Efficient Mixture Preparation on Digital Microfluidic Biochips

Srijan Kumar*, Sudip Roy*, Partha P. Chakrabarti*, Bhargab B. Bhattacharya†, and Krishnendu Chakrabarty‡
*Department of Computer Science and Engineering, IIT Kharagpur, INDIA.
E-mail: {sudipr, srijankumar, ppchak}@cse.iitkgp.ernet.in
†Advanced Computing and Microelectronics Unit, ISI Kolkata, INDIA. E-mail: bhargab@isical.ac.in
‡Department of Electrical and Computer Engineering, Duke University, Durham, USA. E-mail: krish@ee.duke.edu

***Abstract*—Digital microfluidic biochips are recently being developed for on-chip implementation of biochemical laboratory assays. Existing mixing algorithms determine the mixing tree or mixing graph from a given target ratio of several biochemical fluids for on-chip mixture preparation. We present an algorithm to determine a reduced mixing tree by sharing the common subtrees within itself. We observe two transformations that preserve the semantics of the tree: (a) permutation of leaf nodes (input fluids/reagents) within the same level of a mixing tree, and (b) level-shifting of a leaf node to the next lower level by duplicating its appearance. The proposed algorithm utilizes both the intermediate droplets obtained after a split operation when a pair of identical subtrees are identified under permutation of leaf nodes at the same level. Simulation results for a large set of target ratios show that our algorithm reduces the mean values of the total number of mix-split steps, waste droplets and the number of mixer modules required for earliest completion by 16%, 29% and 12% over Min-Mix and by 22%, 34% and 20% over RMA, respectively. Moreover, it reduces the number of checkpoint insertions required for dynamic error recovery against incorrect mix-split steps during mixture preparation.**

I. Introduction

Recently, digital microfluidic (DMF) biochips are being studied by researchers of interdisciplinary fields to develop laboratory-on-a-chip for biochemical assay implementation [1]. A DMF biochip can implement wide-range of biochemical assays on an electrode array of a few square centimeters in size with low consumption of biochemical fluids. In such biochips, only integral unit-volumes (nl or pl) of discrete droplets of fluids are manipulated (i.e., dispensed from reservoirs, moved, merged, mixed, and split) by applying a sequence of electrical potentials to the electrodes coated with a hydrophobic material. Recent years have seen a surge in interest of design automation for DMF biochips [2], [3].

In many bioprotocols, *solution* or *mixture preparation* is an important preprocessing step for mixing two or more fluids in a given ratio of their concentration factors (CFs). For high-throughput applications, these solution preparation steps (dilution and mixing) are automated and integrated on-chip to create a variety of solution concentrations. Recently, several algorithms for automatic dilution and mixing of fluids have been reported in the literature [4], [5], [6].

In this paper, we present an algorithm to determine a mixing tree by sharing the common subtrees within itself. It may be observed that two transformations preserve the semantics of the tree: (a) permutation of leaf nodes (input fluids) within the same level of a mixing tree, and (b) level-shifting a leaf node to the next lower level by duplicating its appearance. Our proposed algorithm utilizes both the intermediate droplets obtained after a split operation, when a pair of identical subtrees are identified under permutation of leaf nodes at the same level. We simulated the existing mixing algorithms, **Min-Mix** [6] and **RMA** [4], along with our algorithm for a large set of target ratios. Simulation results show that the proposed algorithm reduces the mean of the total number of mix-split steps (T_{lb}) and waste droplets (W) by 16% and 29% over **Min-Mix** [6] and by 22% and 34% over **RMA** [4], respectively. This, in turn, reduces the consumption of expensive biochemical fluids to produce a target droplet of desired ratio compared to that by **Min-Mix** and **RMA**. It can reduce the minimum number of mixer modules required for earliest completion of mixture preparation (M_{lb}) [7] by 12% over **Min-Mix** and by 20% over **RMA**. Moreover, the reduced mixing tree obtained by our algorithm requires lesser number of checkpoint insertions for dynamic error recovery against incorrect mix-split steps during mixture preparation in comparison to that by existing mixing algorithms.

The remainder of the paper is organized as follows. Section II describes related prior work on automatic mixture preparation and the problem formulation. The proposed algorithm is presented in Section III with discussions in Section IV. Simulation results on a large set of target ratios are presented in Section V and finally, the paper is concluded in Section VI.

II. Automatic Mixture Preparation

In this paper, we consider the (1:1) mixing model, in which two equal-volume droplets of biochemical fluids are mixed and a subsequent balanced splitting is performed to obtain two unit-volume resultant droplets. One (1:1) mixing and the consecutive balanced splitting together is referred to as a single *mix-split step*. A *mixing tree* of height d is a binary tree representation of several (1:1) mix-split steps denoted by its non-leaf nodes. The root of the mixing tree has level d and the level of any other node in the tree is one less than the level of its parent. Several mix-split steps are deployed by the non-leaf nodes in a mixing tree to achieve the desired ratio of the constituent fluids at the root of the tree and in each such step, two unit-volume droplets are produced, one of which is discarded as a *waste droplet*. If a fluid x_1 of $CF = C_1$ is mixed with another fluid x_2 of $CF = C_2$, then the resultant CF for the mixed fluid is $\frac{C_1x_1+C_2x_2}{2}$. For the mixing tree of height d, the maximum error in CF of any constituent fluid in

978-1-4673-6136-1/13/$31.00

the target mixture with ratio a_1:a_2:...:a_N can be $\frac{1}{2^{d+1}}$, where $a_i \in \mathbb{Z}^+$, L is the sum of ratio integers and $L = 2^d$.

Let, in a mixing tree, T_{ms} be the total number of mix-split steps and W be the total number of waste droplets (without counting any target droplets). A scheduled mixing tree provides the sequence of mix-split steps with timing assignment and the mixer allocation to its non-leaf nodes. For a mixing tree, let M_{lb} be the minimum number of on-chip mixers required to complete the mixing tree execution within d mix-split cycles (lower-bound on the time of completion) as computed by Luo and Akella [7].

A. Prior Work and Motivation

In the literature, there exist three mixing algorithms to determine the mixing tree or mixing graph from a target ratio — **Min-Mix** [6], **RMA** [4] and **RSM** [8].

Min-Mix [6] uses d-bit binary fractions for the CFs corresponding to the target ratio to construct a mixing tree of height d after level-wise pairing of one leaf (non-leaf) node with another leaf (non-leaf) node. For an example target ratio 7:7:5:5:3:3:2 of seven fluids $x_1, x_2, \ldots, x_7$, the 5-bit binary fractions are: 0.00111_2, 0.00111_2, 0.00101_2, 0.00101_2, 0.00011_2, 0.00011_2 and 0.00010_2, respectively. Fig. 1(a) depicts a mixing tree determined by **Min-Mix** and it can be noted that $T_{ms} = 14$, $W = 13$ and $M_{lb} = 4$ for this MinMix-tree.

Another mixing algorithm, **RMA** determines the mixing tree with more disjoint subtrees by decomposing the algebraic expression corresponding to a target ratio. Hence, the RMA-tree has less probability of having common subtrees for using the intermediate waste droplets subsequently. For the same ratio 7:7:5:5:3:3:2, the RMA-tree is shown in Fig. 2.

RSM decomposes the algebraic expression for a target ratio such that new intermediate droplets are produced by mixing earlier intermediate droplets. Thus, it saves the requirement of input fluids in preparing a target mixture. We observe that **RSM** helps in reducing T_{ms}, W and requirement of input fluids when multiple target mixtures with different ratios are to be prepared. However, for a single target ratio, it determines a mixing graph instead of a mixing tree with common subtree(s) and sometimes it may require more mix-split steps compared to a mixing tree with common subtree(s).

We define a *common subtree* as the same subtree with same labels at the leaf-nodes having two occurrences in a mixing tree. It may be noted that, if there is a common subtree in the mixing tree, then the extra intermediate droplet (denoted by the root of that subtree) produced at a lower level can be used at a higher level instead of producing the same intermediate droplet again by performing the same set of mix-split steps. For the MinMix-tree shown in Fig. 1(b), there are two common subtrees of height one with algebraic expressions $\frac{x_1+x_2}{2}$ and $\frac{x_3+x_4}{2}$ (i.e., the number of common subtrees, N_{CS} is two). The roots of these two common subtrees are marked with directed edges indicating that both the split droplets of lower levels are used in higher levels. It explains the reuse of a waste droplet corresponding to the root of a common subtree at a lower level as the leaf node at a higher level. Note that, the identified common subtrees in a mixing tree reduces its T_{ms}, W and M_{lb}. For the above example, we get $T_{ms} = 12$, $W = 9$ and $M_{lb} = 4$ with $N_{CS} = 2$.

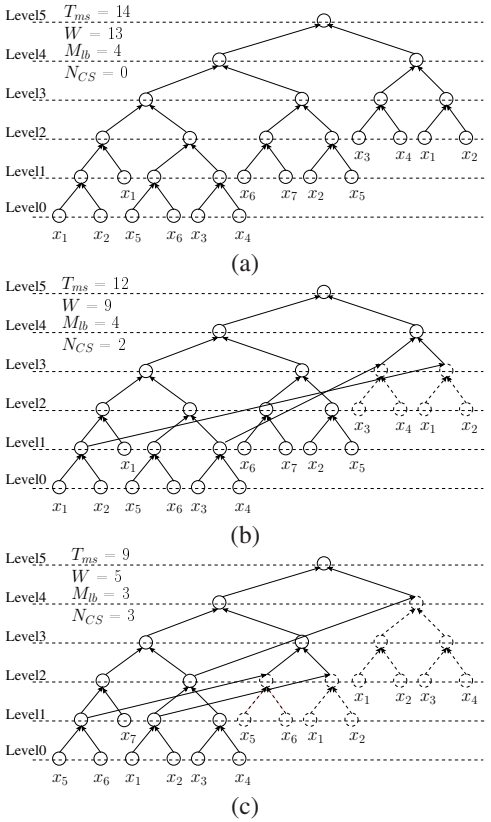

Fig. 1. For target ratio 7:7:5:5:3:3:2, (a) MinMix-tree (b) MinMix-tree with common subtrees, and (c) modified mixing tree obtained by permuting leaf nodes at the same level for different pairing.

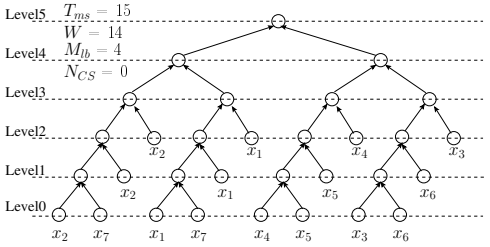

Fig. 2. RMA-tree for target ratio 7:7:5:5:3:3:2.

We observe two transformations that preserves the semantics of a mixing tree: (a) permutation of leaf nodes (input fluids) within the same level of a mixing tree, and (b) level-shifting of a leaf node to the next lower level by duplicating its appearance. With the first transformation, one can construct a different mixing tree with more (and/or longer) common subtrees as shown in Fig. 1(c), where $T_{ms} = 9$, $W = 5$ and $M_{lb} = 3$ with $N_{CS} = 3$. Another way of modifying a mixing tree is to shift a leaf node (variable) to its next lower level with duplicate occurrences at two leaf nodes for different pairings (level-shifting of a leaf-node). This is similar to addition of redundancy for logic sharing in circuit design. For a example target ratio 3:3:3:5:2 of five fluids $x_1, x_2, \ldots, x_5$, the 4-bit binary fractions are: 0.0011_2, 0.0011_2, 0.0011_2, 0.0101_2 and 0.0010_2, respectively. Fig. 3(a) depicts the mixing tree determined by **Min-Mix**, where $T_{ms} = 8$, $W = 7$ and $M_{lb} = 3$. However, this MinMix-tree has one common subtree of height one and if the waste droplet is reused within the mixing tree, we can get $T_{ms} = 7$, $W = 5$ and $M_{lb} = 2$ with

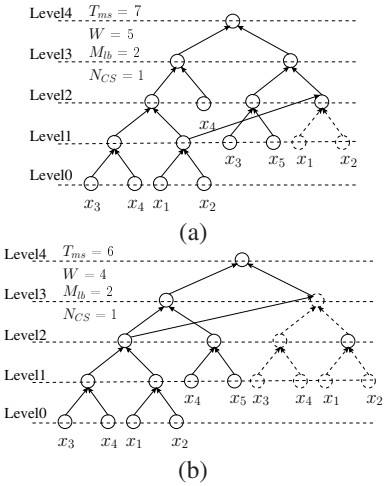

Fig. 3. For target ratio 3:3:3:5:2, (a) MinMix-tree with common subtrees, and (b) modified mixing tree obtained after level-shifting of x_4.

$N_{CS} = 1$. Now, by level-shifting of x_4 from level 2 to level 3 with duplicate occurrences, we get a different mixing tree with a longer common subtree as shown in Fig. 3(b), where $T_{ms} = 6$, $W = 4$ and $M_{lb} = 2$ with $N_{CS} = 1$. Hence, for efficient mixture preparation with a target ratio, our objective is to determine a mixing tree with the maximum possible number of common subtrees in it.

B. Problem Formulation

Because of the two possibilities of modifying a mixing tree to obtain more common subtrees within it, the problem of finding common subtrees is quite different from the well-known problem of finding common sub-expressions in multi-level logic optimization [9] or eliminating common sub-expressions during high-level synthesis of control-intensive designs [10]. Thus, the existing algorithms used in VLSI circuit design and compiler code optimization cannot be used directly. We propose a new algorithm (heuristic) to find common subtrees within a mixing tree under permutation of leaf nodes at the same level. The problem can be formulated as follows.

Given the precision level d, determine a sequence of mix-split steps to produce droplets of the target mixture with a given ratio. **Inputs**: (a) A set of N different fluids, $X = \{x_1, x_2, \ldots, x_N\}$, each with $CF = 100\%$, and (b) a target ratio $T = a_1 : a_2 : \ldots : a_N$ of the N fluids, where $L = \sum_{i=1}^{N} a_i = 2^d$, where (c) d as the height (precision level) of the mixing tree. **Output**: Mixing tree $\mathcal{M}$ of height d with common subtrees. We define an *optimal solution* as the *optimal* mixing tree with the minimum values of T_{ms}, W and M_{lb}.

III. Proposed Mixing Algorithm

It can be shown that finding optimal solution to the problem takes exponential time for computation. Given a target ratio, we propose a heuristic referred as Mixing Tree with Common Subtrees (**MTCS**), to construct the mixing tree with common subtrees under permutation of leaf nodes at the same level as presented in **Algorithm 1**. **MTCS** constructs the mixing tree by scanning the variables level-wise in bottom-up fashion and improving the possibility of having more and longer common subtree(s). It utilizes both the intermediate droplets obtained after a split operation when a pair of identical subtrees are identified. The proposed algorithm determines a mixing tree where some of the extra intermediate droplets (waste droplets) are used as the roots of common subtrees in that mixing tree.

Given a target ratio, first, we determine the *binary matrix* B of size $N \times d$, in which i^{th} row represents d-bit binary fraction for the CF of fluid x_i corresponding to the ratio component a_i. We define a *complete binary matrix*, C, which consists of B and another matrix called as *intermediate binary matrix*, I, which is used to represent intermediate droplets at different levels in the mixing tree. Initially, all the bits are set 0s in I. When a mixing operation is performed and both the intermediate droplets are used, the corresponding bits are made 1 in I. Each row of I consists of exactly two 1s indicating the two levels, at which both the intermediate droplets after split are used. We define a *list of unused intermediate droplets*, U, to store the waste droplets generated during mix-split operation and a hash table is used to maintain the entries of U. The j^{th} column of any matrix M (i.e., B or I) is denoted as M^j and the fundamental steps of **MTCS** are discussed here.

A. Generating AND-ed Matrix

First, an AND-ed matrix A of $(d-1)$ columns is formed after levelwise AND-ing of the right-most column of B with the remaining columns of B. In every next iteration, we form A taking ℓ_1^{th} column of B as the basis and AND-ing that with all other columns towards left of B and I, where ℓ_1 varies from 1 to $(d-1)$. Thus, A consists of two parts — A_B and A_I, representing the AND-ed matrices using B and I, respectively. For any column k, entries in A_B^k (or A_I^k) represent the leaf nodes that are present in both the levels ℓ_1 and $(k+\ell_1)$ of the underlying mixing tree. We denote $\mathcal{BS}(M^k)$ as the number of 1s in k^{th} column of any matrix M.

B. Finding Common Subtree(s)

After getting an AND-ed matrix A, we find the common subtree(s) from the matrix. In **MTCS**, two 1s in A^k are selected to pair (mix) using a scheme, referred as *Pair_Reagents*, given by **Algorithm 2**. For any column k, the 1s (nodes) in A^k are paired to get a mixed node. If $\mathcal{BS}(A_B^k)$ is odd, then a random node x_i is chosen from A_B^k to pair (mix) with a randomly chosen node x_j in A_I^k. A new row is added to I indicating a mixed node formed by pairing (mixing) x_i and x_j in A^k. Other nodes of A_B and A_I are randomly paired among themselves and new rows are added to I corresponding to the mixed nodes. The corresponding bits in B and I (for chosen

Algorithm 1 MTCS (N-fluid target ratio T, d)

1: Let $T = a_1{:}a_2{:}\ldots{:}a_N$ and $L = \sum_i a_i$, i.e., $L \leftarrow 2^d$.
2: Obtain $B(N, d)$ from T. $U \leftarrow \varnothing$. $\ell_1 \leftarrow 1$.
3: **while** $\ell_1 <= d-1$ **do**
4: Obtain AND-ed matrix A with ℓ_1^{th} column of B as the basis.
5: $\ell_2 \leftarrow \ell_1 + 1$.
6: **while** $\ell_2 <= d$ **do**
7: *Pair_Reagents*$(A, \ell_2 - \ell_1)$.
8: $\ell_2 \leftarrow \ell_2 + 1$.
9: $\ell_1 \leftarrow \ell_1 + 1$.
10: If $u \in U$ is a node found in B^p, remove u from U and store it. Make the corresponding bits to 0 in B^p.
11: **for** $k \leftarrow 1$ *to* d **do**
12: Construct $\mathcal{M}$ by level-wise pairing stored nodes at level k and using C^k.
13: Return $\mathcal{M}$.

Algorithm 2 *Pair_Reagents*(A, k)

1: **if** $\mathcal{BS}(A_B^k)$ is odd **then**
2: Choose a random node x_i for which there is an 1 in A_B^k.
3: **if** A_I^k contains 1s **then**
4: Choose a random node x_j for which there is an 1 in A_I^k and add x_j to U.
5: Pair (mix) x_i and x_j and make the corresponding bits to 0 in A^k.
6: Pair other nodes representing 1s in A^k randomly and store them. Make the corresponding bits to 0 in A^k and C.
7: Add the paired nodes representing 1s in A_I^k to U.

x_i and x_j) are made 0. Whenever a node in A_I is paired, it's corresponding waste droplet is inserted into the list U, so that it can be used later.

C. Constructing Mixing Tree with Common Subtree(s)

After getting common subtrees from AND-ed matrices, common nodes are found between U and B (which is already modified after the above two steps). If a node $u \in U$ is found in B^p, it is removed from U and stored. The corresponding bits are made to 0 in B^p. Other stored nodes and nodes in B^p are randomly paired (mixed) among themselves, and this is done for all levels of the underlying mixing tree. Finally, we obtain one node at level d, which is the root node of the underlying mixing tree containing common subtree(s).

Step-by-step formation of the mixing tree (MTCS-tree) for an example target ratio 7:7:5:5:3:3:2 is shown in Fig. 4. In this example, **MTCS** determines a mixing tree shown in Fig. 4(d), which is similar to that of Fig. 1(b) having same output parameters such as $T_{ms} = 9$, $W = 5$ and $M_{lb} = 3$ with $N_{CS} = 3$.

The time complexity of our proposed algorithm **MTCS** can be determined as $\mathcal{O}(Nd^2)$.

IV. Discussions on MTCS

A. Reducing M_{lb} for a Mixing Tree

Note that, **MTCS** may reduce M_{lb} of the mixing tree for a target ratio compared to that obtained by **Min-Mix** and **RMA**. This is because, T_{ms} is reduced by **MTCS**. If there is any common subtree in the mixing tree, then some mix-split steps are not required anymore and already produced intermediate (waste) droplets at any level can be used in higher level of the tree. Thus, the number of mix-split steps at the higher level decreases that reduces the value of M_{lb}. For a target ratio, the mixing tree with lower M_{lb} can be optimally scheduled to complete the mixing process within minimum time using less number of on-chip mixers (i.e., resources) [7]. Hence, the MTCS-tree may also reduce the area of mixture preparation chip for efficient mixing. For an example ratio 7:7:5:5:3:3:2, MTCS-tree (Fig. 4(d)) has $M_{lb} = 3$ that means for the earliest completion of the efficient mixture preparation the desired biochip layout with require only three on-chip mixers. Whereas the corresponding MinMix-tree (or RMA-tree) has $M_{lb} = 4$ that indicates the requirement for more area of the chip layout.

B. Comparison with *RSM [8]*

As mentioned earlier, for a single target ratio **RSM** [8] determines the mixing graph instead of a mixing tree. It uses

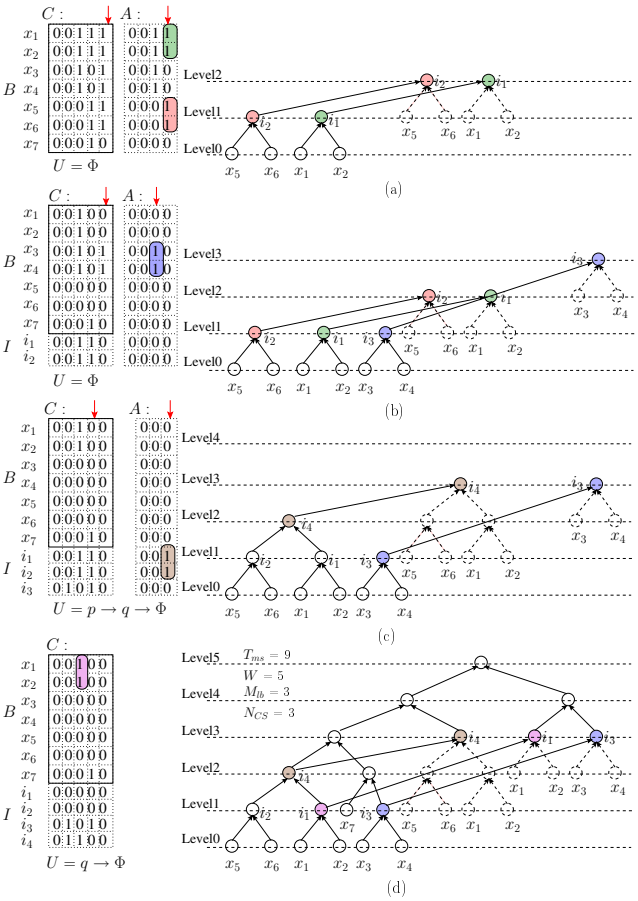

Fig. 4. Step-by-step mixing tree formation by **MTCS** for 7:7:5:5:3:3:2.

two intermediate droplets of two different levels to produce another intermediate CF at higher level and hence reduces the input requirement of reagent fluids. However, we observed that this may increase the total number of mix-split steps (T_{ms}) required and hence the M_{lb}. For an example ratio 7:14:11, the mixing graph obtained by **RSM** is shown in Fig. 5(a) having $T_{ms} = 8$, $W = 5$ and $M_{lb} = 2$, whereas the corresponding MTCS-tree shown in Fig. 5(b) has $T_{ms} = 7$, $W = 5$ and $M_{lb} = 2$ with $N_{CS} = 1$.

C. Check-point Assignment for Dynamic Error Recovery

During mixture preparation of several biochemical fluids, the volume of a split droplets is critical in maintaining the desired concentration of the mixture at every intermediate node. The concentration accuracy of the target mixture highly depends on the accuracy of every mix-split step denoted

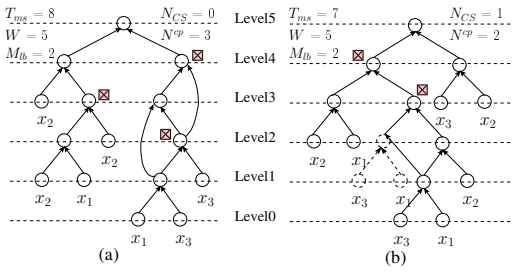

Fig. 5. For target ratio 7:14:11, (a) RSM-tree and (b) MTCS-tree.

TABLE I. MEAN (μ) AND VARIANCE (σ^2) OF T_{ms}, W AND M_{lb} DISTRIBUTIONS FOR **MIN-MIX** [6], **RMA** [4] AND **MTCS** ON 20000 RANDOM RATIOS OF N DIFFERENT FLUIDS WITH $L = 64$.

N	(μ, σ^2) of T_{ms}			(μ, σ^2) of W			(μ, σ^2) of M_{lb}		
	Min-Mix	**RMA**	**MTCS**	**Min-Mix**	**RMA**	**MTCS**	**Min-Mix**	**RMA**	**MTCS**
4	(8.69, 2.40)	(9.66, 3.97)	(7.41, 0.81)	(7.68, 2.40)	(8.66, 3.97)	(5.43, 0.85)	(1.95, 0.13)	(2.06, 0.24)	(1.83, 0.14)
5	(10.15, 2.80)	(11.24, 4.62)	(8.65, 1.12)	(9.15, 2.80)	(10.24, 4.62)	(6.50, 1.15)	(2.23, 0.20)	(2.44, 0.32)	(2.01, 0.05)
6	(11.61, 3.12)	(12.72, 5.06)	(9.87, 1.24)	(10.61, 3.12)	(11.72, 5.06)	(7.59, 1.28)	(2.58, 0.28)	(2.82, 0.38)	(2.19, 0.16)
7	(12.96, 3.99)	(14.08, 6.36)	(10.82, 1.43)	(11.96, 3.99)	(13.08, 6.36)	(8.32, 1.43)	(2.93, 0.32)	(3.21, 0.49)	(2.50, 0.26)
8	(14.19, 4.41)	(15.28, 7.02)	(11.84, 1.44)	(13.19, 4.41)	(14.28, 7.02)	(9.24, 1.38)	(3.25, 0.37)	(3.55, 0.57)	(2.84, 0.23)
9	(15.40, 4.80)	(16.52, 7.83)	(12.85, 1.46)	(14.40, 4.80)	(15.52, 7.83)	(10.17, 1.36)	(3.57, 0.42)	(3.91, 0.68)	(3.11, 0.22)
10	(16.57, 5.17)	(17.72, 8.61)	(13.84, 1.47)	(15.57, 5.17)	(16.72, 8.61)	(11.08, 1.35)	(3.90, 0.47)	(4.27, 0.81)	(3.39, 0.30)
11	(17.61, 5.58)	(18.77, 9.39)	(14.79, 1.48)	(16.61, 5.58)	(17.77, 9.39)	(12.00, 1.34)	(4.22, 0.54)	(4.61, 0.93)	(3.70, 0.38)
12	(18.69, 5.89)	(19.91, 9.92)	(15.75, 1.47)	(17.69, 5.89)	(18.91, 9.92)	(12.93, 1.32)	(4.55, 0.60)	(4.98, 1.05)	(4.03, 0.42)
Average	**(13.98, 4.24)**	**(15.10, 6.98)**	**(11.76, 1.33)**	**(12.98, 4.24)**	**(14.10, 6.98)**	**(9.25, 1.27)**	**(3.24, 0.37)**	**(3.54, 0.61)**	**(2.84, 0.24)**

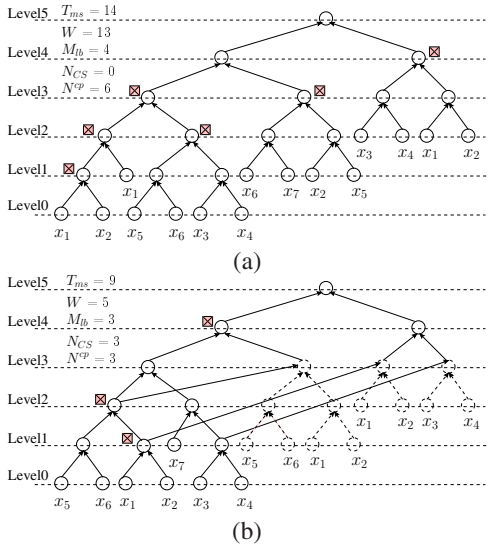

Fig. 6. For target ratio 7:7:5:5:3:3:2, assignment of check-points to (a) MinMix-tree and (b) MTCS-tree (checkpoint is indicated as a 'crossed-box').

by a non-leaf node in the mixing tree. Existing mixing algorithms [6], [4], [8] have been presented assuming the balanced or equal-volume splitting after each mix-split steps. For dynamic error recovery, recently cyber-physical digital microfluidic systems are being designed [11] that can check and verify the volume and/or concentration of intermediate droplets at certain checkpoints in the mixing tree. If there is an error in volume or concentration of an intermediate droplet, the associated mix-split operations are re-executed or some dynamic error recovery steps are performed to recover from the errors immediately and efficiently [11], [12], [13]. One such algorithm for dynamic error recovery is presented by Hsieh et al. [13]. They proposed an algorithm for assigning checkpoints and backup-points. We adopt their scheme to compute the total number of checkpoints required to assign in a mixing tree (N^{cp}), where $N^{cp} = \sum_i N_i^{cp} = \sum_i \lfloor p \times N_i^m \rfloor$, which is the summation of number of checkpoints assigned over all the levels (i varying from 0 to d). Here, N_i^m is the number of non-leaf nodes at level i in the mixing tree and p is assumed as a predefined percentage of N_i^m. For an example ratio 7:14:11, the checkpoint assignments according to [13] is shown in Fig. 5(a) and (b) for RSM-tree and MTCS-tree, respectively (checkpoint indicated as a 'red' color-ed 'crossed-box'). For another example ratio 7:7:5:5:3:3:2, Fig. 6(a) and (b) show the checkpoint assignments to the MinMix-tree and MTCS-tree, respectively. Note that N^{cp} is reduced in the MTCS-tree compared to the corresponding MinMix-tree or RSM-tree.

V. SIMULATION RESULTS

To evaluate our proposed algorithm, we carry out simulations with a large set of synthetic target ratios. In number theory and combinatorics, integer partition is the way of writing an integer as a sum of positive integers, regardless of order [14]. By convention, partitions are usually ordered from largest to 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 of L (where L is the sum of ratio integers) as the target ratios after keeping only N-component partitions for N fluids, where $N > 2$. This is because, in case of dilution ($N = 2$), the mixing tree has only one branch and it is not possible to have common subtree in such a tree. Thus, in considering target ratios of more than two fluids, we also keep those partitions for which the integers are set-wise co-prime, so that two different ratios do not eventually turn into the same ratio. Hence, for $L = 4$, only 2:1:1 and 1:1:1:1 are considered.

In practical bioprotocols, as many as 12 different fluids may need to be mixed to prepare a mixture for use in a biochip [15]. We simulate all three methods (**MinMix**, **RMA** and **MTCS**) for 533366 synthetic target ratios of N different fluids with $L = 64$ (i.e., $d = 6$), where $4 \leq N \leq 12$. Histograms of the distributions of T_{ms}, W, M_{lb} and N^{cp} for all three methods are shown in Fig. 7(a)-(d). It is observed that these distributions shift towards the origin in case of **MTCS** as compared to **MinMix** and **RMA**. These shifts reflect our claim regarding the improvement in the performance parameter values (i.e., T_{ms}, W, M_{lb} and N^{cp}). Hence, comparatively larger number of target ratios have less T_{ms}, W, M_{lb} and N^{cp} values.

We simulate **MinMix**, **RMA** and **MTCS** with 20,000 target ratios taken randomly from 533366 synthetic target ratios of N different fluids with $L = 64$ ($d = 6$). The mean (μ) and variance (σ^2) of the distributions of T_{ms}, W and M_{lb} for all three methods are estimated and Table I shows the variations of them as (μ, σ^2) pairs. The reduction in the values of (μ, σ^2) for **MTCS** is in agreement with the observations made through Fig. 7(a)-(c). **MTCS** improves 15.9%, 28.7% and 12.3% over **MinMix** and 22.1%, 34.4% and 19.8% over **RMA**, for average values of (μ) in distributions of T_{ms}, W and M_{lb}, respectively. It is also evident from Table I that the improvement in T_{ms}, W and M_{lb} increases with the increase in N.

We studied the performance of **MTCS** for varying height (d) of mixing trees as $3 \leq d \leq 6$ (where $L = 2^d$). However, here we have demonstrated the simulation results for six-height mixing trees (since $L = 64$) and found that maximum hight of a common subtree found by **MTCS** can be four. The histogram

for the distribution of N_{CS} found by **MTCS** in the mixing trees for 533366 target ratios (with $L = 64$) is shown in Fig. 7(e).

VI. Conclusions

We propose a mixing algorithm that yields a reduced mixing tree by eliminating common subtrees under permutation at the same level of the tree. Our algorithm reduces the total number of mix-split steps, waste droplets, and the minimum number of mixer modules required for earliest completion compared to the existing algorithms. This, in turn, reduces the number of checkpoints to be inserted for dynamic error recovery. Implementation of level shifting operation for further reduction of the mixing tree is currently under investigation.

Acknowledgment

This work of S. Roy was supported by Microsoft Corporation and Microsoft Research India under the Microsoft Research India PhD Fellowship Award.

References

[1] K. Chakrabarty and T. Xu, *Digital Microfluidic Biochips: Design and Optimization*. CRC Press, 2010.

[2] T.-W. Huang, J.-W. Chang, and T.-Y. Ho, "Integrated Fluidic-Chip Co-Design Methodology for Digital Microfluidic Biochips," in *Proc. of the IEEE ISPD*, 2012, pp. 49–56.

[3] C. C.-Y. Lin and Y.-W. Chang, "Cross-Contamination Aware Design Methodology for Pin-Constrained Digital Microfluidic Biochips," *IEEE TCAD*, vol. 30, no. 6, pp. 817–828, 2011.

[4] S. Roy, B. B. Bhattacharya, P. P. Chakrabarti, and K. Chakrabarty, "Layout-Aware Solution Preparation for Biochemical Analysis on a Digital Microfluidic Biochip," in *Proc. of the IEEE VLSID*, 2011, pp. 171–176.

[5] S. Roy, B. B. Bhattacharya, and K. Chakrabarty, "Optimization of Dilution and Mixing of Biochemical Samples using Digital Microfluidic Biochips," *IEEE TCAD*, vol. 29, no. 11, pp. 1696–1708, 2010.

[6] W. Thies, J. P. Urbanski, T. Thorsen, and S. Amarasinghe, "Abstraction Layers for Scalable Microfluidic Biocomputing," *Natural Computing*, vol. 7, no. 2, pp. 255–275, 2008.

[7] L. Luo and S. Akella, "Optimal Scheduling of Biochemical Analyses on Digital Microfluidic Systems," *IEEE TASE*, vol. 8, no. 1, pp. 216–227, 2011.

[8] Y.-L. Hsieh, T.-Y. Ho, and K. Chakrabarty, "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.

[9] V. K. Singh and A. A. Diwan, "A Heuristic for Decomposition in Multilevel Logic Optimization," *IEEE TVLSI*, vol. 1, no. 4, pp. 441–445, 1993.

[10] S. Gupta, M. Reshadi, N. Savoiu, N. Dutt, R. Gupta, and A. Nicolau, "Dynamic Common Sub-Expression Elimination during Scheduling in High-Level Synthesis," in *Proc. of the ISSS*, 2002, pp. 67–72.

[11] Y. Luo, K. Chakrabarty, and T.-Y. Ho, "Error Recovery in Cyberphysical Digital Microfluidic Biochips," *IEEE TCAD*, vol. 32, no. 1, pp. 59–72, Jan 2013.

[12] Y. Luo, T.-Y. Ho, and K. Chakrabarty, "Dictionary-based Error Recovery in Cyberphysical Digital-Microfluidic Biochips," in *Proc. of the IEEE ICCAD*, 2012, pp. 369–376.

[13] Y.-L. Hsieh, T.-Y. Ho, and K. Chakrabarty, "Design Methodology for Sample Preparation on Digital Microfluidic Biochips," in *Proc. of the IEEE ICCD*, 2012, pp. 189–194.

[14] A. Zoghbi and I. Stojmenović, "Fast Algorithms for Generating Integer Partitions," *Int. J. Comput. Math.*, vol. 70, no. 2, pp. 319–332, 1998.

[15] V. Ananthanarayanan and W. Thies, "Biocoder: A Programming Language for Standardizing and Automating Biology Protocols," *Journal of Biological Engineering*, vol. 4, 2010.

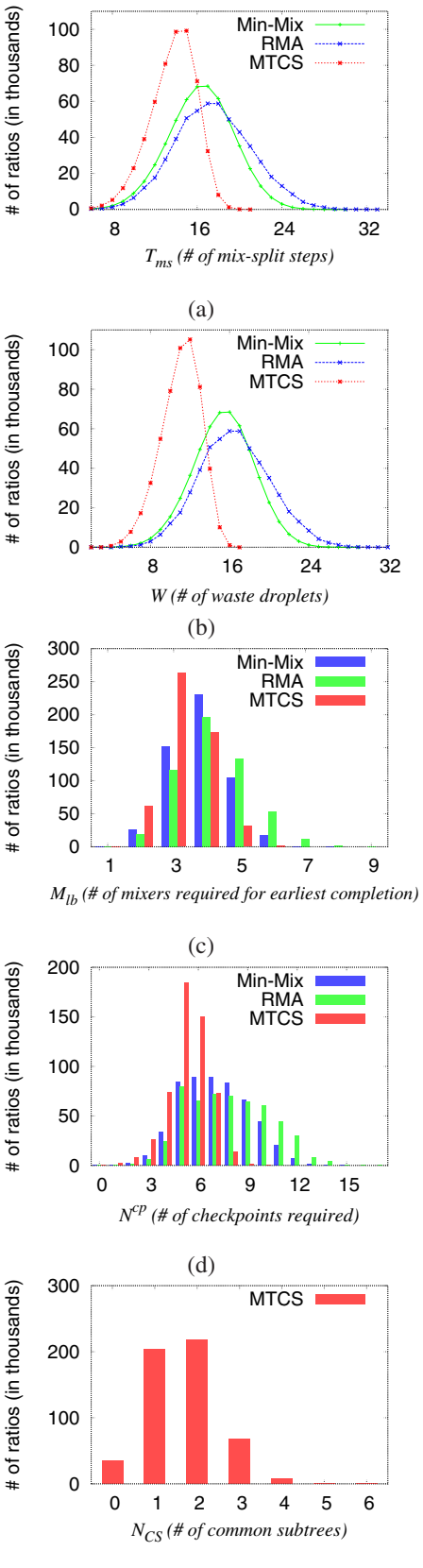

Fig. 7. For 533366 target ratios with $L = 64$ of N ($5 \leq N \leq 12$) different fluids, distributions of (a) T_{ms}, (b) W, (c) M_{lb}, and (d) N^{cp} by **MinMix**, **RMA** and **MTCS** and (e) histogram of N_{CS} for **MTCS**.