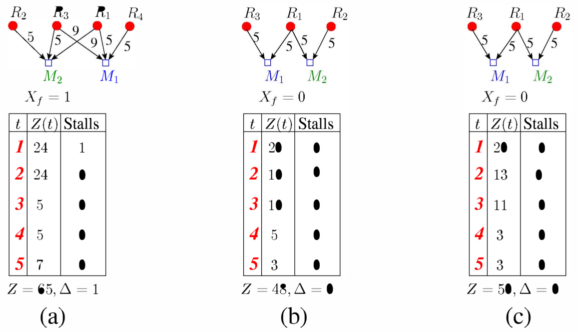

Figure 2. Modified bipartite graphs of different resource allocation, Z and Δ for scheduled (a) *MM*-tree, (b) *RMA*-tree and (c) *RSM*-tree.

an identical time duration, which is much less than the time to transport input droplets of a mixer. However, if the total routing length for a mixture preparation is reduced, then the routing time, the washing load of the used cells, and the chip size will also be reduced.

We observe that if we change the order of reservoirs and mixers (i.e., their relative positions) for the initial bipartite graphs of Fig. 1, the droplet transportation overheads (Z and Δ) are reduced significantly. The modified bipartite graphs for different resource allocation are shown in Fig. 2. It is evident that if we can reduce the number of crossings in the final bipartite graph (i.e., X_f), then Z and Δ can be reduced automatically. This fact motivates us to formulate a problem of routing-aware resource allocation (with reduced X_f in an embedded bipartite graph) for on-chip mixture preparation in an application-specific biochip.

B. Problem Formulation

We envisage *resource allocation* as a mapping function that provides a linear ordering of the reservoirs and that among the mixers, i.e., their relative positions in the chip layout from left-to-right. Consider that the initial and final (determined by the resource allocation ϕ) orders of reservoirs (mixers) are denoted by Π_r^i (Π_m^i) and Π_r^f (Π_m^f), respectively. For a scheduled mixing tree, if the mixers (M_i's) and reservoirs (R_i's) are ordered and relatively placed in such a way that the total number of edge-crossing in the corresponding bipartite graph (X_f) is minimum, then the resource allocation is called an *optimal* (ϕ_{opt}). Our objective is to find an optimal resource allocation (ϕ_{opt}) for an application-specific mixture preparation biochip. We formulate the problem as follows: **Inputs**: (a) N fluids x_1,x_2,...,x_N each with $CF = 100\%$, (b) a target ratio a_1:a_2:...:a_N of N fluids such that $\sum_i^N a_i = 2^d$, where (c) d is the depth of the mixing tree $\mathcal{T}$, and (d) M the number of on-chip mixers. **Outputs**: (a) A scheduled mixing tree $\mathcal{T}_{sch}$, and (b) the optimal resource allocation ϕ_{opt} for the mixture preparation biochip to execute $\mathcal{T}_{sch}$. A scheduled mixing tree $\mathcal{T}_{sch}(P \cup Q, E)$ has a set of non-leaf nodes (P), set of leaf nodes (Q) and a set of directed edges (E) between two nodes. Each node q, $q \in Q$ indicates a mapping of fluid x_i loaded in the reservoir R_i (i.e., the mapping $x_i \rightarrow R_i$) denoted by the function $x_i = fluid(q)$. Whereas, each node p, $p \in P$ has an assigned time cycle t denoted by the function $t = time(p)$ along with an allocated mixer M_j denoted by the function $M_j = mixer(p)$. The time of completion (T_c) for a scheduled mixing tree is computed as $T_c = \max(time(p)), \forall p \in P$. However, as the problem of minimizing the number of edge-crossings in a bipartite graph is NP-hard [13], finding an optimal resource allocation (ϕ_{opt}) is also NP-hard.

III. Resource Allocation for Mixture Preparation

To obtain a suitable (which is near-optimal) resource allocation (ϕ_{opt}) for an application-specific mixture preparation biochip, we propose an algorithm, 'routing-aware resource allocation for mixture preparation' (*RAMP*). The number of on-chip mixers (M) may be either pre-determined or restricted to the minimum number of mixers for minimum-time completion of mixing tree (M_{lb}). The pseudo-code for our proposed algorithm is written in **Algorithm 1**. It uses the barycenter heuristic [14] to reduce the number of edge-crossings in the bipartite graph. The steps of *RAMP* are discussed in the following subsections.

A. Determining Mixing Tree for a Target Ratio

For a target ratio, any existing mixing algorithm can be used to determine the mixing tree. In a mixing tree $\mathcal{T}$, the (leaf and non-leaf) nodes are at different levels 0 to d. The root of $\mathcal{T}$ is at level d and the level of any other node is one less than the level of its parent node. Let T_{ms} be the total number of non-leaf nodes (mix-split cycles) in $\mathcal{T}$.

B. Scheduling a Mixing Tree

Luo and Akella [12] presented a scheme *OSM* for optimal scheduling of a mixing tree with a given number (M) of mixers. A mixing tree $\mathcal{T}$ is scheduled by *OSM* to get the scheduled mixing tree $\mathcal{T}_{sch}$ that provides a sequence of mix-split cycles with timing assignment and mixer allocation to each non-leaf node of the $\mathcal{T}_{sch}$. The time of completion T_c is

Algorithm 1 *RAMP* (a_1:a_2:...:a_N, d, M)

Compute $L = \sum_i^N a_i$.
Determine mixing tree $\mathcal{T}(P \cup Q, E)$ by a mixing algorithm, where $\forall q \in Q$, $f(q) : x_i \rightarrow R_i$. Compute $T_{ms} = |P|$.
Set $T_{lb} = d$ and compute M_{lb} as provided in [12].
If M is not known, set $M = M_{lb}$.
Schedule $\mathcal{T}$ by *OSM* [12] with M mixers to obtain $\mathcal{T}_{sch}(P \cup Q, E)$, so that each node p, $p \in P$, has a pair $\{time(p), mixer(p)\}$ for time cycle and mixer assignments. Compute $T_c = \max(time(p)), \forall p \in P$.
Set initial orders as Π_r^i : $R_1, R_2, \ldots, R_N$ and Π_m^i : $M_1, M_2, \ldots, M_M$.
G = *GetBipartiteGraph*($\mathcal{T}_{sch}$). Compute X_i.
Apply barycenter heuristic on G to get the final orders Π_r^f and Π_m^f, such that $X_f \leq X_i$.
Return final resource allocation $\phi : (\Pi_r^f, \Pi_m^f)$.

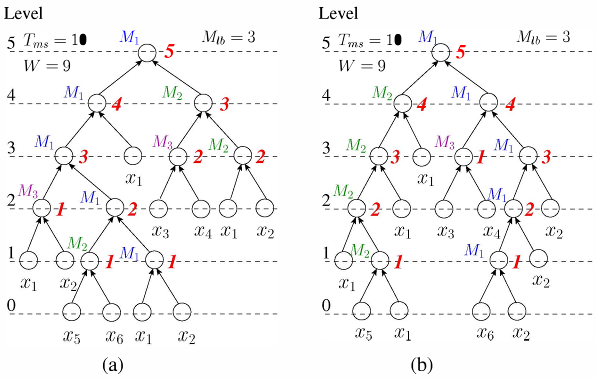

Figure 3. For target ratio 15:7:4:4:1:1 (a) *MM*-tree and (b) *RMA*-tree, both optimally scheduled by *OSM* with $M = M_{lb}$.

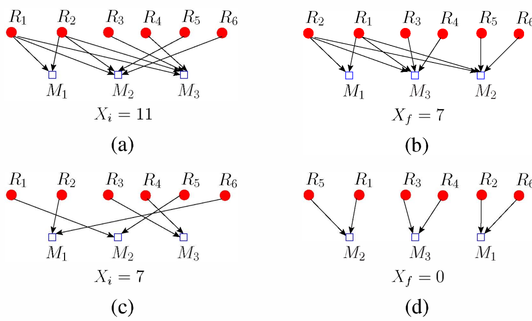

Figure 4. Target ratio 15:7:4:4:1:1. (a) Initial and (b) final bipartite graphs for the scheduled *MM*-tree. (c) Initial and (d) final bipartite graphs for the scheduled *RMA*-tree.

the total number of time cycles required to schedule $\mathcal{T}$ with M mixers for mixture preparation. In order to complete the mixture preparation by only one mixer (i.e., for $M = 1$), one needs $T_c = T_{ms}$, whereas if $M > 1$, T_c will be equal to the maximum time cycle assigned to a non-leaf node in $\mathcal{T}_{sch}$. As aforementioned, M_{lb} is the minimum number of mixers required to complete mixing in minimum time (T_{lb}) and T_{lb} is the depth of the mixing tree, i.e., $T_c = T_{lb} = d$.

Let M^* and M' be the values of M_{lb} for *MM*-tree [4] and *RMA*-tree [6] of a target ratio, respectively, where M' may be greater than M^*. For an example target ratio 15:7:4:4:1:1, we found $M_{lb} = 3$ for both *MM*-tree and *RMA*-tree with $T_{lb} = 5$ as shown in Fig. 3. In the scheduled *MM*-tree and *RMA*-tree with three mixers shown in Fig. 3, each non-leaf node is assigned with a mixer M_i ($i = 1, 2$ *or* 3) and a time cycle t ($t = 1, 2, 3, 4$ *or* 5).

C. Initial Resource Allocation

An application-specific mixture preparation biochip consists of on-chip mixers and the boundary fluid reservoirs placed at fixed locations corresponding to the non-leaf nodes and input fluids, respectively. The initial resource allocation is modeled and determined by **Algorithm 2** in accordance to a bipartite graph $G(X, Y)$, where the set of vertices, X can be divided into two disjoint sets of vertices mapping the set of mixers (X_m), and set of reservoirs (X_r). The set of directed edges Y denotes the droplet transportation

Algorithm 2 *GetBipartiteGraph*$(\mathcal{T}_{sch}(P \cup Q, E))$

Initialize $G = (X, Y)$ as $X = X_r \cup X_m$ and $Y = \phi$, where $X_r = \{R_1, R_2, \ldots, R_N\}$ and $X_m = \{M_1, M_2, \ldots, M_M\}$.
for each $p \in P$ **do**
  if left-child$(p) = q$, $q \in Q$ **then**
    $x_i = fluid(q)$, $Y = Y \cup \overrightarrow{(R_i, mixer(p))}$. /* add a directed edge */
  if right-child$(p) = q$, $q \in Q$ **then**
    $x_i = fluid(q)$, $Y = Y \cup \overrightarrow{(R_i, mixer(p))}$. /* add a directed edge */
Return G.

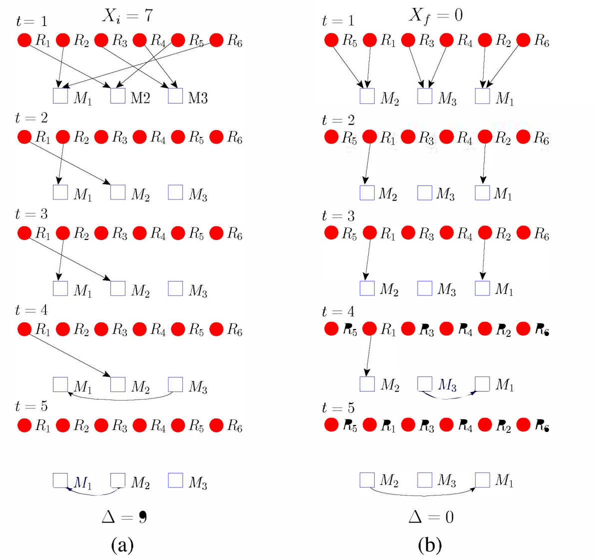

Figure 5. Cycle-wise droplet transportation routes for the scheduled *RMA*-tree (shown in Fig. 3b) in case of (a) initial and (b) final resource allocation.

routes from the fluid reservoirs to mixers, and for each directed edge $(v_1, v_2) \in Y$, $v_1 \in X_r$ and $v_2 \in X_m$. The set of vertices X_r (X_m) of the bipartite graph $G(X, Y)$ has an associated order Π_r^i (Π_m^i) of reservoirs (mixers). Here, an order is the permutation with increasing indices from left-to-right. Let X_i be the total number of edge-crossings among the edges (i.e., droplet transportation routes) in the initial bipartite graph G. As an example, the initial resource allocations for the scheduled *MM*-tree and *RMA*-tree are depicted in Figs. 4(a) and (c), respectively.

D. Routing-Aware Resource Allocation

Note that the initial number of edge-crossings (X_i) can be reduced (to X_f of final bipartite graph), if we can reorder the reservoirs and mixers by changing the relative locations of R_is (i.e., reservoir assignment for the input fluids) and M_js (i.e., mixer allocation to the non-leaf nodes). An optimal resource allocation (ϕ_{opt}) has the minimum number of final edge-crossings (X_f) in the bipartite graph G. We adopt the barycenter heuristic [14] to minimize X_f in the final bipartite graph. For the scheduled *MM*-tree and *RMA*-tree of Fig. 3, the final bipartite graphs for resource allocations are shown in Fig. 4(b) and (d), respectively. We observed that the number of edge-crossings in the corresponding bipartite graph is reduced from 11 to 7 for the *MM*-tree and from 7 to 0 for the *RMA*-tree.

E. Discussions on RAMP

There are several advantages in applying the proposed scheme while designing a biochip for automatic mixture preparation. These are as follows. (a) It provides a suitable resource allocation for a scheduled mixing tree by reducing the total length of droplet transportation routes (Z), i.e., the total number of used cells of the chip, which, in turn, reduces electrode actuations and the area of the layout. (b) It reduces the total stalls required during transportation of input fluid droplets from the reservoirs to the mixers, which, in turn, reduces the time to complete mixture preparation. (c) As it reduces Z, the washing load [15] is also reduced.

IV. Simulation Results

We have carried out simulation experiments with a large number of target ratios to evaluate *RAMP*. In number theory and combinatorics, integer partitioning is a way of writing an integer as a sum of positive integers, regardless of their order [16]. By convention, the partitions are usually ordered from the largest to the smallest, e.g., 4 can be partitioned in five distinct ways: 4, 3+1, 2+2, 2+1+1, 1+1+1+1. We use different distinct partitions (with only N components) of L as the target ratios of N fluids, where $N > 2$. In the case of dilution problem (i.e., $N = 2$), a dilution tree has only one branch and only one mixer can be assigned to execute mixing with no crossovers among the droplet routing paths. Thus, while considering the target ratios of more than two fluids, we keep those partitions in which the integers are set-wise co-prime, so that two different ratios do not eventually turn into the same ratio. Hence, the target ratios 2:1:1 and 1:1:1:1 may be considered for $L = 4$. In real-life bioprotocols, as many as 12 different fluids may need to be mixed to prepare a target mixture that may be used in a biochip [17]. A portion of the data set for a large number of synthetic ratios is depicted in Fig. 6, which shows how some ratios are excluded because of the equivalence of two ratios and the reducibility of one ratio to another.

For some example ratios, comparative results on *MM* [4] and *RMA* [6] for improving X_f are presented in Table I.

We simulate *RAMP* for 6058 synthetic target ratios with $L = 32$ (i.e., $d = 5$) of N different fluids, where $3 \leq N \leq 12$. For all these 6058 target ratios, *RAMP* is applied to both *MM*-trees and *RMA*-trees with $M = M^*$. Figure 7(a) and 7(b) show that for both the mixing algorithms the distributions of X_f move towards the origin from that of X_i. Such shifts indicate that the total number of edge-crossings in the resource allocation can be reduced by *RAMP*, which in turn reflects our claim that *RAMP* can reduce Z and Δ. Hence, for a comparatively large number of target ratios, we observe X_f values reduced from X_i values. Similarly, for 533366 synthetic target ratios with $L = 64$ (i.e., $d = 6$) of N different fluids, where $3 \leq N \leq 12$, *RAMP* can reduce the total number of edge-crossings in the final resource allocation (see Figs. 7c and 7d).

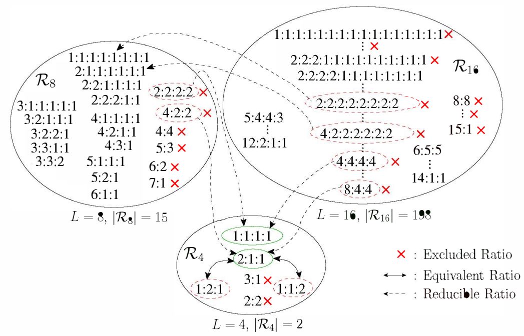

Figure 6. A snapshot of the ratio data set (two ratios are said to be *equivalent* if they represent two different permutations of the same integers; one ratio is *reducible* to another if one can be obtained from the other by factoring out an integer; a ratio is *excluded* if it is reducible to another, or not within the domain of our assumption $3 \leq N \leq 12$).

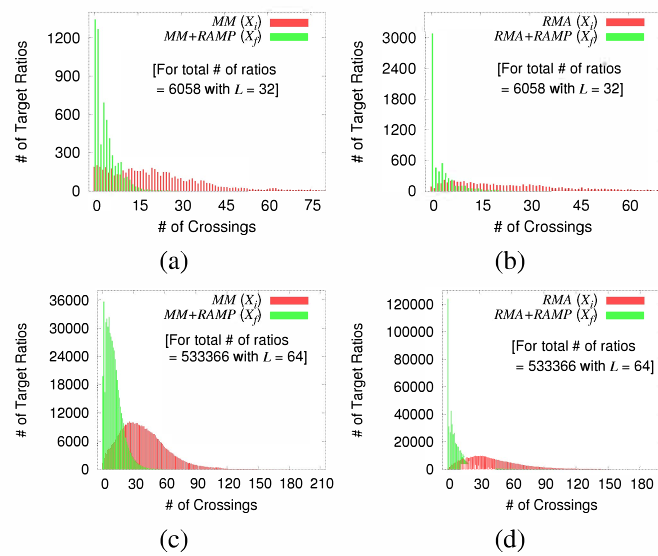

Figure 7. Histograms for improvements in the number of crossings by *RAMP* for $L = 32$ over (a) *MM*, (b) *RMA*, and for $L = 64$ over (c) *MM* and (d) *RMA*, when the *RMA*-trees are scheduled with $M = M^*$ mixers.

We simulate *MM* and *RMA* over 6058 target ratios with $L = 32$ and after scheduling the mixing trees with varying number of mixers we estimate the (μ, σ)-pair for the distributions of the number of initial edge-crossings (X_i). Next, we apply *RAMP* with *MM* and *RMA* for the same target set and estimate the (μ, σ)-pair for the distributions of the number of final edge-crossings (X_f). Moreover, we compute the average percentage improvements in the number of final edge-crossings by *RAMP* over *MM* and *RMA* for varying number of mixers (M). The number of on-chip mixers (M) is varied from 2 to 7 and also equals to the M_{lb} of the mixing tree obtained by *MM* or *RMA*. However, we consider a case of using $M = M^*$ (M^* is M_{lb} of the corresponding *MM*-tree) for both *MM*-trees and *RMA*-trees. The results of (μ, σ)-pair for the distributions of the number of crossings and the average percentage improvements in the number of edge-crossings in the final resource allocation by *RAMP* are shown in Table II. On an average, *RAMP* can provide 75.4%

Table I
RESULTS OF *RAMP* WITH *MM* AND *RMA* FOR SOME (RANDOMLY TAKEN) TARGET RATIOS.

Example Target Ratio	*MM* [4] $< T_{ms}, T_{lb}, M_{lb}, T_c, M >$	*MM* + *RAMP* X_i	*MM* + *RAMP* X_f	*RMA* [6] $< T_{ms}, T_{lb}, M_{lb}, T_c, M >$	*RMA* + *RAMP* X_i	*RMA* + *RAMP* X_f	Best Solution $(\mathcal{T}_{sch}, \Pi_r^f, \Pi_m^f)$
15:7:4:4:1:1	10 and <10, 5, 3, 5, 3>	11	7	10 and <5, 3, 5, 3>	7	0	$\mathcal{T}_{sch}(RMA)$ $\Pi_r^f = (R_5, R_1, R_3, R_4, R_2, R_6)$ $\Pi_m^f = (M_2, M_3, M_1)$
2:3:5:7:11:13:87	18 and <18, 7, 4, 7, 4>	47	14	19 and <7, 4, 7, 4>	21	2	$\mathcal{T}_{sch}(RMA)$ $\Pi_r^f = (R_4, R_1, R_5, R_3, R_7, R_6, R_2)$ $\Pi_m^f = (M_3, M_4, M_1, M_2)$
341:341:342	14 and <10, 2, 10, 2>	1	1	18 and <10, 2, 10, 2>	1	0	$\mathcal{T}_{sch}(RMA)$ $\Pi_r^f = (R_2, R_1, R_3)$ $\Pi_m^f = (M_1, M_2)$
12:7:7:3:3	11 and <5, 3, 5, 3>	7	1	12 and <5, 3, 5, 3>	7	4	$\mathcal{T}_{sch}(MM)$ $\Pi_r^f = (R_3, R_2, R_1, R_4, R_5)$ $\Pi_m^f = (M_3, M_1, M_2)$
18:5:3:3:3	9 and <5, 3, 5, 3>	15	4	10 and <5, 3, 5, 3>	6	2	$\mathcal{T}_{sch}(RMA)$ $\Pi_r^f = (R_3, R_1, R_2, R_4, R_5)$ $\Pi_m^f = (M_3, M_1, M_2)$

Table II
(μ, σ)-PAIR OF DISTRIBUTIONS AND AVERAGE % IMPROVEMENTS BY *RAMP* IN # CROSSINGS OVER 6058 TARGET RATIOS FOR $L = 32$.

# Mixers M	*MM* [4] (μ, σ)	*MM+RAMP* (μ, σ)	*MM+RAMP* $\% \, impr.$	*RMA* [6] (μ, σ)	*RMA+RAMP* (μ, σ)	*RMA+RAMP* $\% \, impr.$
2	(10.4,6.5)	(1.5,2.4)	77.3%	(14.2,8.5)	(2.2,4.2)	85.4%
3	(18.6,12.7)	(3.4,4.1)	75.1%	(20.6,14.1)	(3.6,5.7)	86.3%
4	(20.0,13.9)	(3.7,4.2)	74.5%	(22.0,16.6)	(2.5,4.7)	90.6%
5	(20.7,15.6)	(3.6,3.9)	75.2%	(22.9,17.7)	(2.2,4.1)	91.2%
6	(20.8,16.1)	(3.6,4.0)	75.2%	(23.0,17.9)	(2.0,3.3)	91.7%
7	(20.9,16.2)	(3.6,4.0)	75.2%	(22.9,17.8)	(1.9,3.1)	91.8%
M_{lb}	(20.9,16.2)	(3.6,4.0)	75.2%	(22.9,17.7)	(1.9,3.0)	91.8%
M^*	(20.9,16.2)	(3.6,4.0)	75.2%	(23.2,17.6)	(2.7,4.4)	88.9%
Average % improvement			**75.4%**			**89.7%**

and 89.7% improvements of the number of edge-crossings in resource allocation over that by *MM* and *RMA*, respectively. We found that with a given number of mixers, *RAMP* can reduce the overhead of droplet routing and washing compared to two state-of-the-art mixing algorithms.

V. CONCLUSIONS

In this paper, we have proposed a routing-aware resource allocation scheme for automatic mixture preparation on a DMF biochip. For a given target ratio, we have modeled the fluidic transportation routes from fluid reservoirs to mixers as a bipartite graph, and then used a heuristic procedure to reduce the total number of edge-crossings. The suitability of two basic mixture preparation algorithms for droplet routability is investigated. Simulation results show that in both the cases, an efficient resource allocation can be made which lessens the workload of droplet transportation and washing significantly. It has been observed that *RAMP* when applied on *MM*-trees [4] can provide 75.4% improvements in the number of droplet-path crossovers, whereas that on *RMA*-trees [6] can produce 89.7% improved results for single target generation. We believe that another existing mixing algorithm *RSM* [9], which was primarily designed for producing multiple target ratios, will perform even better, if the proposed scheme *RAMP* is augmented with it. These results will be reported in a future work.

REFERENCES

[1] K. Chakrabarty and T. Xu, *Digital Microfluidic Biochips: Design and Optimization.* CRC Press, 2010.
[2] F. Su and K. Chakrabarty, "High-level Synthesis of Digital Microfluidic Biochips," vol. 3, no. 4, pp. 1–32, 2008.
[3] Y. Zhao and K. Chakrabarty, "Cross-Contamination Avoidance for Droplet Routing in Digital Microfluidic Biochips," *IEEE TCAD*, vol. 31, no. 6, pp. 817–830, 2012.
[4] W. Thies *et al.*, "Abstraction Layers for Scalable Microfluidic Biocomputing," *Natural Computing*, vol. 7, no. 2, pp. 255–275, 2008.
[5] S. Roy *et al.*, "Optimization of Dilution and Mixing of Biochemical Samples using Digital Microfluidic Biochips," *IEEE TCAD*, vol. 29, no. 11, pp. 1696–1708, 2010.
[6] ——, "Layout-Aware Solution Preparation for Biochemical Analysis on a Digital Microfluidic Biochip," in *Proc. of VLSID*, 2011, pp. 171–176.
[7] J.-D. Huang *et al.*, "Reactant Minimization during Sample Preparation on Digital Microfluidic Biochips using Skewed Mixing Trees," in *Proc. of IEEE/ACM ICCAD*, 2012, pp. 377–384.
[8] D. Mitra *et al.*, "On-Chip Sample Preparation with Multiple Dilutions Using Digital Microfluidics," in *Proc. of IEEE ISVLSI*, 2012, pp. 314–319.
[9] Y.-L. Hsieh *et al.*, "A Reagent-Saving Mixing Algorithm for Preparing Multiple-Target Biochemical Samples Using Digital Microfluidics," *IEEE TCAD*, vol. 31, no. 11, pp. 1656–1669, Nov 2012.
[10] S. Kumar *et al.*, "Efficient Mixture Preparation on Digital Microfluidic Biochips," in *IEEE DDECS*, 2013, pp. 205–210.
[11] Y.-L. Hsieh *et al.*, "Design Methodology for Sample Preparation on Digital Microfluidic Biochips," in *Proc. of ICCD*, 2012, pp. 189–194.
[12] L. Luo and S. Akella, "Optimal Scheduling of Biochemical Analyses on Digital Microfluidic Systems," *IEEE TASE*, vol. 8, no. 1, pp. 216–227, 2011.
[13] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.
[14] K. Sugiyama *et al.*, "Methods for Visual Understanding of Hierarchical System Structures," *IEEE TSMC*, vol. 11, no. 2, pp. 109–125, 1981.
[15] Y. Zhao and K. Chakrabarty, "Synchronization of Washing Operations with Droplet Routing for Cross-Contamination Avoidance in Digital Microfluidic Biochips," in *Proc. of IEEE/ACM DAC*, 2010, pp. 635–640.
[16] A. Zoghbi and I. Stojmenović, "Fast Algorithms for Generating Integer Partitions," *Int. J. Comput. Math.*, vol. 70, no. 2, pp. 319–332, 1998.
[17] V. Ananthanarayanan and W. Thies, "Biocoder: A Programming Language for Standardizing and Automating Biology Protocols," *Journal of Biological Engineering*, vol. 4, pp. 1–13, 2010.