www.ietdl.org

Published in IET Computers & Digital Techniques
Received on 20th March 2013
Revised on 3rd July 2013
Accepted on 21st July 2013
doi: 10.1049/iet-cdt.2013.0053

ISSN 1751-8601

Sample preparation with multiple dilutions on digital microfluidic biochips

Sukanta Bhattacharjee, Ansuman Banerjee, Bhargab B. Bhattacharya

Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India
E-mail: sukanta_r@isical.ac.in

Abstract: Digital microfluidic (DMF) biochips offer a versatile platform for implementing several laboratory based biochemical protocols. These tiny chips can electrically control the dynamics of nanoliter volume of discrete fluid droplets on an electrode array by application of actuation patterns. One important step in biochemical sample preparation is dilution, where the objective is to prepare a fluid with a desired concentration factor. The protocols implemented on DMF biochips may require several different concentration values of a sample. In this study, the authors propose a scheme to produce such target droplets from a supply of an input sample and a buffer solution. Simulation results show a significant amount of savings in the number of mix-split steps and waste droplets in comparison to other methods for generating multiple concentration factors.

1 Introduction

Digital microfluidic (DMF) biochips are now being widely used in implementing several laboratory based biochemical protocols. These tiny chips, which consist of an electrode pattern, are capable of electrically controlling navigation and fluidic operations of nano/pico-liter volume of discrete droplets. Among various biochemical assays, mixing and dilution of fluids play a fundamental role in sample preparation.

In many biochemical protocols, solution preparation is a preprocessing step for mixing one or more fluids in a given ratio. The dilution of a biochemical sample/reagent is a special case of solution preparation, where a given fluid is diluted to a desired concentration factor using a diluent, that is, a buffer solution. Dilution is commonly used in biological studies to create variety of stock solutions on a microfluidic device. Moreover, one has to integrate multiple assay operations, such as detection, sample pre-treatment, sample mixing and dilution control on one single chip for an efficient implementation of a protocol.

There are two types of dilution methods: exponential/serial dilution and interpolated dilution. A serial dilution is essentially a series of dilutions used to reduce the concentration of a sample. The sample used in each step is taken from the diluted sample of the previous step. In each exponential dilution step, a unit volume sample/reagent droplet is mixed with a unit volume buffer droplet to obtain two unit volume droplets of half the concentration. If a sample/reagent (100% concentration) is recursively diluted by a buffer solution (0% concentration), the concentration of the sample/reagent becomes $(1/2^d)$ after d steps of mixing and balanced splitting. In each interpolated (1:1) mixing step, two unit volume droplets of concentrations C_1 and C_2 are mixed to obtain two droplets of concentration $(C_1 + C_2)/2$. Such dilution methods produce concentration values whose denominators can be expressed as integral powers of two. Thus, in the proposed method, the target concentration is approximated (rounded-off) as $x/2^n$, where $x \in \mathbb{Z}^+$, $0 \leq x \leq 2^n$ and $n \in \mathbb{Z}^+$, stands for the accuracy level. In other words, the target concentration factor can be achieved in n mix-split steps with a concentration error bounded above by $1/2^{n+1}$.

Traditionally, solution preparation was implemented using continuous-flow microfluidic chips [6–8]. Recently, several automated mixing/dilution schemes for solution preparation on a digital microfludic platform have also been reported [9–15]. Almost all previous work related to dilution with DMFs demonstrated various methods for obtaining a target droplet with the desired concentration when a sample and a neutral buffer solution are supplied.

In many real life applications, a variety of concentrations for the same sample is often required [16]. For example in Trinder's reaction, reference glucose solutions of different concentrations are required to be prepared with dilutions 100, 300 and 800 mg/dl standards (Sigma, 16–11) in deionised water [17]. In protein crystallisation, for each stock solution, many different target concentrations are needed [16]. Many other assays including PCR, Bradford protein assay, Emerson–Trinder enzymatic reaction also require multiple target dilutions.

In this paper, we present a novel strategy of generating multiple target droplets with different concentrations. Compared with three earlier methods [12, 14, 15] our scheme significantly reduces both the number of mix-split steps and waste droplets. Given a supply of sample and

buffer solution, a set of desired target concentration values can be obtained by a series of (1:1) mix-split steps, while at each step, an intermediate droplet is allowed to mix with a sample, buffer, or with a previously generated droplet. We report an efficient heuristic to combine some of the waste droplets produced in the process so as to achieve the target set with a reduced number of mix-split steps and waste production.

The organisation of the remainder of the article is as follows. We present an overview of DMF biochips and discuss related prior work in Section 2. Section 3 presents an outline of the proposed method for generating multiple target concentrations. Section 4 presents simulation results with a comparative study with other existing approaches. An architectural layout for possible implementation of the proposed scheme is suggested in Section 5. Section 6 concludes the presentation.

2 Background and prior art

2.1 Basics of biochips

Although continuous flow microfluidic biochips are widely used for implementing various biochemical analyses or assays [6–8], they suffer from lack of programmability. The later generation of DMF biochips offers dynamic reconfigurability and architectural scalability and hence, they can be used as programmable 'microfluidic processors'.

A DMF-based lab-on-a-chip uses electrical actuation to manipulate discrete droplets of nanoliter volume of fluids on a two-dimensional (2D) electrode array. A unit cell in the array includes a pair of electrodes that act as two parallel plates. The bottom plate contains a patterned array of individually controlled electrodes, and the top plate acts as a common ground electrode. A droplet is sandwiched between the two plates and rests on the hydrophobic surface of an electrode. A droplet can be moved to an adjacent electrode by activating it by a control voltage (above some threshold value), and at the same time, deactivating the electrode on which the droplet was sitting. This electronic method of wettability control creates interfacial tension gradients that move the droplet to the charged electrode. By varying the patterns of control voltage activation, several fluidic operations, such as merging, splitting, mixing and dispensing of droplets, can be executed. In addition to electrodes, optical detectors such as light-emitting diodes and photo-detectors are also integrated on the DMF biochips for monitoring colorimetric bioassays. The various fluidic modules include mixers, splitters, detectors, waste reservoirs, dispensers and heaters. Detailed description of a DMF biochip and the four fundamental fluidic operations (dispensing, transporting, mixing and splitting) have been elaborated in the literature [2, 4, 5, 18].

2.2 Prior art

A DMF biochip typically manipulates discrete fluid droplets on a uniform 2D array of identical electrodes. Thus, the volume of a merged droplet is usually an integral multiple of that of a single droplet (unit volume). It is a challenge to achieve a desired concentration factor ($\mathcal{CF}$) that minimises the number of mix-split and/or waste droplets. A single target mixing algorithm based on bit-scanning (Min-Mix) method was proposed by Thies *et al.* [19] for mixing two or more sample/reagent fluids at any given ratio assuming the (1:1) mixing model. In the special case of diluting a sample, the Min-Mix method first approximates the target $\mathcal{CF}$ as a n-bit binary string depending on the accuracy level if the maximum allowable error in target concentration is $1/2^{n+1}$, and then scans the bits from right-to-left to decide on the sequence of mix-split steps. The current bit determines whether the sample or the buffer droplet is to be mixed with the most recently produced droplet. This method has the advantage that no earlier droplets (hence no storage unit) are required; only the current droplet along with the initial sample or the buffer is required for the next step. However, it produces one waste droplet at each mix-split step except the last one. It completes the dilution process in at most n mix-split steps with a maximum error of $1/2^{n+1}$ in the target $\mathcal{CF}$. Another method known as DMRW [20] generates a single dilution of a sample using binary search. This approach reduces the number of waste droplets compared with the Min-Mix method.

Recently, a reagent saving mixing algorithm for multiple target sample preparation has been proposed by Hsieh *et al.* [14]. In the case of dilution, it actually shares the same dilution tree generated by the Min-Mix method. For example, the dilution tree for the target set $\mathcal{L} = \left\{\frac{11}{64}, \frac{21}{64}, \frac{31}{64}, \frac{41}{64}, \frac{51}{64}\right\}$ is shown in Fig. 1, where the coloured node represents a target concentration and a bullet symbol • indicates an intermediate waste droplet.

A method for generating a set ($\mathcal{L}$) of multiple concentrations without using any intermediate storage, was reported by Mitra *et al.* [12]. In this method, a dilution graph is constructed; next, a complete weighted directed graph $G=(V, E)$ is created, where $V = \{\mathcal{L} \cup \{(0/2^n)\}\}$ and $E = V \times V$. The cost of an edge $(i, j) \in V \times V$ is set as the cost of the shortest path between i and j in the dilution graph. In other words, the cost of an edge (i, j) represents the minimum number of mix-split steps required to generate concentration j from concentration i. Thus, a shortest path that starts from $0/2^n$ and includes all concentration values of $\mathcal{L}$ represents the sequence of target concentrations to be generated. As an example, the resultant sequence of the targets for $\mathcal{L}$ is

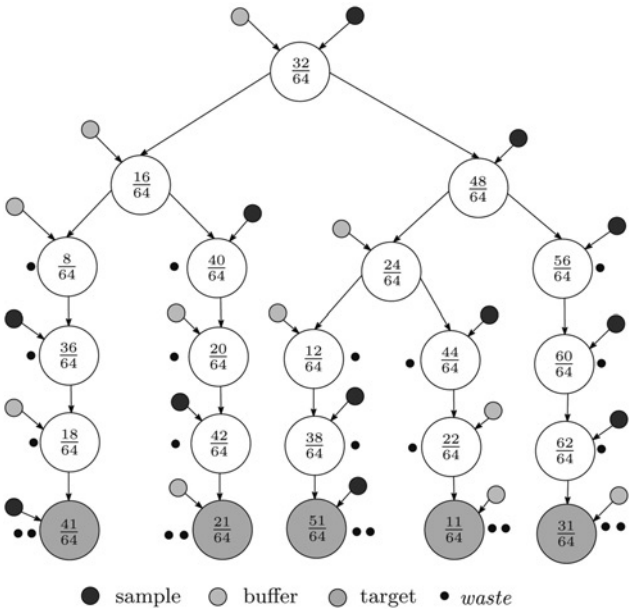

Fig. 1 *Dilution tree generated by Hsieh et al. [14] method*

doi: 10.1049/iet-cdt.2013.0053

$$\frac{0}{64} \xrightarrow{6} \frac{31}{64} \xrightarrow{3} \frac{51}{64} \xrightarrow{4} \frac{11}{64} \xrightarrow{3} \frac{41}{64} \xrightarrow{3} \frac{21}{64}$$

where an arrow label indicates the number of mix-split steps needed to reach the successor from its predecessor. Fig. 2 shows the sequence of droplets produced along the shortest path going through the target concentrations.

The Min-Mix method was generalised for multiple concentration sample preparation [1] and also for generating a single target concentration with a multiple demand [13]. A reactant minimising sample preparation approach was reported by Huang *et al.* [15] and Chiang *et al.* [21].

3 Proposed methodology

In this section, we discuss the proposed scheme for generating multiple target concentrations along with an illustrative example. As discussed earlier, the concentration factors ($\mathcal{CF}$) are rounded off as n-bit fractional binary numbers and their range excluding the supplied values $\mathcal{CF} = 0$ and $\mathcal{CF} = 1$, can be represented as $1/2^n$, $2/2^n$, …, $(2^n - 1)/2^n$. The problem can now be stated as follows: Given a set of k distinct targets $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$, where $1/2^n \le t_i \le (2^n - 1)/2^n$, for $i = 1, 2, \ldots, k$, generate $\mathcal{T}$ using only sample ($2^n/2^n$) and buffer ($0/2^n$) droplets, such that the number of waste droplets is minimised. The number of (1:1) mix-split operations should also be minimised as a secondary objective. Before discussing our approach, we first present a few useful definitions and some important observations.

3.1 Dilution tree

Definition 1: A dilution tree of order n is defined as a full binary tree [22] with $2^n - 1$ nodes representing the concentration factors $1/2^n$, $2/2^n$, …, $(2^n - 1)/2^n$.
In a dilution tree, the node labelled as $2^{n-1}/2^n$ is the root node. The left child of an internal node ($x/2^n$) represents the concentration generated by mixing the current droplet ($x/2^n$) with buffer ($0/2^n$). Similarly, the right child of $x/2^n$ represents the concentration generated by mixing the current droplet with sample ($2^n/2^n$).

Example 1: Fig. 3 shows the dilution tree of order 5.
An useful observation for generating a target set from a dilution tree is as follows.

Observation 1: In order to produce a target set, exploration of only the relevant portion of dilution tree is sufficient.

In the above observation 'relevant portion' means the minimum subset of nodes in the dilution tree that contains all the targets in a target set. The leaf nodes of the tree represent the targets with two droplets generated therein. Some of the targets may as well be generated in the internal nodes of the dilution tree. If an internal target node has only one child, one need not regenerate that target because one waste droplet is generated in that node. However, if an internal target node has two children, then it should be regenerated. A simple example illustrates Observation 1.

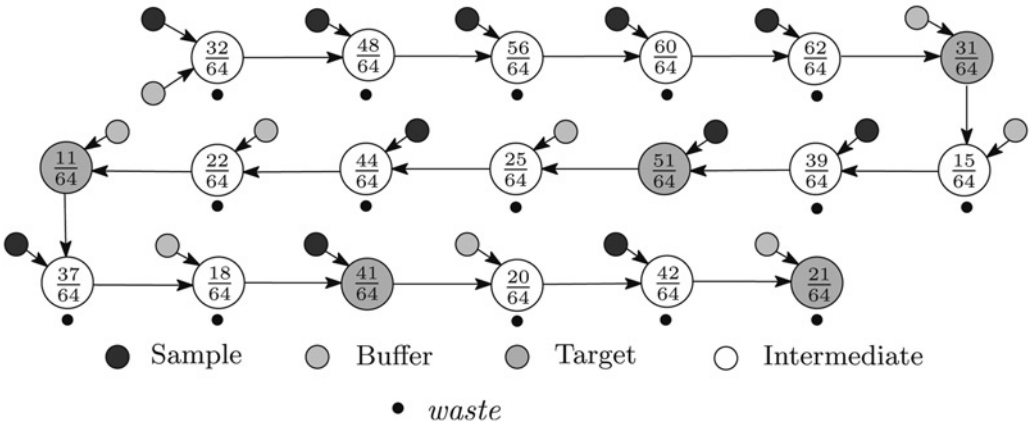

Fig. 2 *Sequence of mix-split steps for generating $\mathcal{L}$ by the method of Mitra et al. [12]*

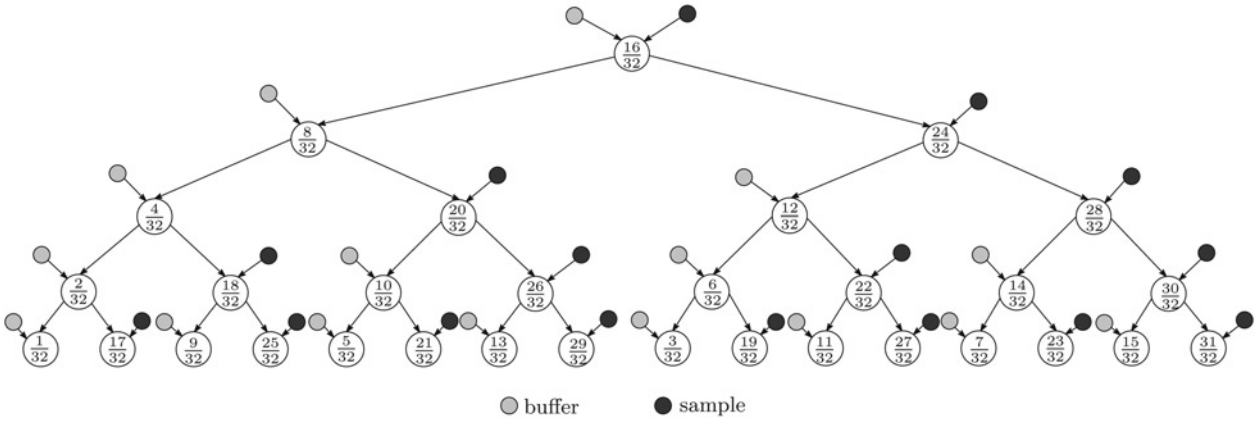

Fig. 3 *Dilution tree of order 5*

Example 2: Consider a target set

$$\mathcal{T} = \left\{\frac{5}{64}, \frac{6}{64}, \frac{8}{64}, \frac{12}{64}, \frac{15}{64}, \frac{20}{64}, \frac{26}{64}, \frac{31}{64}, \frac{34}{64}, \frac{45}{64}, \frac{49}{64}, \frac{57}{64}, \frac{59}{64}, \frac{62}{64}, \frac{63}{64}\right\}$$

to be generated from initial concentrations 0/64 and 64/64. The relevant portion of the dilution tree of order 6 is shown in Fig. 4. The coloured nodes represent target concentrations. The number of waste droplets (•) is shown in Fig. 4 along with the tree nodes. One can easily observe from Fig. 4 that only the target concentrations 8/64 and 62/64 need to be regenerated. The total number of (1:1) mix-split steps is found to be 37 along with production of 13 waste droplets.

3.2 Pruning on the dilution tree

We now present another useful observation that motivates us to develop a pruning heuristic on the dilution tree.

Observation 2: When a target set is generated using the dilution tree, some waste droplets are generated. Moreover, combining two waste droplets may also produce some target droplets.

There are many waste droplets generated in the dilution tree. Not all waste droplets can be used to generate target droplets. The following example shows how waste droplets can be reused to produce some elements of $\mathcal{T}$ in the dilution tree of Fig. 4.

Example 3: Consider the dilution tree of Fig. 4 representing the target set $\mathcal{T}$ of Example 2. The subset of target concentration factors {12/64, 20/64, 26/64, 34/64} should not be used for generating other target concentrations as each target concentration has only one waste droplet generated; however the intermediate waste droplets with concentrations {4/64, 10/64, 30/64, 36/64, 44/64, 50/64, 52/64, 54/64, 56/64} can be used for generating other targets. Moreover, for each of the leaf target concentrations {5/64, 6/64, 15/64, 31/64, 45/64, 49/64, 57/64, 59/64, 63/64} two droplets are generated; one of them can be used for generating other target concentrations.

A subset of target droplets can be generated in various ways from waste droplets. For example, the target droplet 31/64 can be generated by combining waste droplets of 52/64 and 10/64. Similarly waste droplets of 4/64 and 6/64 can be used to generate the target 5/64.

One therefore has to decide which target should be generated first and which set of waste droplets is be used in the process. In order to facilitate this decision, an integer cost is associated with each waste droplet. An intermediate target droplet will have at most one waste droplet, so using it for generating another target is not worthwhile. In the case where a leaf node represents a target, one of them can be used to produce another target droplet. Before defining the cost of waste droplets, we introduce some terminologies.

Definition 2: The target_path for a leaf target (t_{leaf}) concentration is defined as the path following the parent of t_{leaf} until another target node or a node with two children is reached.

Definition 3: A variable 'which_target_path' associated with a concentration factor ($\mathcal{C}$) denotes the target concentration whose target_path contains $\mathcal{C}$. A simple example explains the above definitions.

Example 4: In Fig. 4, the target_path for 57/64 is (57/64, 50/64, 36/64), target_path for 5/64 is (5/64, 10/64) and target_path for 62/64 is null as it is an internal node. For each $\mathcal{CF}$ in {57/64, 50/64, 36/64}, 'which_target_path' is 57/64 because each of them belongs to the target_path of 57/64.

We now define the cost of a leaf target node in the dilution tree of a target set. Informally, the cost of a target leaf node t_{leaf} is defined as the number of mix-split steps that can be saved if t_{leaf} can be generated from other waste droplets. More formally, the cost of a waste droplet is assigned as the cost of 'which_target_path' assigned to that waste. Assignment of these costs is performed by executing Algorithm 1 (see Fig. 5). An example of cost assignment is

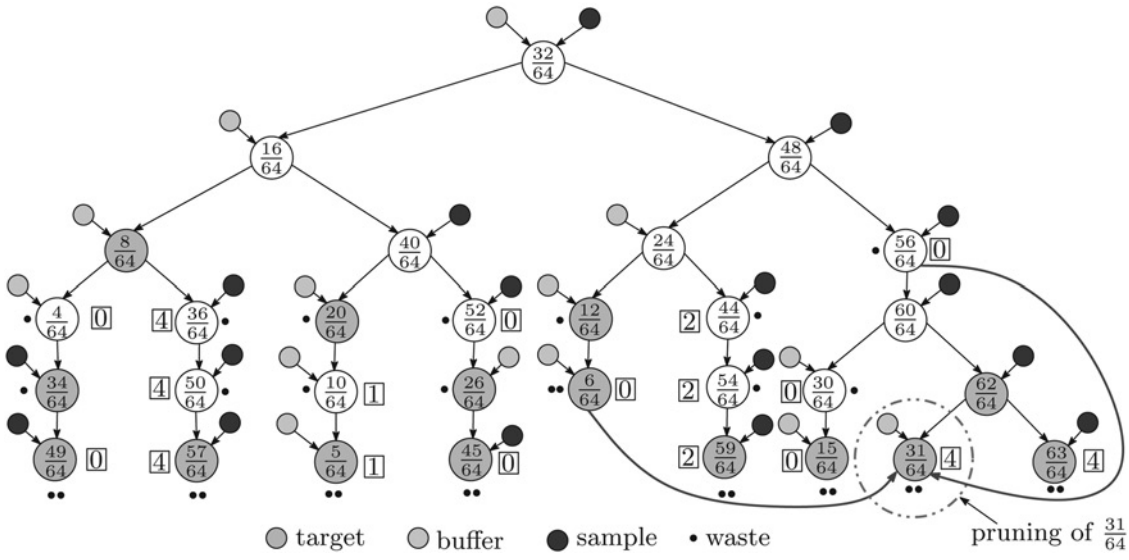

Fig. 4 *Dilution tree for target set $\mathcal{T}$*

doi: 10.1049/iet-cdt.2013.0053

Algorithm 1

Input: Dilution tree $\mathcal{DT}$ for given a target set $\mathcal{T}$
Output: Cost of waste droplets and *which_target_path* of *waste* droplets
Initially set the cost of all droplets to 0 and *which_target_path* to dummy concentration having cost 0;
foreach *leaf node* (t_{leaf}) *of* $\mathcal{DT}$ **do**
  Set $extra\_cost = 0$;
  Set $count$ = the number of nodes traversed using parent pointer from t_{leaf} until an intermediate target node or an intermediate concentration with two children is found;
  if *an intermediate concentration* ($\mathcal{IC}$) *with two children is found* **then**
    if $\mathcal{IC}$ *is target* **then**
      Set $extra\_cost$ = number of nodes from $\mathcal{IC}$ to root of $\mathcal{DT}$;
  Set the *which_target_path* to t_{leaf} for each node along the traversed path from t_{leaf};
  Set the cost of t_{leaf} to $(count + extra\_cost)$;

Fig. 5 *Cost assignment*

shown in Fig. 4 where the number in the square box represents the cost of a waste droplet. The following example illustrates how Algorithm 1 (see Fig. 5) assigns cost to the leaf targets and computes 'which_target_path' for the waste droplets in dilution tree of a given target set.

Example 5: Let us consider the cost assignment to leaf target 57/64 in Fig. 4. We start traversing through the parent pointer of 57/64 until an intermediate target concentration or a node having two children is found. We therefore stop when 8/64 is reached, which is an intermediate target node. The number of nodes traversed is 2, that is, count = 2. extra_cost is set to 2 as number of concentrations from 8/64 to the root concentration 32/64 is 2. Hence, the cost of 57/64 is 4. The target_path of 57/64 is (57/64, 50/64, 36/64) and 'which_target_path' for each the concentrations {57/64, 50/64, 36/64} is 57/64.
Let us interpret the cost of 57/64. If the concentration 57/64 is produced from other waste droplets, the right subtree of 8/64 need not be generated. Moreover, the target concentration 8/64 has one waste droplet. Hence, 8/64 need not be regenerated, and this case we can save four mix-split steps.

After assigning cost to waste droplets, a cost matrix $(\mathcal{CM})_{m\times m}$ is defined, where m is the number of waste droplets that can be used from a relevant portion of the dilution tree. The ith row of $\mathcal{CM}$ represents waste$_i$, jth column of $\mathcal{CM}$ represents waste$_j$ and $\mathcal{CM}[i, j]$ represents the number of mix-split step(s) that can be saved by mixing waste$_i$ and waste$_j$ while producing a leaf target droplet. Algorithm 2 (see Fig. 6) shows the essential steps for constructing the cost matrix. Algorithm 2 combines waste$_i$ and waste$_j$ to check whether a leaf target concentration (t_{leaf}) can be produced. If so, it ensures that neither of waste$_i$ or waste$_j$ arrives from target_path of t_{leaf}.

Based on Algorithm 1 (see Fig. 5) and Algorithm 2 (see Fig. 6), one can prune the dilution tree. The pruning strategy is described in Algorithm 3 (see Fig. 7). It first assigns the cost and sets 'which_target_path' for each waste droplet. Next, the cost matrix $\mathcal{CM}$ is constructed. This is followed by choosing, in a greedy manner, the largest cost target droplet that can be pruned. It may be noted that the maximum may occur multiple times in $\mathcal{CM}$ for different (i, j) pairs. Algorithm 3 (see Fig. 7) chooses the pair (i, j) where the cumulative cost of waste$_i$ and waste$_j$ is minimised. Before pruning, it sets the cost of 'which_target_path' for waste$_i$ and waste$_j$ to 0. It prevents leaf targets from pruning that have waste$_i$ or waste$_j$ in their target_path. It may so happen that waste$_i$ or waste$_j$ does not belong to any *target_path* of a leaf target. In this case, setting its cost to 0 has no effect as it has only one waste droplet and it is used in pruning. After pruning, the set of waste droplets is modified and the cost matrix is recalculated for further pruning. The following example explains the steps of Algorithm 3(see Fig. 7).

Example 6: The initial cost matrix for the waste droplet set {4/64, 5/64, 6/64, 10/64, 15/64, 30/64, 31/64, 36/64, 44/64, 45/64, 49/64, 50/64, 52/64, 54/64, 56/64, 57/64, 59/64, 63/64} has a maximum entry of 4. There are several valid waste droplet combinations, which generate targets of cost 4. Table 1 shows the different alternatives of cost 4 targets. The highlighted row shows the minimum cumulative cost of two waste droplets that produce 31/64 of cost 4. The dilution tree after pruning 31/64 is shown in Fig. 8. The concentrations 56/64 and 6/64 have no waste droplet, so they are removed from further consideration along with 31/64. Now the new waste set becomes {4/64, 5/64, 10/64, 15/64, 30/64, 36/64, 44/64, 45/64, 49/64, 50/64, 52/64, 54/64, 57/64, 59/64, 63/64} and the cost matrix is recalculated. The new cost matrix has only one entry of cost 4. Hence 52/64 and 63/64 are combined and target concentration 57/64 is generated. The dilution tree after pruning 57/64 is shown in Fig. 9. The concentrations 52/64, 63/64, 57/64, 50/64, 36/64 are removed and the modified waste droplet

Algorithm 2

Input: Dilution tree $\mathcal{DT}, wastes, m$
// *wastes* is an m element array of waste droplets.
Output: Cost matrix $(\mathcal{CM})_{m\times m}$
Initialise $(\mathcal{CM})_{m\times m}$ by 0;
foreach *(i,j) where* $1 \le i \le m, 1 \le j < i$ **do**
  Let $result = mix\_split(wastes[i], wastes[j])$;
  if $result$ *is leaf target concentration and which_target_path of* $wastes[i]$ *or* $wastes[j]$ *is not equal to* $result$ **then**
    Set $\mathcal{CM}[i, j]$ = cost of $result$;

Fig. 6 *Generate cost matrix*

Algorithm 3

Input: Dilution tree $\mathcal{DT}$
Output: Pruned dilution tree.

Assign cost to each waste droplets. (Algorithm 1);
Create cost matrix ($\mathcal{CM}$). (Algorithm 2);
while *true* **do**
  Find the maximum element in $\mathcal{CM}[i,j]$ where $cost(wastes[i]) + cost(wastes[j])$ is minimum;
  // $cost(c)$ returns the cost of concentration c
  Assume that the maximum occurs in $\mathcal{CM}[i,j]$;
  if $\mathcal{CM}[i,j] \leq 1$ **then**
    return;
  Set $target_i$ to *which_target_path* of $wastes[i]$;
  Set $target_j$ to *which_target_path* of $wastes[j]$;
  Set cost of $target_i$ and $target_j$ to 0;
  Prune the subpath containing the removed target as leaf node from $\mathcal{DT}$;
  Remove $wastes[i]$, $wastes[j]$ and the subpath that contains the target from $wastes$;
  Construct cost matrix ($\mathcal{CM}$) (Algorithm 2) for modified waste droplet set;

Fig. 7 *Pruning*

set becomes {4/64, 5/64, 10/64, 15/64, 30/64, 44/64, 45/64, 49/64, 54/64, 59/64}. The new cost matrix has all entries 0 and thus no further pruning is possible. Fig. 9 shows the final pruned tree. One can easily verify that after pruning, all target droplets have been generated. However, it may so happen that after pruning, there are still some targets left. In such a case, the dilution tree for remaining target droplets has to be generated separately and this process continues until all targets are produced.

In the above example, the total number of (1:1) mix-split steps required is 26 and 5 waste droplets are generated, that is, we obtain 30% savings in mix-split steps and 62% savings for waste droplets over the base approach when pruning is applied on the dilution tree. The running time of the pruning algorithm is quadratic in the number of waste droplets. One need not explicitly calculate the cost matrix every time if the entries for waste droplets are sorted initially. For each leaf target sorted in the decreasing order

Table 1 Different alternatives for maximum entry (4) in the initial cost matrix

$waste_1$	$waste_2$	cost ($waste_1$)	cost ($waste_2$)	total cost	target	cost (target)
5/64	57/64	1	4	5	31/64	4
52/64	10/64	0	1	1	31/64	4
57/64	6/64	4	0	4	31/64	4
56/64	**6/64**	**0**	**0**	**0**	**31/64**	**4**
59/64	4/64	2	0	2	31/64	4
56/64	59/64	0	2	2	57/64	4
63/64	52/64	4	0	4	57/64	4

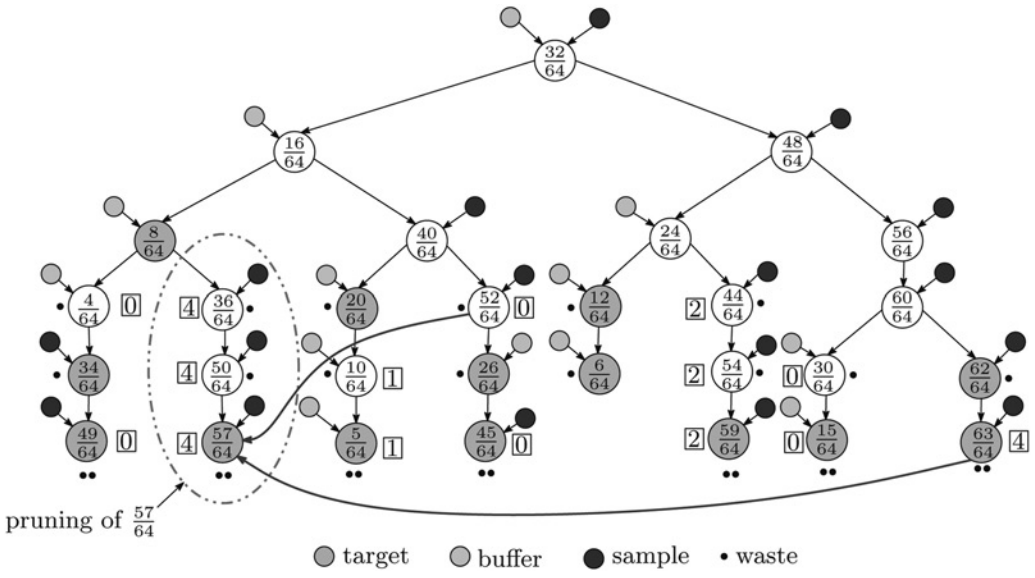

Fig. 8 *Tree after pruning 31/64*

doi: 10.1049/iet-cdt.2013.0053

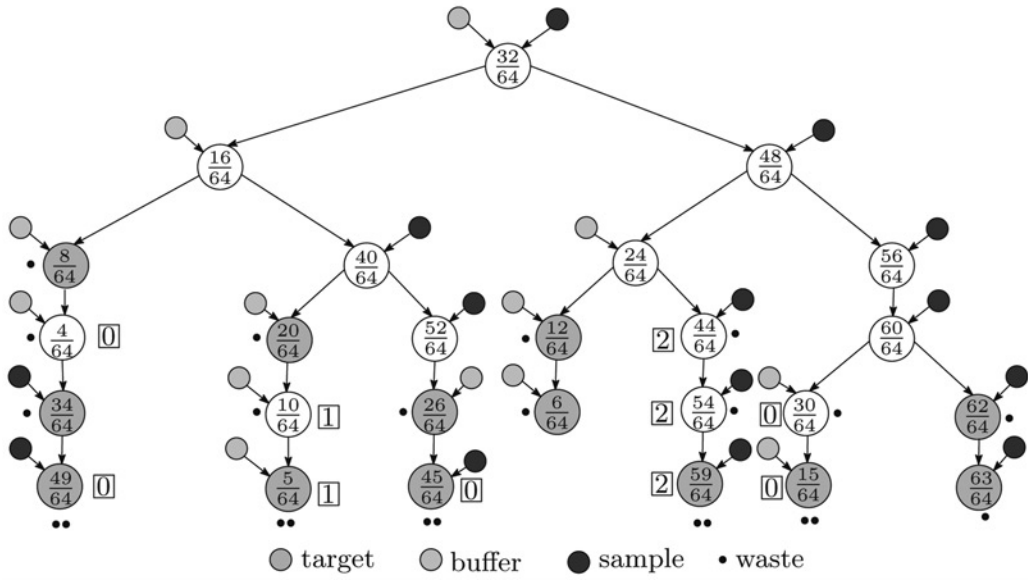

Fig. 9 *Final pruned tree*

of cost, one can find all valid combinations of waste droplets that generate a particular leaf target in $O(\mathcal{W})$ time and choose the one with least cumulative cost; here $\mathcal{W}$ denotes the number of waste droplets that can be considered for generating a particular leaf target in the dilution tree. Note that pruning can occur only for leaf target nodes. For a dilution tree with $\mathcal{T}_{\text{leaf}}$ leaf targets, each pruning step therefore requires $O(\mathcal{T}_{\text{leaf}} \times \mathcal{W})$ operations in the worst case.

4 Simulation results

To demonstrate the effectiveness of the proposed method, we have compared it with those reported by Hsieh *et al.* [14] (Method-1), Mitra *et al.* [12] (Method-2) and Huang *et al.* [15] (Method-3). We have implemented a prototype tool to perform simulation in Python. For simulation purposes, we have chosen $n = 10$, that is, each target concentration is rounded off to a 10-bit fractional number. We have considered 5000 random target sets of different sizes (10, 30, 50, 100, 200). For each of the 5000 target sets, we calculated the savings (%) in the number of mix-split steps and in the waste count of the proposed method with respect to three earlier methods. The histogram plots of savings are shown in Figs. 10 and 11. In these plots, the horizontal axis denotes the %-savings and the vertical axis shows the number of target sets with a particular value of %-savings. We use the following expression for computing the % savings

$$\text{mix\_split}_{\text{savings}} = \frac{\text{mix\_split}_{\text{Method}-i} - \text{mix\_split}_{\text{proposed}}}{\text{mix\_split}_{\text{Method}-i}} \times 100\% \quad (1)$$

$$\text{waste}_{\text{savings}} = \frac{\text{waste}_{\text{Method}-i} - \text{waste}_{\text{proposed}}}{\text{waste}_{\text{Method}-i}} \times 100\% \quad (2)$$

The histograms of %-savings in the number of (1:1) mix-split steps (Fig. 10) reveal that the proposed method has non-negative savings, which increases with the cardinality of the target set. The proposed method outperforms all three earlier sample preparation methods w.r.t the number of mix-split steps and waste droplets. It may be observed that our method has no savings in comparison with Method-1 and Method-2 for small number of target sets (10); however when the target set size is increased to 30, the proposed

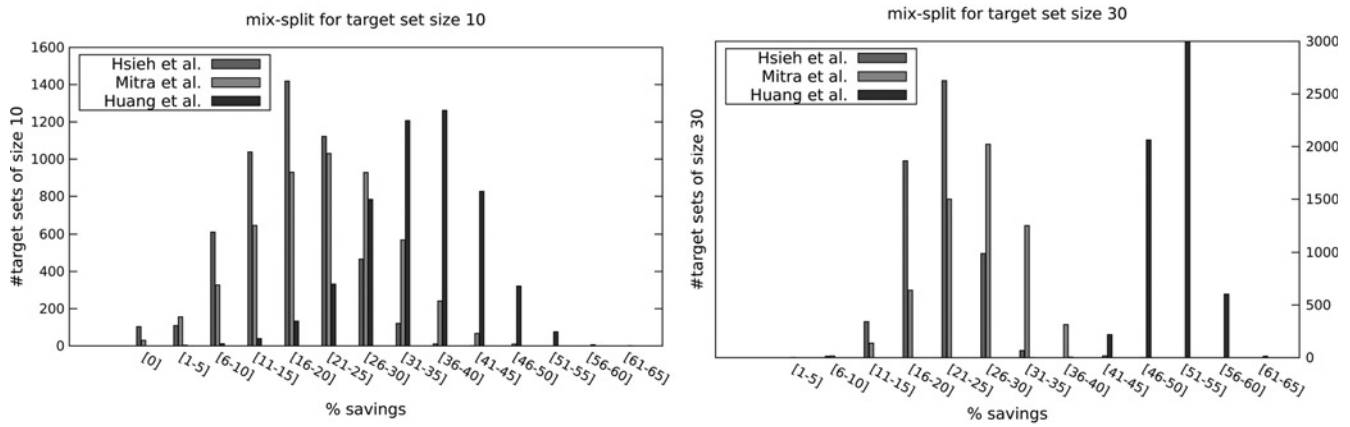

Fig. 10 *Histogram of %-savings in mix-split steps of the proposed method compared with Hsieh et al. [14], Mitra et al. [12] and Huang et al. [15]*

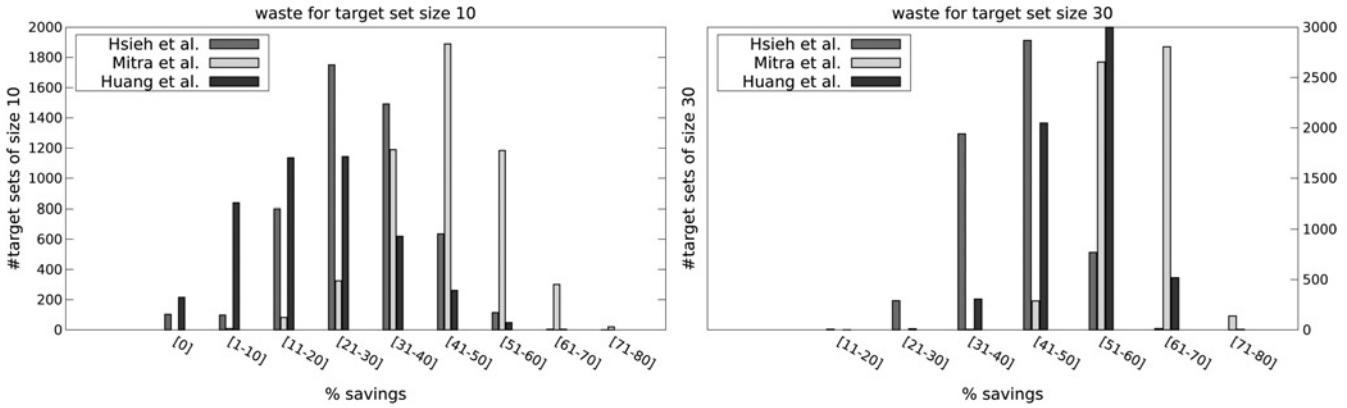

Fig. 11 *Histogram of %-savings in waste droplets of the proposed method compared with Hsieh et al. [14], Mitra et al. [12] and Huang et al. [15]*

method has positive %-savings compared with other methods. Similarly, Fig. 11 shows the %-savings of waste droplets with respect to three earlier methods.

Our experiments reveal that with increasing cardinality of the target set, the proposed scheme scales up favourably. This happens because of the fact that the effect of pruning becomes more dominant when the target set size increases. As a result, improved solutions are obtained that yield fewer mix-split steps and reduced wastage. We considered target sets of size 10, 30, 50, 100 and 200 in our experiments. Fig. 12 shows the average behaviour of the proposed method against three earlier methods. The results demonstrate significant savings in terms of both mix-split steps and waste droplets. We have also compared sample and buffer consumption of our method with those obtained by a reactant-minimising algorithm REMIA [15]. Fig. 13 shows that the proposed method requires fewer sample droplets for a target set of size 20 or more, as well as fewer buffer droplets. The proposed method not only reduces sample preparation time but it also minimises consumption of valuable reactants (sample, buffer).

The method proposed by Hsieh *et al.* [14] uses a technique based on identification of common sub-trees in a dilution tree for generating multiple sets. Recently, a reagent saving dilution algorithm REMIA has been proposed by Huang *et al.* [15] that uses n exponential gradients for generating any target set. In that method, for multiple target generation, one may have to store all the exponential gradients. However, in both of these methods, the possibility of mixing two intermediate droplets, that is, pruning was not explored. The performance of our algorithm is found to be significantly improved when pruning is applied.

5 Architecture

In the proposed method, a target droplet can be generated in two ways: either by combining the current droplet with a supply/buffer droplet or by mixing two previously generated droplets, that is, by applying the aforesaid pruning procedure. Thus, the number of pruning has a direct consequence on the storage requirement in the realisation of this proposed scheme. Before pruning, a droplet generated earlier is to be stored until it is mixed with a desired droplet. The extra storage required for pruning is bounded above by the number of maximum prunings that can occur in the dilution tree. In order to estimate the additional storage cells experimentally, we have performed extensive simulation on various examples. Since some applications may require a large number of target concentrations (up to one hundred) [16], we have simulated 5000 synthetic random target sets of size 10, 30, 50 and 100. From Fig. 14 it can be seen that the maximum pruning count is 23; thus, for implementing our test data at most 23 extra storage cells are required. However, for a target set size up to 50, only 15 extra storage cells will be sufficient. The pruning algorithm can also be adapted based on the available storage cells. In this case, the number of pruning should be restricted in order to satisfy the limited

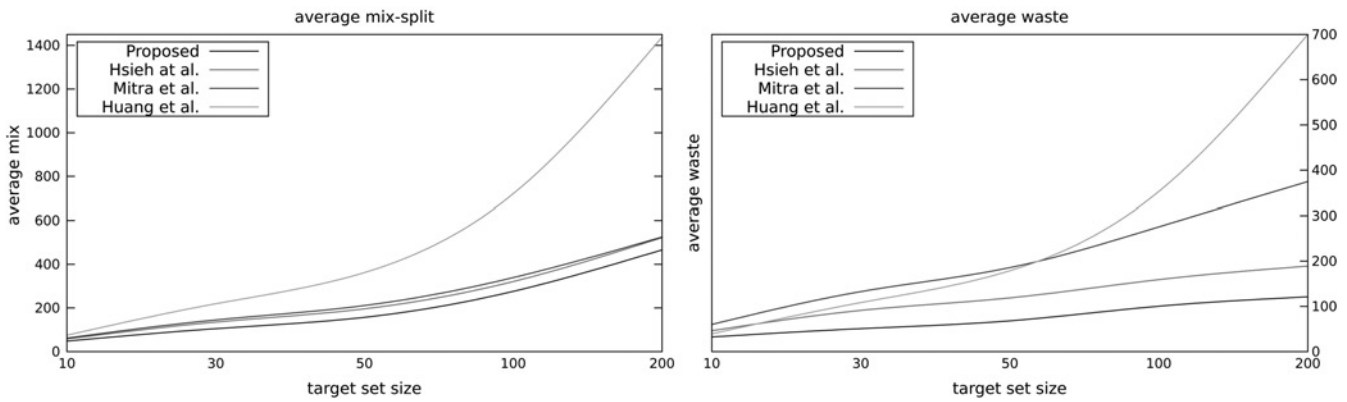

Fig. 12 *Average number of mix-split and waste for the proposed method and for those by Hsieh et al. [14], Mitra et al. [12] and Huang et al. [15]*

doi: 10.1049/iet-cdt.2013.0053

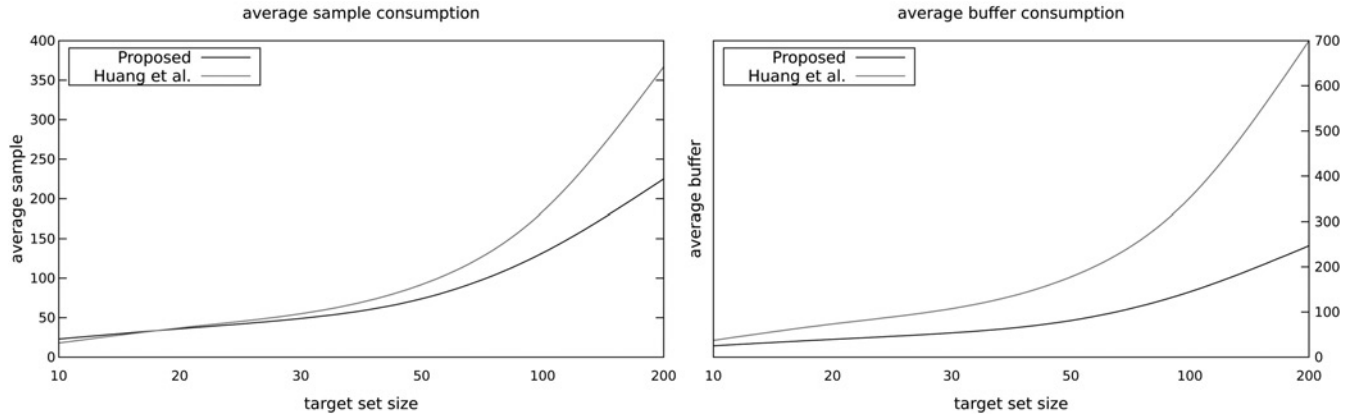

Fig. 13 *Consumption of average sample and buffer consumption for the proposed method with Huang et al. [15]*

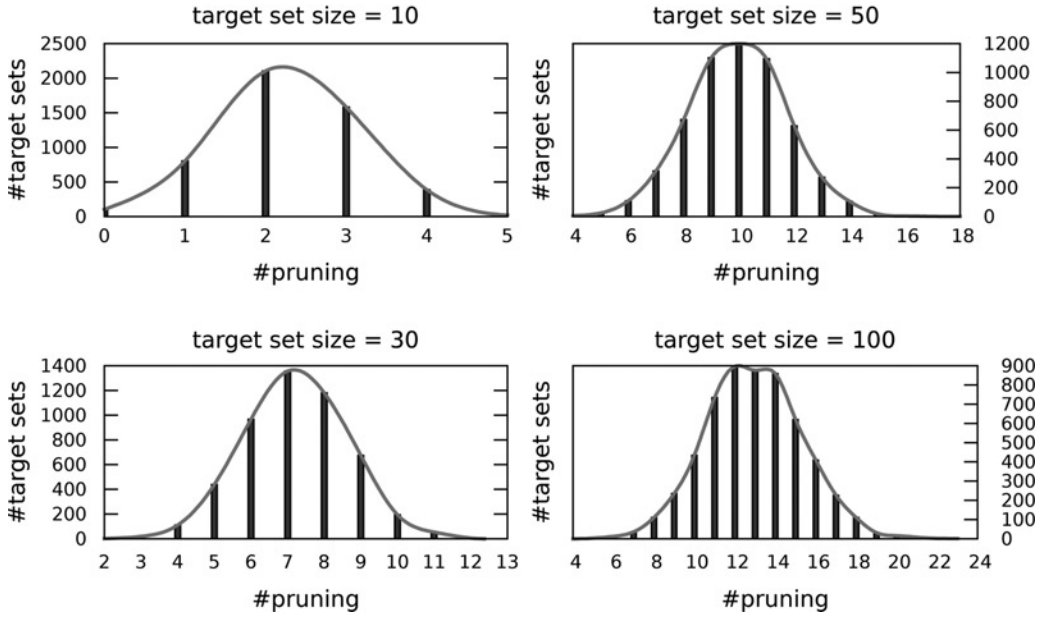

Fig. 14 *Histogram of pruning for random target sets of different size*

availability of storage cells. A separate stack of n storage cells is required for generating the targets, which are present on the final dilution tree. Here, a stack is used so that the dilution tree can be processed in depth-first order [23]. An architectural layout of a DMF biochip that supports the execution of our algorithm is shown in Fig. 15.

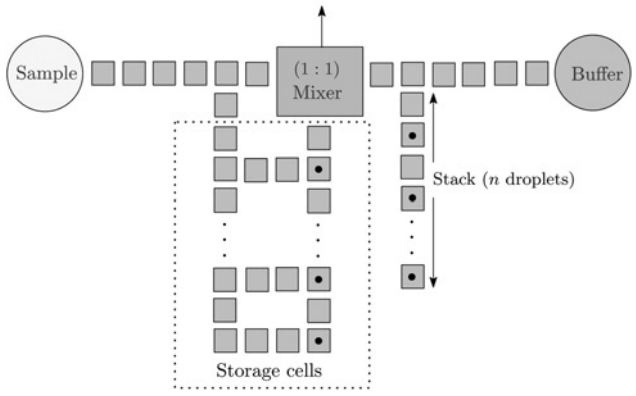

Fig. 15 *Architectural layout of a DMF biochip*

6 Conclusions

We have proposed an efficient method for generating a set of target droplets with various concentration factors. We have introduced the notion of a dilution tree that represents the entire range of target concentration factors. Our heuristic constructs only the relevant portion of the dilution tree and expedites the search, based on the proposed pruning technique. A search technique allows us to select in a cost effective fashion, a suitable subset of intermediate droplets, which can be best reused in the process. Simulation results reflect the improvement of the proposed method in comparison with current approaches. The efficiency of our method increases in terms of mix-split (sample preparation time) and waste (sample cost) with the increase of target set size. This happens because the effect of pruning becomes more dominant with the increasing target set size. The scheme also saves valuable reagents (sample and buffer) significantly. A simple architectural layout with one mixer module and a stack-organised storage electrodes is also designed. The proposed method has inherent parallelism, which is apparent from the dilution graph. Thus sample preparation time can further be reduced by deploying

multiple mixers. In this case, an appropriate scheduling of mixers will be required to minimise mixing time. Such a combined optimisation issue can be studied as a future research problem.

7 Acknowledgment

A preliminary version of this paper appeared in the Proceedings of ISED 2012 [1]. The authors wish to thank Dr. Rishad Shafik, Dr. Maurizio Palesi, Dr. Satrajit Ghosh and anonymous reviewers for their critical comments and valuable reviews. The work of B. B. Bhattacharya was supported by a special grant to Nanotechnology Research Triangle from Indian Statistical Institute, Kolkata, India.

8 References

1 Bhattacharjee, S., Banerjee, A., Bhattacharya, B.B.: 'Multiple dilution sample preparation using digital microfluidic biochips,'. Proc., Int. Symp. Electronic System Design, 2012, pp. 188–192
2 Fair, R.B., Khlystov, A., Tailor, T.D., *et al*.: 'Chemical and biological applications of digital-microfluidic devices', *IEEE Design Test Comput.* , 2007, **24**, (1), pp. 10–24
3 Pollack, M.G., Fair, R.B., Shenderov, A.D.: 'Electrowetting-based actuation of liquid droplets for microfluidic applications', *Appl. Phys. Lett.*, 2000, **77**, (11), pp. 1725–1726
4 Cho, S.K., Moon, H., Kim, C.-J.: 'Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits', *J. Microelectromech. Syst.*, 2003, **12**, (1), pp. 70–80
5 Fouillet, Y., Jary, D., Chabrol, C., Claustre, P., Peponnet, C.: 'Digital microfluidic design and optimization of classic and new fluidic functions for lab on a chip systems', *Microfluidics Nanofluidics*, 2008, **4**, pp. 159–165
6 Lee, K., Kim, C., Ahn, B., *et al*.: 'Generalized serial dilution module for monotonic and arbitrary microfluidic gradient generators', *Lab Chip*, 2009, **9**, pp. 709–717
7 Walker, G.M., Monteiro-Riviere, N., Rouse, J., Neill, O.T.: 'A linear dilution microfluidic device for cytotoxicity assays', *Lab Chip*, 2010, **7**, pp. 226–232
8 O'Neill, A., Monteiro-Riviere, N., Walker, G.: 'A serial dilution microfluidic device for cytotoxicity assays'. Engineering in Medicine and Biology Society, 2006, pp. 2836–2839
9 Roy, S., Bhattacharya, B.B., Chakrabarti, P.P., Chakrabarty, K.: 'Layout-aware solution preparation for biochemical analysis on a digital microfluidic biochip'. Proc., VLSI Design, 2011, pp. 171–176
10 Roy, S., Bhattacharya, B.B., Chakrabarty, K.: 'Waste-aware dilution and mixing of biochemical samples with digital microfluidic biochips'. Proc., DATE, 2011, pp. 1059–1064
11 Xu, T., Pamula, V., Chakrabarty, K.: 'Automated, accurate, and inexpensive solution-preparation on a digital microfluidic biochip'. Proc. Biomedical Circuits and Systems Conf., BioCAS, 2008, pp. 301–304
12 Mitra, D., Roy, S., Chakrabarty, K., Bhattacharya, B.B.: 'On-chip sample preparation with multiple dilutions using digital microfluidics'. Proc. ISVLSI, 2012, pp. 314–319
13 Roy, S., Bhattacharya, B.B., Ghoshal, S., Chakrabarty, K.: 'Low-cost dilution engine for sample preparation in digital microfluidic biochips'. Proc. Int. Symp. Electronic System Design, 2012, pp. 203–207
14 Hsieh, Y.-L., Ho, T.-Y., Chakrabarty, K.: 'A reagent-saving mixing algorithm for preparing multiple-target biochemical samples using digital microfluidics', *IEEE Trans. CAD Integr. Circuits Syst.*, 2012, **31**, (11), pp. 1656–1669
15 Huang, J.-D., Liu, C.-H., Chiang, T.-W.: 'Reactant minimization during sample preparation on digital microfluidic biochips using skewed mixing trees'. Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD), Nov. 2012, pp. 377–383
16 Xu, T., Chakrabarty, K., Pamula, V.K.: 'Defect-tolerant design and optimization of a digital microfluidic biochip for protein crystallization', *IEEE Trans. CAD Integr. Circuits Syst.*, 2010, **29**, (4), pp. 552–565
17 Srinivasan, V., Pamula, V.K., Fair, R.B.: 'Droplet-based microfluidic lab-on-a-chip for glucose detection', *Anal. Chim. Acta.*, 2004, **507**, (1), pp. 145–150
18 Chakrabarty, K., Su, F.: 'Digital microfluidic biochips - synthesis, testing, and reconfiguration techniques' (CRC Press, 2007)
19 Thies, W., Urbanski, J.P., Thorsen, T., Amarasinghe, S.P.: 'Abstraction layers for scalable microfluidic biocomputing', *Nat. Comput.*, 2008, **7**, (2), pp. 255–275
20 Roy, S., Bhattacharya, B.B., Chakrabarty, K.: 'Optimization of dilution and mixing of biochemical samples using digital microfluidic biochips', *IEEE Trans. CAD Integr. Circuits Syst.*, 2010, **29**, (11), pp. 1696–1708
21 Chiang, T.-W., Liu, C.-H., Huang, J.-D.: 'Graph-based optimal reactant minimization for sample preparation on digital microfluidic biochips'. VLSI Design and Test, 2013
22 Aho, A.V., Hopcroft, J.E., Ullman, J.D.: 'The design and analysis of computer algorithms' (Addison-Wesley, 1974)
23 Tarjan, R.E.: 'Depth-first search and linear graph algorithms', *SIAM J. Comput.*, 1972, **1**, (2), pp. 146–160