Theory and Analysis of Generalized Mixing and Dilution of Biochemical Fluids Using Digital Microfluidic Biochips

SUDIP ROY, Indian Institute of Technology
BHARGAB B. BHATTACHARYA, Indian Statistical Institute
SARMISHTHA GHOSHAL, Indian Institute of Engineering Science and Technology
KRISHNENDU CHAKRABARTY, Duke University

Digital microfluidic (DMF) biochips are recently being advocated for fast on-chip implementation of biochemical laboratory assays or protocols, and several algorithms for diluting and mixing of reagents have been reported. However, all methods for such automatic sample preparation suffer from a drawback that they assume the availability of input fluids in pure form, that is, each with an extreme concentration factor (*CF*) of 100%. In many real-life scenarios, the stock solutions consist of samples/reagents with multiple *CF*s. No algorithm is yet known for preparing a target mixture of fluids with a given ratio when its constituents are supplied with random concentrations. An intriguing question is whether or not a given target ratio is feasible to produce from such a general input condition. In this article, we first study the feasibility properties for the generalized mixing problem under the (1 : 1) mix-split model with an allowable error in the target *CF*s not exceeding $\frac{1}{2^d}$, where the integer d is user specified and denotes the desired accuracy level of *CF*. Next, an algorithm is proposed which produces the desired target ratio of N reagents in $\mathcal{O}(Nd)$ mix-split steps, where N (≥ 3) denotes the number of constituent fluids in the mixture. The feasibility analysis also leads to the characterization of the total space of input stock solutions from which a given target mixture can be derived, and conversely, the space of all target ratios, which are derivable from a given set of input reagents with arbitrary *CF*s. Finally, we present a generalized algorithm for diluting a sample S in minimum (1 : 1) mix-split steps when two or more arbitrary concentrations of S (diluted with the same buffer) are supplied as inputs. These results settle several open questions in droplet-based algorithmic microfluidics and offer efficient solutions for a wider class of on-chip sample preparation problems.

Categories and Subject Descriptors: B.7.1 [**Integrated Circuits**]: Types and Design Styles—*Algorithms implemented in hardware*; B.7.2 [**Integrated Circuits**]: Design Aids—*Layout*; B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance; J.3 [**Life and Medical Sciences**]: Biology and Genetics, Health

General Terms: Algorithms, Design

Additional Key Words and Phrases: Biochips, design automation, digital microfluidics, dilution and mixing, sample preparation

---

A preliminary version of a part of this article appeared in Roy et al. [2013b].
The work of S. Roy was supported by the Microsoft Research India PhD Fellowship Award, 2010–2014. The work of B. B. Bhattacharya was supported by a special grant to Nanotechnology Research Triangle from the Indian Statistical Institute, Kolkata. The work of S. Ghoshal was supported in part by the Dept. of Sc. & Tech. (DST) Women Scientists Scheme (WOSA), 2009–2012. The work of K. Chakrabarty was supported in part by the US National Science Foundation under grants CCF-0914895 and CNS-1135853.
Authors' addresses: S. Roy, Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur-721302, India; email: sudiproy.bkp@gmail.com; B. B. Bhattacharya, Advanced Computing and Microelectronics Unit, Indian Statistical Institute, 203, B.T. Road, Kolkata-700108, India; email: bhargab@isical.ac.in; S. Ghoshal, School of Materials Science and Engineering, Indian Institute of Engineering Science and Technology, Howrah-711103, India; email: sharmi.bhatta@gmail.com; K. Chakrabarty, Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708; email: krish@ee.duke.edu.

DOI: http://dx.doi.org/10.1145/2629578

ACM Reference Format:
Roy, S., Bhattacharya, B. B., Ghoshal S., and Chakrabarty, K. 2014. Theory and analysis of generalized mixing and dilution of biochemical fluids using digital microfluidic biochips. *ACM J. Emerg. Technol. Comput. Syst.* 11, 1, Article 2 (September 2014), 33 pages.
DOI: http://dx.doi.org/10.1145/2629578

1. INTRODUCTION

Recent escalation in healthcare cost for cardiovascular diseases, cancer, diabetes, and global HIV/AIDS crisis has fueled a new field of interdisciplinary research centered around "lab-on-a-chip (LoC)" [Sista et al. 2008; Tan et al. 2008]. A typical LoC (or a biochip) is a tiny microfluidic device of few square centimeters in area, on which several biochemical laboratory protocols or assays can be conveniently implemented. In general, two kinds of technologies are used in practice: (i) continuous-flow microfluidic (CMF) biochips, where fluid flow is actuated through pre-etched microchannels on-chip using micropumps and microvalves, and (ii) digital microfluidic (DMF) biochips, which use electrical actuation to manipulate (dispensing, navigation, mixing, splitting, washing, sensing) discrete droplets of nano- or pico-liter volume of biochemical fluids on a two-dimensional electrode array [Chakrabarty and Xu 2010]. Recently, DMF biochips have gained wide acceptance in developing LoC applications because of their flexibility and programmability [Abdelgawad and Wheeelr 2009; Chatterjee et al. 2006; Miller and Wheeler 2009].

A schematic diagram of the top view of a DMF biochip and the cross-sectional view of a basic cell are shown in Figure 1(a) and (b), respectively. A unit cell in the array includes a pair of electrodes that act as two parallel plates, of which the bottom plate contains a patterned array of individually controlled electrodes and the top plate is coated with a continuous ground electrode. A droplet is sandwiched between the two plates and rests on a hydrophobic surface over an electrode. It can be moved by applying a control voltage (above a threshold voltage) to an adjacent electrode and, at the same time, deactivating the electrode just under the droplet. By varying the patterns of control voltage activation, all the fundamental fluidic operations such as merging, splitting, mixing, and dispensing of droplets can be executed [Chakrabarty and Xu 2010; Cho et al. 2003; Fair 2007; Fouillet et al. 2008].

An ideal on-chip biochemical analysis system should provide high-throughput operation with low consumption of expensive reactant fluids and with negligible likelihood of error as a benefit of minimal human intervention [Abdelgawad and Wheeelr 2009; Chakrabarty and Xu 2010; Fair 2007]. It should also be integrated with sensors, fully-automated, inexpensive, and reliable. The emerging application areas of biochips include among others, clinical diagnostics, especially the immediate point-of-care diagnosis of diseases, enzymatic analysis (e.g., glucose and lactate assays), DNA analysis (e.g., PCR and nucleic acid sequence analysis), proteomic analysis involving proteins and peptides [Luk and Wheeler 2009], immunoassay, and environmental toxicity monitoring [Fair et al. 2007; Ren et al. 2003]. The worldwide market for in-vitro diagnostics (IVD) in 2007 was estimated at $38 billion,[1] and execution of 1.5 billion diagnostic tests per year worldwide has been predicted for malaria alone.[2] In tandem with the ITRS forecast[3] 2007, which highlights "Medical" as being a "System Driver" for the future, recent years have seen a surge of interest in various design automation methods

[1]Global In Vitro Diagnostic Market Analysis; http://www.prlog.org/10080477-global-in-vitro-diagnostic-market-analysis.html.
[2]World Malaria Day 2009: Key Figures; http://www.rollbackmalaria.org/worldmalariaday/keyfigures.
[3]International Technology Roadmap for Semiconductors; http://www.itrs.net/.

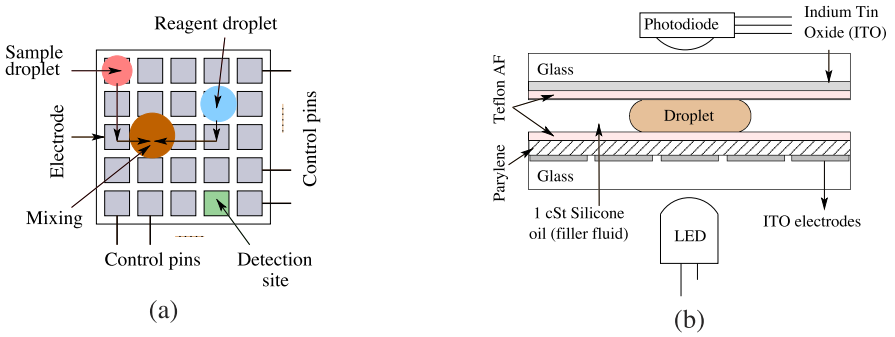


Fig. 1. (a) Top view of a digital microfluidic biochip and (b) Cross-sectional view of a cell at a detection site [Chakrabarty and Xu 2010; Fair et al. 2007].

for microfluidic biochips [Huang and Ho 2011; Huang et al. 2010; Lin and Chang 2011; Luo et al. 2012, 2013; Maftei et al. 2012].

The preparation of a diluted sample or a mixture of several reactant fluids in a certain ratio is an essential pre-processing step required in many biochemical protocols, and hence, its implementation on a DMF chip will greatly reduce the completion time of an assay [Luk and Wheeler 2009]. For example, these procedures are needed in real-time PCR for cDNA, immunoassays for detecting cytokines in serum samples; in the enzymatic glucose assay (Trinder's reaction), where a dilution factor (*DF*) of 200 or more is used [Ren et al. 2003]. Another example is the protocol for extracting proteins from heterogeneous fluids by precipitation [Jebrail and Wheeler 2009] that requires several reagents with different concentration levels, such as 50 mg/mL of BSA solution (sample), 20% TCA (precipitant), 70/30 v/v chloroform/acetonitrile (rinse solution) and 100 mM borate buffer containing 1% SDS (resolubilizing buffer). Similarly, the dilution step is required in almost all biochemical analysis. Dilution gradients, that is, diluting a sample over a certain range of concentration factors (*CF*s), play an important role in drug design. It is also needed in the quantification of test results using sensors, because the *CF* of the droplet under detection may be beyond the on-chip sensor range.

Off-chip sample preparation is slow and thus poses a significant obstacle to the automation of biochemical assays. For high-throughput applications, recently several dilution/mixing techniques and devices have been reported for automatic solution preparation with CMF-based LoC's [Lee et al. 2009; Yusuf et al. 2009]. Similarly, a number of algorithms have appeared in the literature for diluting fluids on a DMF platform [Ren et al. 2003], *GAG* [Griffith et al. 2006], *twoWayMix* [Thies et al. 2008], *DMRW* [Roy et al. 2010], *IDMA* [Roy et al. 2011a], *MD* [Bhattacharjee et al. 2012], *REMIA* [Huang et al. 2012], *GORMA* [Chiang et al. 2013], *WARA* [Huang et al. 2013], *NFSP* [Dinh et al. 2014], and for mixing a number of reagents such as *MinMix* [Thies et al. 2008], *RMA* [Roy et al. 2011b], *RSM* [Hsieh et al. 2012], *MTCS* [Kumar et al. 2013], *CoDOS* [Liu et al. 2013]. However, prior work on automatic sample preparation with digital microfluidics suffers from a key limitation. All these methods (except *GAG*, *DMRW* and *IDMA*) assume that the input fluids are supplied only in pure form, that is, each with the extreme value of $CF = 100\%$. In many practical situations, such pure samples and reagents may not be readily available in stock. Furthermore, all the cited methods for digital microfluidics use a sequence of droplet mix/split cycles while diluting a sample. These steps often produce several waste droplets, which carry samples/reagents with various *CF*s, thereby causing loss of expensive reactants [Thies et al. 2008]. Some of the dilution algorithms [Griffith et al. 2006; Roy et al. 2010;

2011a] can produce target concentration of a reactant fluid from the supply of impure concentration levels. Some other dilution/mixing algorithms [Dinh et al. 2014; Hsieh et al. 2012; Huang et al. 2012, 2013; Kumar et al. 2013; Liu et al. 2013] can recycle the unused intermediate droplets while consuming the supply of pure (100%) concentration levels of the reactant fluids. However, the preparation of a target mixture of three or more reagents from a supply of input fluids having impure concentration, has not been studied earlier. This calls for a new and more general approach to the mixing problem of digital microfluidics.

Main Results. The key contributions of this article are summarized as follows.

— We consider the generalized mixing problem: given a supply of N (≥ 3) fluids each with a random *CF*, prepare a target mixture with a given ratio of its constituents on a DMF platform. The objective is to minimize the number of $(1:1)$ mix-split steps such that the error in each of the constituent *CF*s of the target does not exceed $\frac{1}{2^d}$, where the integer d is user specified and denotes the desired accuracy level of *CF*. Our analysis shows that when the input fluids are supplied with arbitrary values of *CF*s, a given target may not always be feasible to produce under this scheme. We derive the necessary and sufficient conditions for *reachability* of a target in this generalized mixing scenario. A mixing algorithm is then proposed which produces the desired target ratio of N reagents in $\mathcal{O}(Nd)(1:1)$ mix-split steps, provided the reachability conditions are satisfied.
— Next, the proposed feasibility analysis is used to characterize the set of input stock solutions from which a given target mixture can be derived, and conversely, the set of all target ratios that can be derived from a given set of input reagents.
— Finally, we solve the generalized dilution problem with related inputs: given a supply of a sample S with N arbitrary concentration factors (each diluted with the same buffer), the objective is to produce a desired dilution of a sample S using the minimum number of $(1:1)$ mix-split steps. For $N = 2$, a simple scaling technique can be used to determine the optimum mix-split sequence that produces, in $\mathcal{O}(d)$ time, the target droplet with a maximum error of $\frac{1}{2^{d+1}}$ in its *CF*. For $N \geq 3$, we conjecture that the problem is *NP-hard* and propose a heuristic algorithm based on dynamic programming.

The rest of the article is organized as follows. Related previous work on dilution and mixing of biochemical fluids using DMF biochips is presented in Section 2. Section 3 deals with the formulation of the generalized mixing problem, describes the reachability conditions of a target mixture, their solution space, and related algorithms. The generalized dilution problem with related inputs is studied in Section 4. Finally, conclusions are drawn in Section 5.

2. AUTOMATIC DILUTION AND MIXING OF BIOCHEMICAL FLUIDS

2.1. Preliminaries

A DMF biochip operates with discrete droplets on a uniform 2D-array of equi-sized electrodes, hence the volume of a droplet is usually an integral multiple of that of a single droplet. Various $(k:\ell)$ mixing models may be considered, where a k-unit volume of one fluid is mixed with an ℓ-unit volume of another fluid to produce a $(k+\ell)$-unit volume of the resultant mixture in a single mixing operation. Three such mixing models are: (i) $k = \ell = 1$, (ii) $k = \ell \neq 1$, and (iii) $k \neq \ell$, where k, ℓ are positive integers. Note that the first one, that is, the $(1:1)$ mixing model is the easiest to implement.

While specifying a sample, the term *concentration factor* (*CF*) (or *dilution factor* (*DF*)) is used to quantify the amount of raw fluid in the prepared sample [Herold and

Rasooly 2009]. The *concentration factor* is defined as the ratio of initial volume of the sample to the final volume of the prepared mixture (i.e., the reciprocal of the dilution factor, $CF = \frac{1}{DF}$). The fluid with which the sample is mixed for dilution is called the diluent or buffer solution, for instance, water or other liquid, which is neutral to the sample. Hence, in the preparation of a mixture, a pure buffer solution can be viewed as having 0% concentration (i.e., $CF = 0$) with respect to a sample or reagent, whereas, it can be viewed as 100% with respect to itself. By convention, a pure sample or reagent fluid is viewed as having 100% concentration, that is, $CF = 1$. Dilution is commonly used in biological studies to create a variety of solution concentrations of a fluid with the help of a buffer solution [Herold and Rasooly 2009]. On the other hand, mixing is used to prepare a mixture (solution) of three or more different fluids with a desired ratio of their concentrations. Hence, dilution is a special case of mixing of two input fluids, one of which is a buffer solution. In general, dilution of a sample fluid with $CF = C_1$ can be achieved by mixing it with another sample of same fluid with $CF = C_2$, if $C_2 < C_1$. The CF of the resultant fluid lies between C_1 and C_2 because, if the samples with $CF = C_1$ and C_2 are mixed in a volumetric ratio of $k : \ell$, then the resulting $(k + \ell)$-unit volume fluid has a $CF = C_r = \frac{k.C_1 + \ell.C_2}{k+\ell}$. After a balanced splitting of the resultant volume, two $\left(\frac{k+\ell}{2}\right)$-unit volume resultant droplets are produced.

In this work, we assume the $(1 : 1)$ mixing model for dilution and mixing, that is, every *mix-split cycle* consists of a mix operation between two unit-volume fluid droplets and followed by a balanced split operation of the mixed fluid. Thus, if a fluid droplet with $CF = C_1$ is mixed with its another droplet of the same reagent, with $CF = C_2$ (where $0 \leq C_2 \leq C_1$), then the final CF of the each of the resulting two droplets becomes $\frac{C_1+C_2}{2}$, assuming that the same dilutent is used in preparing the two droplets. One mix operation and a subsequent split are together called as one *mix-split operation, cycle, or step*. Since after one $(1 : 1)$ mix-split operation, the resulting CF becomes the mean value of those of the two source droplets, for the sake of convenience, the desired target CF of a sample, (C_t), $0 \leq C_t < 1$, is expressed as a d-bit binary fraction (by truncating it off beyond d^{th}-bit), when an accuracy level of d is desired. In other words, the CF of a reagent in a target droplet can be expressed as $\frac{x}{2^d}$, where $x \in \mathbb{Z}^+$, $0 \leq x \leq (2^d - 1)$. Note that, the $CF = 0$ (1) can be represented as $\frac{0}{2^d}$ $\left(\frac{2^d}{2^d}\right)$.

In the process of dilution and mixing, a mix-split cycle is deployed repeatedly in finitely many times with different fluid droplets (input sample, buffer, or other intermediate droplets produced in earlier cycles) to achieve the desired dilution (mixture) of the input fluid(s). These set of mix-split cycles and their sequential dependence are represented by a task graph known as *dilution* or *mixing tree* [Thies et al. 2008]. In a $(1 : 1)$ dilution or mixing tree, each leaf node corresponds to a unit-volume droplet of an input fluid and an internal (or non-leaf) node denotes the resultant mixture obtained by applying a $(1 : 1)$ mix-split step on two unit-volume fluid droplets corresponding to its two children. An internal node denotes one unit-volume droplet of the resultant mixture, which is used in the next mix operation denoted by its parent node. The remaining droplet is either discarded as waste or stored for reuse in a subsequent mix operation, depending upon the underlying mixing algorithm [Griffith et al. 2006; Hsieh et al. 2012; Huang et al. 2012, 2013; Kumar et al. 2013; Liu et al. 2013; Roy et al. 2010, 2011a; Thies et al. 2008]. Such a discarded droplet is referred to as a *waste droplet*. The mix-split operation denoted by an internal node can be executed only when each of its two children is either already processed or is a leaf node. To represent this task dependence, every edge of the tree between a node (except the root) and its parent is directed towards the latter. A post-order traversal of the mixing tree provides the sequence of $(1 : 1)$ mix-split steps required to produce the target droplet denoted by the root of the

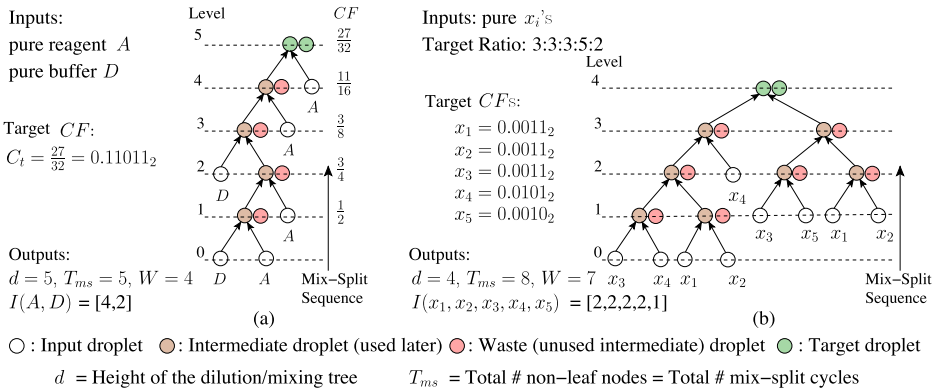


Fig. 2. (a) A Dilution tree denoting sequence of five mix-split cycles for target *CF*, $C_t = 84.375\%$ ($\approx \frac{27}{32} \equiv 0.11011_2$) obtained by *twoWayMix* [Thies et al. 2008] using pure *A* and *D*. (b) For an example target ratio $3:3:3:5:2$, the bit-representations and the mixing tree obtained by *MinMix* [Thies et al. 2008] using 100% conc. of all five input fluids.

Table I. List of Notations Used in this Article

Symbol	Meaning
N	Number of input fluids
d	Accuracy level of desired *CF*
T_{ms}	Total number of $(1:1)$ mix-split cycles needed in dilution/mixing
W	Total number of waste droplets generated
I	An array of N integers, where each entry denotes the number of the corresponding input droplets required in the dilution/mixing process

tree. The total number of non-leaf nodes in a mixing tree thus indicates the number of $(1:1)$ mix-split cycles required to produce the target dilution or mixture from the inputs, and the maximum height of the tree is determined by the desired accuracy level d of the target *CF*. The key concepts and notation presented here are explained with two examples in Figure 2.

In all subsequent discussions, we will use the notation given in Table I.

2.2. Prior Work: Dilution and Mixing Algorithms

Griffith et al. [2006] proposed an algorithm, referred as *GAG*, of $\mathcal{O}(d^3)$ time complexity, to determine a d-length sequence of mix-split cycles for automated dilution of sample fluid on a DMF biochip, such that the error in desired *CF* does not exceed $\frac{1}{2^{d+1}}$. A similar algorithm, namely *DMRW*, was proposed in [Roy et al. 2010] (also a variant of it, *IDMA*, reported in Roy et al. [2011a]) that utilizes the $(k:k)$ mixing model with the help of a rotary mixer. All these methods require on-chip storage of several intermediate droplets, which are reused while obtaining the desired target dilution of the sample. These algorithms attempt to minimize W, and it is observed that $T_{ms} \geq d$. Although these methods can perform dilution of a sample starting from two arbitrary boundary concentrations, they fail to optimize the number of mix-split steps, waste, or reactant usage, when more than two *CF* values of the same sample (related inputs) are available in the stock solution. Furthermore, they cannot produce a target mixture of more than two reagents.

Another dilution algorithm, called *twoWayMix*, of $\mathcal{O}(d)$ time complexity, was reported in Thies et al. [2008] for mixing of two sample fluids at any desired ratio using the $(1:1)$ mixing model. This dilution method has two advantages: (i) no storage of

droplets with intermediate *CF*s is required; the current droplet is mixed only with an input fluid (sample or buffer) in the next step, and (ii) the number of mix-split cycles is minimum. However, it produces a waste droplet at every mix-split cycle (excepting the last one, when the two target droplets are produced). Thus, in this dilution algorithm, $T_{ms} = d$. Also, this method suffers from the drawback that each input has to be supplied with the *CF* of 100% only. For example, Figure 2(a) shows the dilution tree of height 5 obtained by *twoWayMix*, which depicts the production of a target with *CF* = 84.375%, given a sample A and a buffer solution D as input. Since the accuracy level is assumed to be 5, the target *CF* is expressed as a 5-bit fractional binary number 0.11011_2. Hence, for this example, $W = 4$ and $I(A, D) = [4, 2]$.

Note that dilution is an essential step in drug design where a concentration gradient of a drug is often needed. For example, it is important to determine the minimum amount of an antibiotic that inhibits the visible growth of bacteria isolate (defined as Minimum Inhibitory Concentration (MIC)). The drug with the least *CF* (i.e., with highest dilution) that is capable of arresting the growth of bacteria, is considered to have the MIC [Cira et al. 2012]. In order to conduct such tests, drugs with several dilution gradient patterns are needed [Sugiura et al. 2010].

Recently, several other algorithms for diluting a sample to generate serial or linear gradients, or to produce random multiple *CF*s, with reduced wastage and time, have been reported [Bhattacharjee et al. 2013, 2014; Chiang et al. 2013; Huang et al. 2013; Mitra et al. 2012; Roy et al. 2013a]. However, all these methods need the input fluid with a *CF* of 100%.

Huang et al. recently proposed two other dilution algorithms for reactant minimization called *REMIA* [Huang et al. 2012] and *WARA* [Huang et al. 2013]. These algorithms can recycle the unused intermediate droplets while consuming the supply of pure concentration levels of the reactant fluids. Another improved dilution technique, called graph-based optimal reactant minimization algorithm (*GORMA*), has been reported by Chiang et al. [2013]. More recently, a dilution method, called *NFSP*, has been reported by Dinh et al. [2014] for generating multiple target concentrations of a sample fluid from the supply of pure input fluids with reactant and waste minimization. However, these algorithms did not consider the case where the input fluids are supplied with random concentration factors.

Thies et al. [2008] also proposed a mixing algorithm, called *MinMix*, to determine the mixing tree for a desired concentration ratio of N different fluids. First, the *CF* values of each of the constituents in the target ratio is expressed as a d-bit binary fraction. Next, these N d-bit binary representations are bit-wise scanned from right-to-left to construct, in a bottom-up fashion, a mixing tree of height d (here $T_{ms} \geq d$).

An example mixing tree for the target ratio {3:3:3:5:2} of five input fluids is shown in Figure 2(b), and the components of the target ratio (in binary), which are used to construct the tree are shown on the left. As evident from the figure, executing such a mixing tree on-chip may require storage units depending on the number of mixer modules available and the scheduling scheme of the tree.

Another mixing algorithm, namely *RMA* [Roy et al. 2011b] determines the mixing tree for a target ratio of input fluids using a top-down factorization based technique. An improved mixing algorithm, referred to as *RSM* [Hsieh et al. 2012], also uses a similar technique. Kumar et al. [2013] proposed another mixing algorithm, called *MTCS*, which can reduce the total number of mix-split steps and reactant usage by reusing intermediate waste droplets. A routing-aware resource-allocation scheme for on-chip mixture preparation, known as *RAMP*, was reported in Roy et al. [2013c] that can provide the suitable placement of resources (mixers, fluid reservoirs) on a chip layout while implementing a mixing algorithm. However, all the existing methods of mixture preparation [Hsieh et al. 2012; Kumar et al. 2013; Roy et al. 2011b; Thies et al. 2008]

assume that the reagents are available with only 100% *CF*, and they may fail to produce a target mixture, if the input fluids are supplied with arbitrary concentrations.

Hsieh et al. [2012] proposed an algorithm for reagent saving in sample preparation and Kumar et al. [2013] proposed a mixing algorithm called *MTCS*, which can recycle the unused intermediate droplets while consuming the supply of pure concentration levels of the reactant fluids. Recently, Liu et al. [2013] described a mixing algorithm called *CoDOS* to generate the mixing trees with common dilution operation sharing in order to recycle the unused intermediate droplets. However, these methods do not address the problem of mixing and dilution when the input fluids are available with arbitrary concentrations.

3. MIXING FROM ARBITRARY CONCENTRATIONS OF TWO OR MORE FLUIDS

In this section, we formally state the generalized mixing problem of droplet-based microfluidics and derive some theoretical results regarding the reachability of a given target mixture from a stock solution. We also provide an algorithm for generating the mix-split steps whenever the target is reachable.

We formulate the generalized mixing problem as follows.

Inputs

(a) A stock of N distinct fluids $A_1, A_2, \ldots, A_N$ ($N \geq 1$) with *CF*s $b_1, b_2, \ldots, b_N$ respectively, each diluted with the same buffer solution D, where $0 < b_i \leq 1$, $1 \leq i \leq N$.
(b) A stock of buffer solution D. Thus, the total number of input fluids including D is $(N+1)$; $(N+1) \geq 2$.
(c) A target mixture M of $\{A_1, A_2, \ldots, A_N, D\}$ with a ratio of *CF*s $\{c_1 : c_2 : \ldots : c_N : c_{N+1}\}$, where $0 \leq c_i < 1$, $1 \leq i \leq N+1$, such that $\sum_{i=1}^{N+1} c_i = 1$. Here c_{N+1} denotes the *CF* of the buffer solution D in the target mixture.
(d) An integer d to represent the accuracy level of the target *CF*s c_i in the target mixture M.

In other words, the maximum allowable error in each of the constituent *CF*s of M is bounded above by $\frac{1}{2^d}$. Also, we say that M is *reachable* from the inputs if it possible to produce M within the aforementioned error bound.

Output. A mixing tree $\mathcal{T}$ of height at most d, representing the task graph of the $(1:1)$ mix-split steps, execution of which will produce the target mixture droplet, whenever the target is reachable from the given inputs.

As mentioned earlier, the existing mixture preparation algorithms [Thies et al. 2008; Roy et al. 2011b; Hsieh et al. 2012] cannot be applied directly to produce the target droplet(s) under this general condition. To solve this problem, we develop the relevant theory of reachability of a target mixture as described as follows.

3.1. Target Mixture Reachability Conditions

We first present an example that motivates this study. Assume that we have a stock of three input fluids A, B and C with *CF* 80%, 70% and 45% respectively, each diluted with the same buffer solution D. We need to prepare a target mixture M_1 of A, B, C and D in the respective proportion of 20%, 30%, 41% and 9%, and another mixture M_2 in the proportion 20%, 30%, 9% and 41%. Note that, in both the cases, *CF* value of each of the constituents is smaller than that in the stock. However, it can be shown that

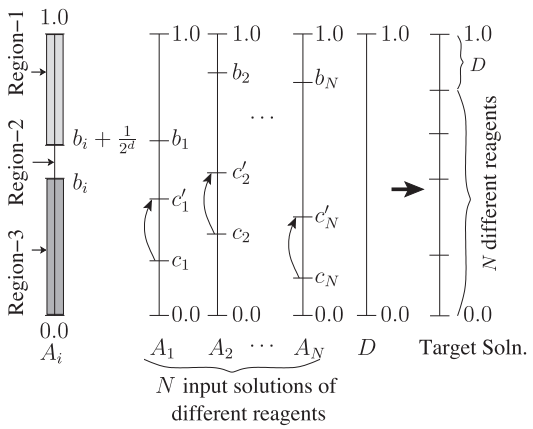


Fig. 3. Mapping of target *CF*s in the input scales (three regions in the scale for a fluid i are shown).

the target mixture M_1 cannot be prepared directly using this stock on a DMF biochip, whereas, M_2 can be prepared.

In generalized mixing, four cases may arise. The target mixture consists of

(i) at least one fluid (or there is only one fluid) A_i such that $c_i > \left(b_i + \frac{1}{2^d}\right)$ (Region-1);
(ii) only one fluid A_i such that $b_i \leq c_i \leq \left(b_i + \frac{1}{2^d}\right)$ (Region-2);
(iii) at least two fluids A_i and A_j of *CF*s c_i and c_j, respectively, such that $b_i \leq c_i \leq \left(b_i + \frac{1}{2^d}\right)$ (Region-2), and $c_j \geq \frac{1}{2^d}$;
(iv) all the fluids such that $0 \leq c_i < b_i$ holds for all i (Region-3).

These regions in the scale of *CF*s are shown in Figure 3.

For Case (i), the target mixture is *not reachable* within the error bound of desired *CF*s, because the *CF* of one of the constituent fluids in the target mixture exceeds that of the supplied input at least by the error margin. On the other hand, for Case (ii), the target mixture is *reachable* within the error bound by default, that is, the input fluid with *CF* b_i can be regarded as the output droplet(s). However, for Case (iii), the mixture is *not reachable*. It is apparent from the fact that if we have to prepare a mixture of at least two fluids, the *CF* of one fluid in the target mixture cannot be made equal to b_i without making the *CF* of the other fluid zero. For Case (iv), the target mixture is reachable depending on whether the following *reachability conditions* hold.

For subsequent discussion on mixing, we assume that the input *CF* values are treated in exactness, whereas, each of the output *CF* values in the target mixture is approximated as a d-bit binary fraction (i.e., by truncating-off other bits beyond the d^{th} bit in its binary representation). This will guarantee that its error is no more than $\frac{1}{2^d}$. Such discretization of target *CF*s is used in conformity with the inherent constraints imposed by the (1 : 1) mixing model and the use of finitely many mix-split steps in digital microfluidics.

In order to handle impure input fluids, we introduce the basic concept of scaling that can be used transform a target concentration in a proportionate scale. This is analogous to converting a temperature from Fahrenheit to Celsius scale. Hence, if c_i is the target concentration factor in the interval scale between 0.0 and 1.0, then the

converted/scaled concentration factor in the interval scale between 0.0 and b_i ($b_i \leq 1.0$) can be written as follows:

$$\frac{c'_i - 0.0}{1.0 - 0.0} = \frac{c_i - 0.0}{b_i - 0.0} \tag{1}$$

$$\text{Therefore, } c'_i = \frac{c_i}{b_i}. \tag{2}$$

In order to prepare a target mixture for Case (iv), that is, $\forall i$, $0 \leq c_i < b_i$, where $i = 1$ to N, the *necessary* and *sufficient condition of reachability* can be formally stated as the following theorem. Let $c'_i = \frac{c_i}{b_i}$, and let $\mathcal{B}[c'_i]$ denote the d-bit binary approximation (fraction) of c'_is obtained by truncating it off beyond the d^{th} bit.

THEOREM 3.1. *A target mixture is reachable with an error bound of* $\frac{1}{2^d}$ *in the desired CF of each input fluids, that is, a* (1 : 1) *mixing tree of height at most d can be constructed using the supply of input fluids, if and only if* $\sum_{i=1}^{N} \mathcal{B}[c'_i] \leq 1$, *where* $\sum_{i=1}^{N} \mathcal{B}[c'_i]$ *is the value obtained by binary addition.*

PROOF. To prove the sufficient condition, we note that the sum of the scaled target *CF*s is equal to $\sum_{i=1}^{N} c'_i = \left(\frac{c_1}{b_1} + \frac{c_2}{b_2} + \ldots + \frac{c_N}{b_N}\right)$. If $\sum_{i=1}^{N} \mathcal{B}[c'_i] = 1$ is satisfied, no additional buffer solution (D) is needed as input for preparing the mixture. Otherwise, the remaining portion of the buffer solution (D) is to be supplied as the input to maintain the *CF* of buffer solution D in the target mixture as $c'_{N+1} = 1 - \sum_{i=1}^{N} c'_i$, where $0 \leq c'_{N+1} < 1$. Note that each target *CF* c'_i has been approximated as a d-bit truncated binary fraction $\mathcal{B}[c'_i]$. Since the condition $\sum_{i=1}^{N} \mathcal{B}[c'_i] \leq 1$ holds, a mixing tree can always be constructed by *MinMix* [Thies et al. 2008] by including an extra part of the buffer solution D, if required. In this mixing tree, had each of the input reagents (represented by the leaf nodes) been pure, a target mixture with a ratio $\{c'_1 : c'_2 : \ldots : c'_N : c'_{N+1}\}$ would have been produced satisfying the given error bound. Therefore, when the input reagents are fed with the actual *CF*s $\{b_1, b_2, \ldots, b_N, D\}$, a mixture with a ratio $\{b_1.c'_1 : b_2.c'_2 : \ldots : b_N.c'_N : 1.c'_{N+1}\}$ will be produced. Since $c'_i = \frac{c_i}{b_i}$ for all i, $1 \geq i \geq N$, the resulting ratio will be $\{c_1 : c_2 : \ldots : c_N : c_{N+1}\}$. Further, as $b_i \leq 1$, the target ratio is also achieved within the given error bound.

The proof for necessity follows from the fact that the falsity of the condition $\sum_{i=1}^{N} \mathcal{B}[c'_i] \leq 1$ makes a (1 : 1) mixing tree of height d impossible to construct, as the sum of the constituent *CF*s in the target ratio including the buffer cannot exceed 1. □

The following corollary provides a simpler sufficient condition for reachability:

COROLLARY 3.2. *A target mixture is reachable by a mixing tree of height at most d using the supply of input fluids, with an error bound of* $\frac{1}{2^d}$ *in CF of each fluid, if* $0 < \sum_{i=1}^{N} \frac{c_i}{b_i} \leq 1$.

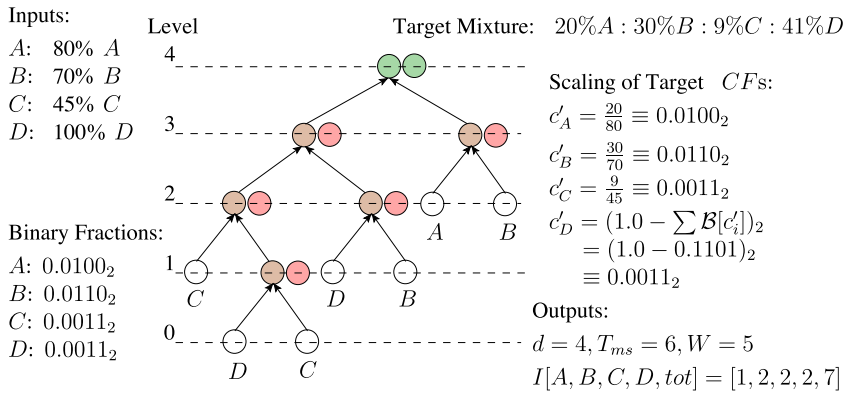


Fig. 4. Mixing tree obtained for a target mixture (20%A : 30%B : 9%C : 41%D) from a stock of 80% A, 70% B and 45% C, each diluted with the buffer solution D.

This follows from the observation that if the condition holds for real values of c_i's and b_i's, it will also hold when their d-bit truncated binary values are used in the sum irrespective of the choice of d.

3.2. Examples of Generalized Mixing

We will now illustrate the reachability conditions with several examples. Consider the target mixture M_1 as mentioned earlier. This belongs to Case (iv) of the reachability condition; in this case, $b_1 = 0.8, b_2 = 0.7, b_3 = 0.45; c_1 = 0.2, c_2 = 0.3, c_3 = 0.41$. Therefore, $\sum_{i=1}^{N} \mathcal{B}[c'_i] > 1$ even for $d = 2$. Hence, the target mixture M_1 is not reachable from the given stock even when an of error of 25% is allowed in the target *CF*s. On the other hand, the target mixture M_2, can be prepared from the same stock, because, $\sum_{i=1}^{3} \frac{c_i}{b_i} = 0.88 < 1$, and by Corollary 3.2, the reachability condition is satisfied. Figure 4 shows the mixing tree for producing one/two droplet(s) of the target mixture M_2: {20% A, 30% B, 9% C, 41% D}.

Consider the task of preparing another target mixture: {20% A, 30% B, 14.5% C, 35.5% D} from the same stock of input fluids: A, B and C with *CF* 80%, 70% and 45% respectively. In this case, $\sum_{i=1}^{3} \frac{c_i}{b_i} = 1.0008 > 1$. If we choose $d = 4$, the corresponding target *CF*s in 4-bit truncated binary fraction become 0.0100_2, 0.0110_2, and 0.0101_2, respectively. So, $\sum_{i=1}^{3} \mathcal{B}[c'_i] = 0.0100_2 + 0.0110_2 + 0.0101_2 = 0.1111_2 < 1$. Therefore, the reachability condition is satisfied for an accuracy level of 4. An extra amount of buffer solution D is needed with the input fluids to prepare the target mixture and the binary fraction corresponding to the extra requirement of D is 0.0001_2. Figure 5 shows the mixing tree constructed from the binary fractions corresponding to the target *CF*s of the reachable mixture.

In this 4-level mixing tree, when the input fluids A, B, C and D are supplied with *CF* 80%, 70%, 45%, and 100% respectively, the actual target ratio that is obtained is {20% A, 26.25% B, 14.0625% C, 39.6875% D}. Note that the desired target ratio and the obtained ratio are $\{A : B : C : D\} = \{0.0011_2, 0.0100_2, 0.0010_2, 0.0101_2\}$ and $\{0.0011_2, 0.0100_2, 0.0010_2, 0.0110_2\}$ respectively, when the *CF*s are approximated as

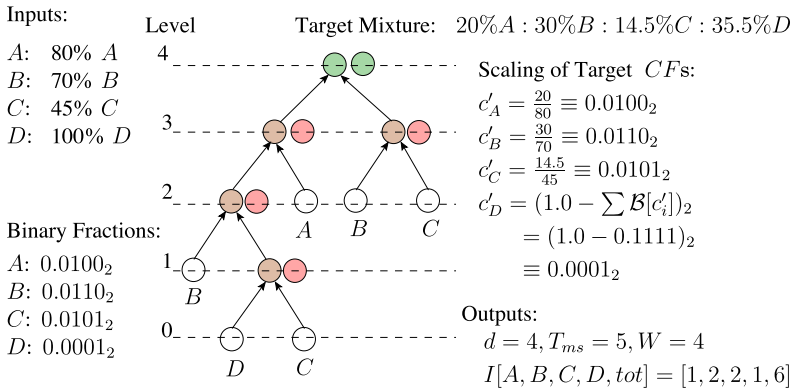


Fig. 5. Mixing tree obtained for a target mixture ($20\%A : 30\%B : 14.5\%C : 35.5\%D$) from a stock of 80% A, 70% B and 45% C, each diluted with the buffer solution D.

Table II. Reachability Conditions (Some Examples are Shown in Appendix A.1)

Instance	$\sum_{i=1}^{N} \frac{c_i}{b_i}$	$\sum_{i=1}^{N} \mathcal{B}[c'_i]$	Reachability	Excess D requirement
1	< 1.0	< 1.0	Reachable	D required (Ex.3)
2	< 1.0	$= 1.0$	Reachable	D not required (Ex.6)
3	< 1.0	> 1.0	Not Possible	—
4	$= 1.0$	< 1.0	Reachable	D required (Ex.4)
5	$= 1.0$	$= 1.0$	Reachable	D not required (Ex.5)
6	$= 1.0$	> 1.0	Not Possible	—
7	> 1.0	< 1.0	Reachable	D required (Ex.7)
8	> 1.0	$= 1.0$	Reachable	D not required
9	> 1.0	> 1.0	Non-Reachable	— (Ex.2)

4-bit-truncated binary fractions. The error in the CF of D is 0.0001_2, which satisfies the desired bound.

We list several instances of reachability conditions in Table II for different values of $\sum_{i=1}^{N} \frac{c_i}{b_i}$ and $\sum_{i=1}^{N} \mathcal{B}[c'_i]$ as stated in Theorem 3.1 and Corollary 3.2, along with the four cases (Case (i) to Case (iv)). Details of these instances and a few other examples demonstrating target mixture reachability for the four different cases (Case (i) to Case (iv)) are given in Appendix A.1, and in Table VII.

3.3. Proposed Algorithm: *GMA*

We now present an algorithm referred to as Generalized Mixing Algorithm (*GMA*), for generalized mixture preparation by incorporating the reachability conditions. The objective is to prepare a mixture with a desired ratio of CFs of N fluids $A_1, A_2, \ldots, A_N$ and buffer solution D, from their stock with arbitrary CFs (each diluted with D).

The pseudocode is given here as Algorithm 1. The algorithm returns a $(1 : 1)$ mixing tree whenever the target ratio is reachable from the given stock, otherwise, sends a message that it is not reachable. In *GMA*, we use the *MinMix* algorithm [Thies et al. 2008] in Step 15 and Step 19 on the binary fractions obtained in Step 11 and Step 18, respectively. For an accuracy level of d in target CFs, the time-complexity for producing the mixing tree by *GMA* will be $\mathcal{O}(dN)$ from N input CFs. In this mixing tree, the number of mix-split steps to reach the target will be minimum as in the *MinMix*

algorithm [Thies et al. 2008]. Note that in practical cases, an imprecise (unbalanced) mix-split operation may cause an unexpected concentration error in the target mixture. Therefore, in sample preparation, it is often desirable to minimize the number of mix-split steps. Also, the total number of input droplets (each of unit volume) required to produce the mixture, will be determined by the number of leaf nodes in the mixing tree, which is $\mathcal{O}(dN)$. Other mixing algorithms such as *RMA* [Roy et al. 2011b], *RSM* [Hsieh et al. 2012], *MTCS* [Kumar et al. 2013] or *CoDOS* [Liu et al. 2013] can also be used in Steps 15 and 19 of *GMA* for optimizing various cost parameters such as waste or reactant usage.

ALGORITHM 1: *GMA* $\left(\langle b_1, b_2, \ldots, b_N\rangle, \langle c_1, c_2, \ldots, c_N, c_{N+1}\rangle, d\right)$

1: **if** for *at least one* fluid A_i in the target ratio, $c_i > \left(b_i + \frac{1}{2^d}\right)$ **then**
2: **Return** an empty mixing tree $\mathcal{T}$ and the target mixture cannot be obtained from the input fluids. /*Case (i): *Target mixture is not reachable.*/
3: **end if**
4: **if** for *only one* fluid A_i in the target ratio, $\left(b_i + \frac{1}{2^d}\right) \geq c_i \geq b_i$ **then**
5: **Return** an empty mixing tree $\mathcal{T}$ and the input fluid A_i with *CF* b_i can be treated as the target droplet. /*Case (ii): *Target mixture is reachable by default within the error bound.*/
6: **end if**
7: **if** for *at least two* fluids A_i and A_j in the target ratio, $\left(b_i + \frac{1}{2^d}\right) \geq c_i \geq b_i$ and $\left(b_j + \frac{1}{2^d}\right) \geq c_j \geq b_j$ **then**
8: **Return** an empty mixing tree $\mathcal{T}$ and the target mixture cannot be obtained from the input fluids. /* Case (iii): *Target mixture is not reachable.* */
9: **end if**
10: **for all** i, $b_i > c_i \geq 0$ **do**
11: Compute $c_i' = \frac{c_i}{b_i}$ and express c_i' as a d-bit binary fraction $\{0.s_{i1}s_{i2}\ldots s_{i(d-1)}s_{id}\}$, which is rounded off after looking at the $(d+1)^{th}$ bit, where $s_{ij} \in \{0, 1\}$.
12: **end for**
13: Compute the binary bit-sum $\mathcal{BS} = \sum_{i=1}^{N} \mathcal{B}[c_i']$, by binary addition.
14: **if** $\mathcal{BS} == 1.0$ **then**
15: No excess D is required and a mixing tree $\mathcal{T}$ is constructed using N d-bit binary fractions $\mathcal{B}[c_i']$ for N input fluids by *MinMix*. /* Case (iv): *Target mixture is reachable.* */
16: **else if** $\mathcal{BS} < 1.0$ **then**
17: Excess D is required.
18: Determine d-bit binary fraction for D as $\mathcal{B}[D] = (1.0 - \mathcal{BS})$ with 0-bit padding in the right, if required.
19: A mixing tree $\mathcal{T}$ is constructed using $(N+1)$ d-bit binary fractions $\mathcal{B}[c_i']$ for N input fluids and D by *MinMix*. /* Case (iv): *Target mixture is reachable.* */
20: **else**
21: **Return** an empty mixing tree ($\mathcal{T}$) and the target droplet cannot be obtained directly from the input fluids. /* Case (iv): *Target mixture is not reachable.* */
22: **end if**

3.4. Reachability Study: Solution Space in $\mathbb{Z}^+$

In this subsection, we explore some further implications of the reachability condition. As an example, consider a stock of three input fluids A, B, C with certain *CF* values. For convenience of discussion in this subsection, we consider that each *CF* can only take

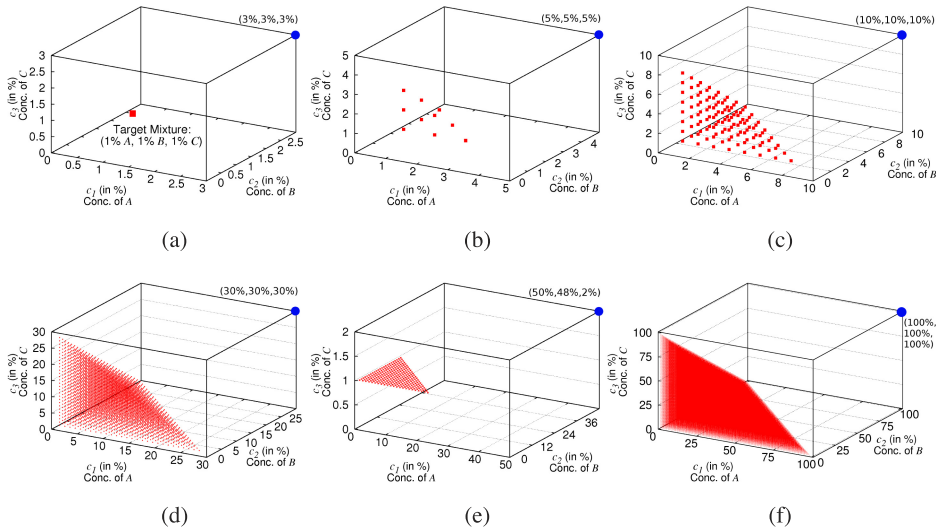


Fig. 6. Solution points in 3D space (of $\mathbb{Z}^+$) for three *CF*s (c_1, c_2 and c_3) of three fluids (A, B and C) in the target mixture, when the input *CF*s are: (a) {3% A, 3% B, 3% C}, (b) {5% A, 5% B, 5% C}, (c) {10% A, 10% B, 10% C}, (d) {30% A, 30% B, 30% C}, (e) {50% A, 48% B, 2% C}, and (f) {100% A, 100% B, 100% C}; the stock is also assumed to contain the *buffer* fluid.

an integral percentage value. We can identify all the target ratios which are reachable from this stock by checking the reachability condition. Figure 6 shows the results of our simulation experiments; the regions (in the 3D space) corresponding to all possible combinations of target *CF*s that are reachable from the respective input stock solutions are shown. For example, from the supply of three input fluids {3% A, 3% B, 3% C}, each diluted with the buffer solution D, only one target mixture {1% A, 1% B, 1% C} can be achieved in the solution space of $\mathbb{Z}^+$ (Figure 6(a)). Similarly, there are 10 target mixtures (i.e., solution points), which are reachable from the supply of three input fluids {5% A, 5% B, 5% C}. These solution points are shown in the 3D plot of Figure 6(b). For some other input combinations, for instance, {10% A, 10% B, 10% C}, {30% A, 30% B, 30% C}, {50% A, 48% B, 2% C}, {100% A, 100% B, 100% C}, the total number of target mixtures reachable from the input fluids (i.e., solution count) turns out to be 120, 4060, 276, and 161700, respectively, the 3D plots of which are shown in Figures 6(c)–(f).

We say that the *CF* values (in integral percentage) of the input fluids are "nearly-equally distributed" (E.D.), when they are equal or almost equal. If the *CF* value of one input fluid is much larger than the otherwise E.D. components, then we call such a ratio as "Skew". Table III presents the solution counts in $\mathbb{Z}^+$ space for the target ratios which are reachable from a stock of three, four, or five different fluids with a given input *CF*s. It is observed that all possible combinations of *CF*s (integral percentage values) in the target mixture can be achieved, when the input fluids are supplied at 100% concentration each.

Next, we define the *inverse reachability problem* as follows. Given the *CF* of each component fluid in the target mixture, the problem is to determine all the combinations of *CF*s in the stock solution from which the target mixture can be produced (i.e., reachable). We call them *valid inputs*. As before, for the sake of simulation experiments, we assume that each of the *CF* values of the inputs and the target ratio is in integral percentage only.

Table III.

The number of solutions in $\mathbb{Z}^+$ space for some examples of generalized mixing three or more fluids (where D is the buffer solution).

N	CFs of input fluids (diluted with D)	Ratio type	Size of target space
3	$\langle A_1, A_2, A_3 \rangle$		
	33%,33%,34%	E.D.	5456
	10%,10%,10%	E.D.	120
	3%,3%,94%	Skew	31
	3%,3%,17%	Skew	5
	3%,3%,3%	E.D.	1
	2%,2%,4%	E.D.	0
4	$\langle A_1, A_2, A_3, A_4 \rangle$		
	25%,25%,25%,25%	E.D.	12650
	10%,10%,10%,10%	E.D.	210
	4%,4%,4%,88%	Skew	22
	4%,4%,4%,37%	Skew	9
	4%,4%,4%,4%	E.D.	1
	3%,3%,3%,6%	E.D.	0
5	$\langle A_1, A_2, A_3, A_4, A_5 \rangle$		
	20%,20%,20%,20%,20%	E.D.	15334
	15%,15%,15%,15%,17%	E.D.	3038
	5%,5%,5%,5%,80%	E.D.	16
	5%,5%,5%,5%,80%	E.D.	16
	5%,5%,5%,5%,80%	E.D.	16

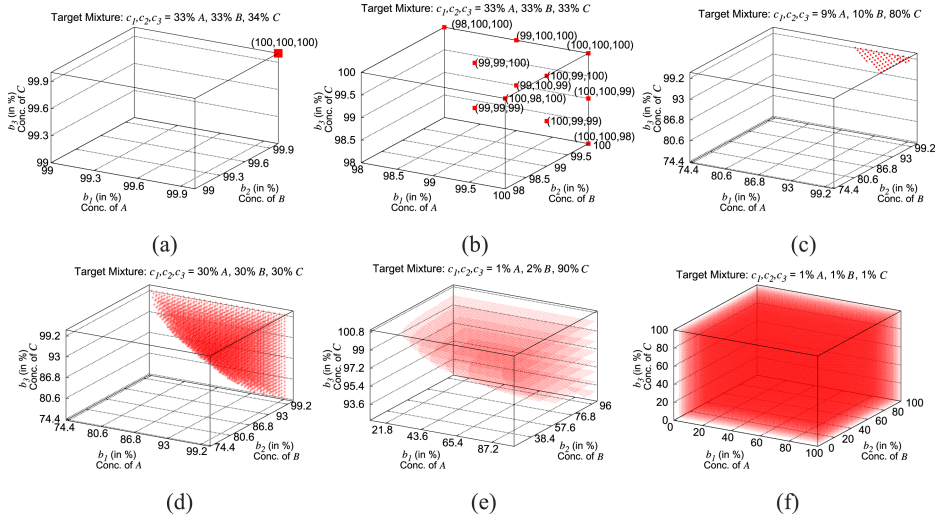


Fig. 7. Solution points in 3D space (of $\mathbb{Z}^+$) for three input CFs (b_1, b_2 and b_3) of three fluids (A, B and C), when the constituent CFs in the target mixture are: (a) {33% A, 33% B, 34% C, 0% *buffer*}, (b) {33% A, 33% B, 33% C, 1% *buffer*}, (c) {9% A, 10% B, 80% C, 1% *buffer*}, (d) {30% A, 30% B, 30% C, 10% *buffer*}, (e) {1% A, 2% B, 90% C, 7% *buffer*}, and (f) {1% A, 1% B, 1% C, 97% *buffer*}.

Figure 7 presents the solution points (in 3D space) corresponding to the valid input CFs from which a specific target mixture is reachable. For example, to achieve a target mixture of A, B and C with the ratio of CFs as {33% A : 33% B : 34% C : 0% D}, where D is the buffer solution, there is only one stock solution with CFs of the input fluids $b_1 = b_2 = b_3 = 100\%$ in the solution space of $\mathbb{Z}^+$ (Figure 7(a)). Similarly, there are

11 valid combinations of *CF*s of input stock from which the target mixture with the ratio of *CF*s as $\{33\%\ A : 33\%\ B : 33\%\ C : 1\%\ D\}$ can be prepared, where D is the buffer solution. These solution points are shown in the 3D plot of Figure 7(b). For some other target mixtures, where the *CF*s of the buffer solution D are 1%, 7%, 10% and 97%, the total number of possible combinations of *CF*s of the input fluids (i.e., solution counts) increases drastically as shown in Figures 7(c)–(f).

Note that inverse reachability is helpful in judicious stock planning while preparing a sample. For instance, as observed earlier, the mixture $\{33\%\ A, 33\%\ B, 33\%\ C, 1\%\ D\}$ can be produced from 11 valid inputs; among them, $\{98\%\ A, 100\%\ B, 100\%\ C\}$, $\{100\%\ A, 100\%\ B, 100\%\ C\}$, $\{100\%\ A, 98\%\ B, 100\%\ C\}$, $\{100\%\ A, 100\%\ B, 98\%\ C\}$ require 11 $(1:1)$ mix-split steps to produce it with an accuracy level $d = 9$. The remaining other input combinations will require 12 to 14 mix-split steps. Thus, the mixing time or the cost of reactants can be optimized by making a suitable choice among the valid inputs which are available as stock.

THEOREM 3.3. *If in the target mixture, there is no buffer solution D mixed with other component fluids, that is, if $c_D = 0\%$, then there is only one input combination (solution in $\mathbb{Z}^+$) from which the target mixture is reachable. This unique solution is the input combination for which the CF of each component fluid, except the buffer solution, is 100%.*

PROOF. If for a target mixture of N component fluids (except buffer solution D), $c_1 + c_2 + \ldots + c_N = 100\%$, then there is no trace of buffer in the target mixture. Hence, each of the N input fluids $A_1, A_2, \ldots, A_N$ should not contain any buffer solution D, that is, they are not diluted with D and each of them should have a *CF* of 100%. Thus, there exists only one solution for the input stock, that is, $b_1 = b_2 = \ldots = b_N = 100\%$. □

Table IV presents the solution counts in $\mathbb{Z}^+$ space for some example cases of generalized mixing from arbitrary concentrations of three, four, and five different fluids to a target mixture. It is observed that if the target mixture does not contain any buffer (D), then there is only one stock of input fluids (all having 100% *CF*), from which this mixture can be prepared.

The simulated results on the number of solutions with increasing portion of the buffer solution D in the target mixture are depicted in Figures 8(a)–(b). From the data presented in Table IV and the nature of the curves in Figure 8, we observe that while the portion of buffer solution (D) in the target mixture increases, the number of possible combinations of *CF*s of input fluids (i.e., the solution counts) grows exponentially. Moreover, it is observed that compared to the "E.D." ratios, this growth in the number of solutions with the increasing portion of the buffer solution D in the target mixture is faster for the "Skew" ratios.

4. DILUTION FROM TWO OR MORE ARBITRARY CONCENTRATIONS OF THE SAME FLUID

During sample preparation with a DMF biochip, droplets of a reagent with multiple *CF* values are often obtained as a byproduct. Instead of being discarded, droplets may be used to replenish the stock and they can also be utilized in various ways for generalized mixture preparation as discussed in the previous subsection. Here, we show how they can be efficiently used for diluting a sample. Two earlier dilution methods, namely *DMRW* [Roy et al. 2010] and *IDMA* [Roy et al. 2011a] can also be used to dilute a sample S from a supply of two arbitrary bounding *CF*s of S. However, these methods cannot minimize the number of mix-split steps; instead, they are targeted towards optimizing waste droplets. Till date, to the best of our knowledge, there exists no dilution algorithm that produces a sample S of desired dilution with a minimum number of

Table IV.

The number of solutions in $\mathbb{Z}^+$ space for some examples of generalized mixing (where D is the buffer solution).

N	CFs in target mixture	Ratio type	Size of input space
	$\langle A_1, A_2, A_3, D\rangle$		
3	33%,33%,34%,0%	E.D.	1
	33%,33%,33%,1%	E.D.	11
	9%,10%,80%,1%	Skew	66
	30%,30%,30%,10%	E.D.	4445
	1%,2%,90%,7%	Skew	31333
	1%,1%,1%,97%	E.D.	969989
	$\langle A_1, A_2, A_3, A_4, D\rangle$		
4	25%,25%,25%,25%,0%	E.D.	1
	24%,24%,24%,24%,4%	E.D.	3801
	1%,1%,1%,96%,1%	Skew	51850
	19%,19%,19%,19%,24%	E.D.	2203171
	6%,6%,6%,6%,76%	E.D.	64840143
	1%,1%,1%,1%,96%	E.D.	95989659
	$\langle A_1, A_2, A_3, A_4, A_5, D\rangle$		
5	20%,20%,20%,20%,20%,0%	E.D.	1
	19%,19%,19%,19%,19%,5%	E.D.	109071
	1%,1%,1%,1%,91%,5%	Skew	78559108
	15%,15%,15%,15%,15%,25%	E.D.	126176046
	9%,9%,9%,9%,12%,52%	E.D.	2068926241
	1%,1%,1%,1%,66%,30%	Skew	2069867322

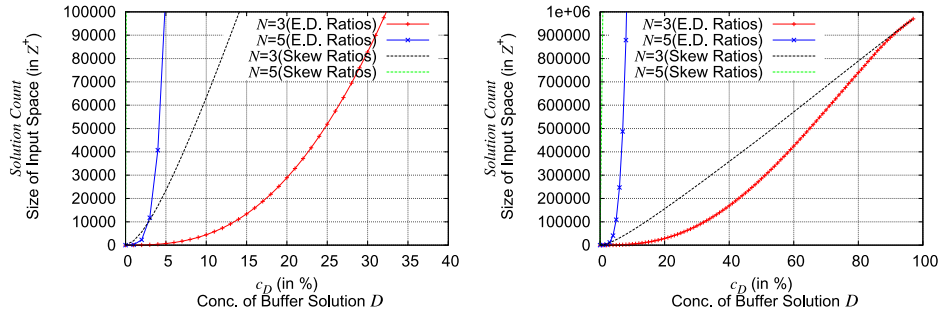


Fig. 8. Variation of the number of solutions in multi-dimensional space (of $\mathbb{Z}^+$) for input CFs (b_1, b_2 and b_3) of three fluids (A, B and C) with increasing concentration of buffer D in the target mixture. (a)–(b) Increasing the ranges of scales along X- and Y-axes.

mix-split steps from a supply of N arbitrary CFs of S, $N \geq 3$ (i.e., when the inputs are related). This problem was mentioned as one of the open problems by Thies et al. [2008]. This motivates us to formulate a more generalized dilution algorithm, which can be used to solve the given problem.

First, for $N = 2$, we show that a simple scaling technique can be used to optimize the mix-split sequence that produces, in $\mathcal{O}(d)$ time, the target droplets with a maximum error of $\frac{1}{2^{d+1}}$ in its CF. For $N \geq 3$, we conjecture that the problem is *NP-hard* and propose a heuristic algorithm based on pseudo-polynomial time dynamic programming.

4.1. Dilution from Any Two Arbitrary Concentrations of the Same Fluid

The problem of generalized dilution of a fluid from *two* arbitrary concentrations of the same fluid can be formally stated as follows.

Inputs

(a) A target *CF* C_t of a sample S, where C_t is a positive real number and $0 \leq C_t < 1$.
(b) Two arbitrary *CF*s, one lower (C_ℓ) and one higher (C_h), of S, each diluted with a buffer D, where $0\% \leq C_\ell < C_t < C_h \leq 100\%$.
(c) An integer d to represent the accuracy level of C_t.

Output. A dilution tree $\mathcal{T}$ with depth at most d representing a sequence Z_d of $(1:1)$ mix-split steps to produce target droplets with $CF = C_t$ using only C_ℓ and C_h.

The two input *CF*s are considered in exactness, that is, without any approximation. For the dilution problem, we represent C_t by rounding-it off as a d-bit binary fraction. Thus, the maximum error in C_t will be bounded above by $\frac{1}{2^{d+1}}$.

As discussed earlier, the *twoWayMix* algorithm [Thies et al. 2008] cannot be used to dilute a sample S starting from two arbitrary *CF*s of S. In order to solve this problem, a scaling technique can be used as described in the following.

4.1.1. Scaling Technique for Generalized Dilution. The basic concept of scaling of concentration values has been already discussed in Section 3.1. With the three inputs C_ℓ, C_h and C_t, and the error bound $\frac{1}{2^{d+1}}$, three cases may arise: (i) $C_t > \left(C_h + \frac{0.5}{2^d}\right)$ or $C_t < \left(C_\ell - \frac{0.5}{2^d}\right)$, (ii) C_t is in the range of $\left(C_h \text{ to } (C_h + \frac{0.5}{2^d})\right)$ or $\left(C_\ell \text{ to } (C_\ell - \frac{0.5}{2^d})\right)$, and (iii) C_t is in the range of $C_\ell \leq C_t \leq C_h$.

For Case (i), the desired target *CF* is not reachable from the two *CF*s supplied as inputs. For Case (ii), the desired target *CF* is reachable from the two input *CF*s and the input droplet with either C_h or C_ℓ can be treated as the target droplet according to the value of C_t in the range of $\left(C_h \text{ to } (C_h + \frac{0.5}{2^d})\right)$ or $\left(C_\ell \text{ to } (C_\ell - \frac{0.5}{2^d})\right)$, respectively. For the third case (Case (iii)), we scaling C_t to C'_t as follows.

Note that the desired value of C_t is taken as input assuming a scale (Scale-1) between $CF = C_\ell$ and $CF = C_h$. In order to map C_t (from Scale-1) to C'_t in Scale-2 (between $CF = 0$ and $CF = 1$), we need to apply the following transformation given by Equation (3) (see Figure 10).

$$\frac{C'_t - 0}{1 - 0} = \frac{C_t - C_\ell}{C_h - C_\ell}$$

$$\text{Therefore, } C'_t = \frac{C_t - C_\ell}{C_h - C_\ell} \tag{3}$$

We then represent C'_t by rounding-it off as a d-bit binary fraction.

4.1.2. Proposed Algorithm: GDA. In order to determine the sequence of $(1:1)$ mix-split steps required for generalized dilution from two arbitrary input *CF*s, we propose a scheme, called as Generalized Dilution Algorithm (*GDA*), the pseudo code of which is provided in Appendix A.2. The algorithm *GDA* calls *twoWayMix* [Thies et al. 2008] to produce C'_t. It terminates in $\mathcal{O}(d)$ time to determine a dilution tree with a minimum number of $(1:1)$ mix-split steps (at most d). Also it does not need any intermediate storage units.

The following theorem formalizes the transformation given by Equation (3). The proof is provided in Appendix A.3.

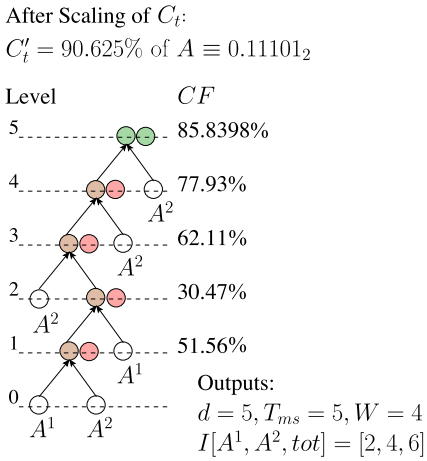


Fig. 9. The dilution tree obtained by *GDA* for an example of generalized dilution with desired $CF = 85.84\%$.

THEOREM 4.1. *If the d-bit binary representation of C'_t, where $C'_t = \dfrac{C_t - C_\ell}{C_h - C_\ell}$, is used with the two input CFs C_h and C_ℓ, then* GDA *produces two target droplets of $CF = C_t$ with an error in CF less than or equal to $\frac{1}{2^{d+1}}$.*

In contrast to *DMRW* [Roy et al. 2010] and *IDMA* [Roy et al. 2011a], the proposed dilution algorithm *GDA* can prepare the target droplets of desired concentration from any two input fluids with impure concentration levels without any need of storing intermediate droplets, just as in the case of *twoWayMix*. Other recently designed dilution algorithms such as *REMIA* [Huang et al. 2012], *GORMA* [Chiang et al. 2013] or *WARA* [Huang et al. 2013], can recycle the intermediate waste droplets in the dilution process; however, they do not provide any clue on how a target concentration can be achieved from the supply of random concentrations. The proposed dilution algorithm *GDA* solves this problem by using a simple transformation based on scaling.

4.1.3. Examples of Generalized Dilution. An example of producing a target droplet with *CF* $C_t = 85.84\%$ from the supply of two input *CF*s 93.75% and 9.375% of the same fluid is shown in Figure 9.

GDA first transforms $C_t = 85.84\%$ from Scale-1 to $C'_t = 90.625\%$ in Scale-2, which is represented as a 5-bit rounded-off binary fraction 0.11101_2, for $d = 5$. Then it scans this bit-string from right-to-left to determine the sequence of $(1:1)$ mix-split steps for producing the target droplet(s) from the two input *CF*s $C_h = 93.75\%$ and $C_\ell = 9.375\%$. *GDA* outputs the corresponding dilution tree shown in Figure 9(b). Note that the error in C_t is $(|85.84\% - 85.8398\%|)$, that is, 0.0002%, which is much less than the allowable error of $\frac{1}{64}$ or 1.5625%, for the given choice of d.

For comparison, we consider at most 10 $(1:1)$ mix-split steps in our simulation experiments, and assume that there is only one mixing module on-chip. Simulated results for *DMRW* [Roy et al. 2010], *IDMA* [Roy et al. 2011a], and *GDA*, are compared, among which the first two schemes may require a number of on-chip storage units (electrodes) for storing the intermediate droplets produced during the sequential mix-split cycles. The following output parameters are considered for comparison — the total number of $(1:1)$ mix-split steps (T_{ms}), the total number of waste droplets (W), the total number of storage units required (q) and the demand array $I[d_1, d_2, tot]$, where d_1 and

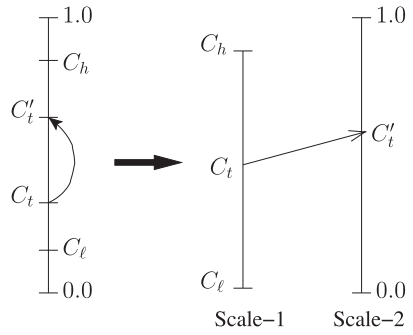


Fig. 10. Mapping of target *CF* between two scales.

Table V.

Comparative values of the total number of (1 : 1) mix-split steps T_{ms}, the number of waste droplets W, the number of required storage units q, and demand array I, for some examples.

Example	(C_ℓ, C_h, C_t); Transformed C'_t (≡ binary fraction)	$T_{ms}^{\dagger}$			$W^{\ddagger}$			$q^{\$}$			$I(C_h, C_\ell)^{\S}$		
		DMRW	*IDMA*	*GDA*	*DMRW*	*IDMA*	*GDA*	*DMRW*	*IDMA*	*GDA*	*DMRW*	*IDMA*	*GDA*
Ex.1	(0%, 100%, 50.10%)* 50.10% (≡ 0.1000000001_2)	14	10	10	9	2	9	9	2	0	[6,5]	[2,2]	[2,9]
Ex.2	(0%, 100%, 20.02%)* 20.02% (≡ 0.0011001101_2)	13	13	9	3	3	8	4	4	0	[1,4]	[1,4]	[5,5]
Ex.3	(2.25%, 84.375%, 30.76%) 34.72% (≡ 0.01011001_2)	13	13	8	4	4	7	6	6	0	[2,4]	[2,4]	[4,5]
Ex.4	(9.86%, 72.17%, 39.55%) 47.65% (≡ 0.0111101_2)	12	13	7	5	4	6	8	6	0	[3,4]	[3,3]	[5,3]
Ex.5	(21.78%, 89.16%, 50.10%) 42.03% (≡ 0.011010111_2)	15	14	9	5	5	8	11	9	0	[3,4]	[3,4]	[6,4]

* No transformation is required, that is, $C'_t = C_t$, † number of mix-split cycles, ‡ number of waste droplets, $ number of storage units required and § number of input droplets used.

d_2 indicate the required numbers of input droplets of *CF*s C_ℓ and C_h, and *tot*, which is the total number of input droplets used, that is, $tot = d_1 + d_2$.

In Table V, the results are shown for five example cases, and for each of them, we show the transformed C'_t represented as a d-bit binary fraction, which is rounded-off by considering the $(d+1)^{th}$ bit. Note that for the first two examples, no transformation is required, whereas, for the last three examples C_t is transformed into another *CF* C'_t. As evident from these examples, *GDA* outputs the dilution tree with the least number of (1 : 1) mix-split steps (T_{ms}) and does not need any storage unit (q).

4.2. Dilution from Three or More Arbitrary Concentrations of the Same Fluid

In this section, we present a scheme for generalized dilution of a sample fluid A from the supply of N arbitrary concentrations (referred to as *related* inputs) of A, each diluted with the same buffer. This problem of generalized dilution is important from a stock-management perspective. Multiple concentrations of a drug, often referred to as dilution gradient [Bhattacharjee et al. 2013], are needed in pharmaceutical design where the same drug with different concentration levels are to be tested for their efficacy. These drugs are expensive and often there is a global crisis of the raw drug (100%) in stock [Ventola 2011; Talsma 2013]. Thus, there is a strong need of recycling the waste droplets or reuse of available concentrations. In this particular problem, we have addressed how dilution of a sample can be accomplished by using a recycling technique. Note that the classical *MinMix* method [Thies et al. 2008] produces droplets with different concentration levels as waste, in each step, while diluting a sample.

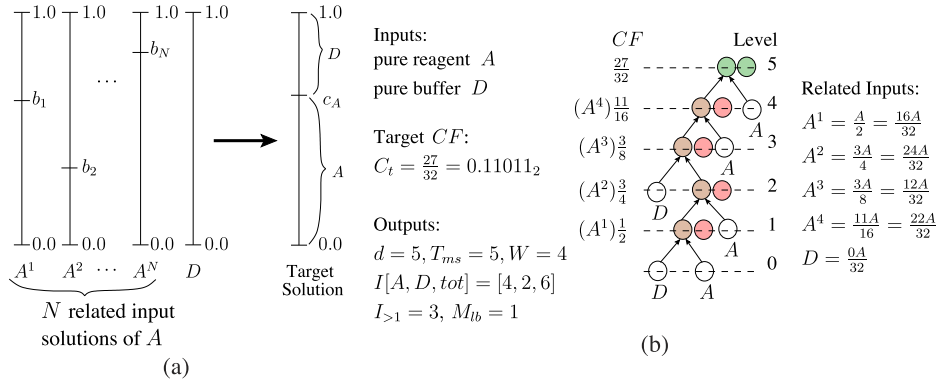


Fig. 11. (a) Related input and target *CF*s of same fluid in the scale 0 to 1. (b) An example of producing related input fluids as by-products of dilution process for $C_t = \frac{27}{32}$ by *twoWayMix* [Thies et al. 2008].

These droplets (or the dilutions of the same sample that are available in stock) can thus be reused to produce a new target concentration level. Hence, a solution of this problem has high practical significance from the viewpoint of reactant savings and error recovery during sample preparation [Hsieh et al. 2014]. Also, this was posed as an open problem by Thies et al. [2008]. We formulate the generalized dilution problem for $N \geq 3$ as follows.

Inputs

(a) N inputs $A^1, A^2, \ldots, A^N$ ($N \geq 2$) of a fluid A with *CF*s $b_1, b_2, \ldots, b_N$, respectively, each diluted with the buffer solution D, where b_i is a positive real number and $0 < b_i \leq 1$.
(b) A pure buffer solution D is available in stock. Thus, the total number of inputs including D is $(N + 1)$, where $(N + 1) \geq 3$.
(c) Desired *CF* C_A of the fluid A, where C_A is a positive real number and $0 \leq C_A < 1$.
(d) An integer d, which denotes the required accuracy level of C_A.

Output. A dilution/mixing tree $\mathcal{T}$ of depth at most d, representing a task graph of $(1 : 1)$ mix-split steps for producing diluted target droplets of A with *CF* C_A.

Note that if $C_A > \left(\max_i\{b_i\} + \frac{0.5}{2^d}\right)$ $(\forall i, i = 1 \text{ to } N+1)$, the desired *CF* is not reachable from the inputs. Again, if C_A is in the range of $\left(b_i \text{ to } (b_i + \frac{0.5}{2^d})\right)$ for any i ($i = 1$ to $N + 1$), then the desired *CF* is reachable and the input droplet with $CF = b_i$ can be treated as the target droplet. For all other cases, we need a precise formulation of this problem, because the input *CF*s are to be chosen appropriately among N values so that the number of mix-split steps is minimized (see Figure 11(a)).

Let T_{ms} be the total number of nonleaf nodes representing $(1 : 1)$ mix-split steps in the desired dilution/mixing tree. The height d' of the dilution/mixing tree will be equal to or less than the accuracy level d of the target *CF*, that is, $d' \leq d$. Let M_{lb} be the minimum number of mixers required for the earliest possible completion of the dilution/mixing tree.

For a dilution/mixing tree, we maintain an input-fluid-demand array $I[A^1, \ldots, A^{N+1}, tot]$ that stores the required number ($I[A^i]$) of droplets of fluid A^i, and the total number of required input droplets (tot). The input droplets may be obtained by salvaging the waste droplets produced by other protocols. In many sample preparation algorithms based on the $(1 : 1)$ mix-split model, one droplet is usually thrown as waste.

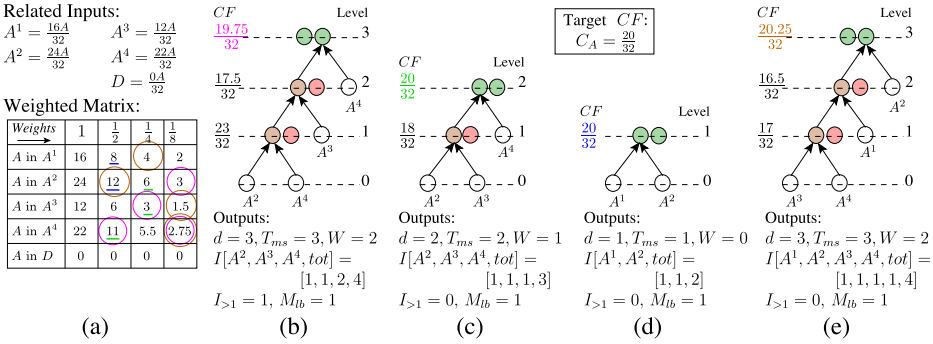


Fig. 12. (a) Weighted matrix for related input *CF*s. (b)–(e) Four different dilution trees for the target *CF* of $\frac{20}{32}$.

Hence, in the proposed algorithm, we use a parameter $I_{>1}$ to indicate how many extra (more than one) droplets of each *CF* (except the buffer D) are required as inputs to the dilution/mixing tree.

We use T_{ms}, d', M_{lb} and $I_{>1}$ as different performance parameters to characterizing the dilution/mixing tree. The proposed algorithm, described next, can be tuned to cater to different optimization criteria as needed by the user: (i) T_{ms} is minimized, or (ii) d' is minimized, or (iii) M_{lb} is minimized, or (iv) $I_{>1}$ is minimized.

4.2.1. Recycling of Waste Droplets for Dilution. First, we show an example that how multiple *CF*s of a fluid A (A^1,A^2,A^3 and A^4) can be obtained as waste (by-product). Consider the production of a target *CF* $C_t = \frac{27}{32}$ from its 100% sample and buffer by running *twoWayMix* [Thies et al. 2008]. The corresponding dilution tree is shown in Figure 11(b). We now demonstrate how these waste droplets can be recycled to produce another target droplet with $C_A = \frac{20}{32}$, if a method of generalized dilution can be used.

We show four different dilution trees for this target (*CF* $C_A = \frac{20}{32}$) in Figure 12(b)–(e). The best solution among them in terms of minimum number of mix-split steps is shown in Figure 12(d). However, if only three related inputs A^2,A^3 and A^4 are available after A^1 is used in some other dilution process, then the dilution tree shown in Figure 12(c) is the best solution, as it leads to lower T_{ms} and consumes a fewer number of total input droplets (obtained from $I[\ ]$) than those of the tree shown in Figure 12(b).

A solution (dilution tree) can be envisaged with the help of a *weighted matrix* ($\mathcal{W}$) that is computed from the *CF* values of the related inputs as shown in Figure 12(a). For A^i, the component of A that contributes to the target concentration follows a geometric progression with a common ratio of $\frac{1}{2}$, depending on the level of the tree at which it is used. For example, in Figure 12(a) $\mathcal{W}$ consists of 24, 12, 6, 3 for A^2. In the dilution tree of Figure 12(c), the contributions of A^2 and A^3 to the target *CF* are $\frac{6}{32}$ and $\frac{3}{32}$, respectively, as they are at depth 2 of the tree, whereas that of A^4 is $\frac{11}{32}$, as it is at depth 1 of the tree. These contributions are easy to visualize based on the basic principles of dilution. The selection of contributions from $\mathcal{W}$ for the target *CF* provides the binary fractions of the constituent fluids corresponding to a dilution/mixing tree. For example, the binary fractions of A^2, A^3 and A^4 for the dilution tree shown in Figure 12(c) can be written as 0.01_2, 0.01_2 and 0.10_2, respectively. A selection of numbers from $\mathcal{W}$ is *valid*, if a dilution/mixing tree can be constructed using *MinMix* [Thies et al. 2008] from the binary fractions corresponding to the selection.

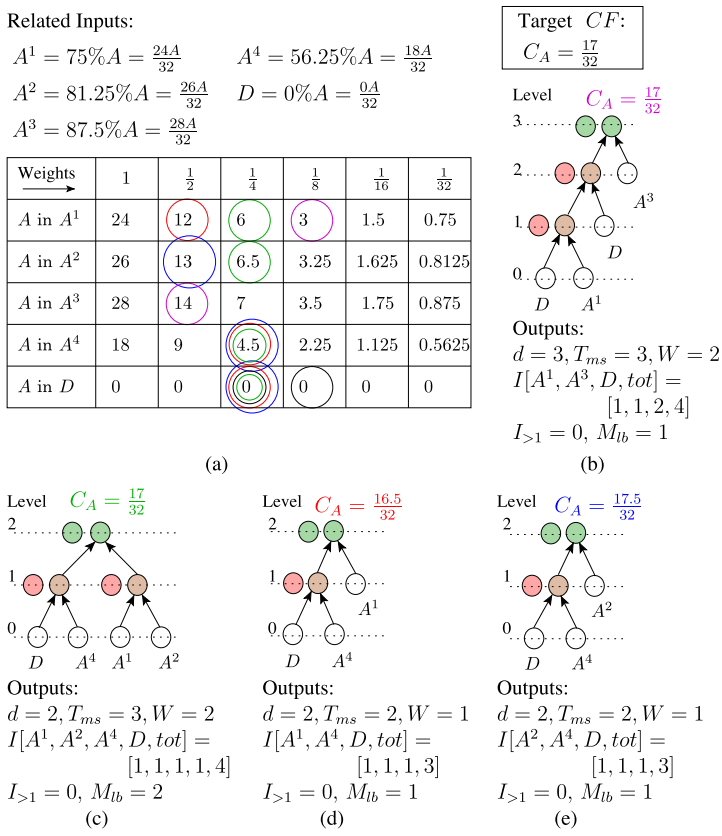


Fig. 13. (a) Weighted matrix and date for constructing the dilution/mixing trees. Dilution/Mixing trees for preparing a sample fluid A with target CF $C_t = \frac{17}{32}$ obtained using a heuristic scheme. (b)–(c) Two suboptimal solutions, (d)–(e) Two best solutions.

As another example of generalized dilution, Figure 13 shows the dilution/mixing trees for preparing a sample fluid A with target $CF = \frac{17}{32}$ from the supply of five related inputs of fluid A with the following CFs:b $\frac{24}{32}$, $\frac{26}{32}$, $\frac{28}{32}$, $\frac{18}{32}$ and $\frac{0}{32}$. The last CF in this list refers to the buffer solution D.

4.2.2. Problem Formulation and Proposed Algorithm. First, we represent b_i as $\frac{B_i}{2^d}$ (for all i) and C_A as $\frac{T}{2^d}$, where B_is and T are positive real numbers. For each input A^i, $(d+1)$ geometric progression (G.P.) terms with the first term B_i and common ratio $\frac{1}{2}$, that is, $B_i, \frac{B_i}{2}, \frac{B_i}{4}, \cdots, \frac{B_i}{2^d}$, are stored in an one-dimensional array $\mathcal{W}[i]$. A two-dimensional array $\mathcal{W}$ of size $(N+1) \times (d+1)$ stores the G.P. terms for all N inputs and D. Each element $\mathcal{W}[i,j]$ is associated with a weight $w_j = \frac{1}{2^j}$ and corresponds to the contribution $\mathcal{W}[i,j].w_j$ of fluid A^i, if it is at level $(d'-j)$ of the dilution/mixing tree of height d' to be determined. Next, a set Q is formed with all the elements of $\mathcal{W}$ except the elements of the row for D. An optimal dilution/mixing tree (based on any of the optimality criteria mentioned earlier) can be constructed by selecting some of the elements as a set R from Q (i.e., $R \subset Q$) and using some additional amount of D, if needed. While determining R from Q, the following parameters are computed: (a) the total weight of all the elements

in R as $V = \sum_{\forall x,\, x \in R} w_j$, (b) the sum of all the elements in R as $T' = \sum_{\forall x,\, x \in R} x$, that is, the target CF becomes $\frac{T'}{2^d}$, and (c) the height of the underlying dilution/mixing tree as $d' = max(j)$ such that for all i,j, $\mathcal{W}[i,j]$ is selected in R (hence, $0 < d' \leq d$).

Note that, if $V > 1$, the underlying dilution/mixing tree cannot be constructed and hence the target is not reachable, thereby showing an *invalid* selection of elements in R; in other cases, the target is reachable. Moreover, if $V = 1$, no extra buffer (D) droplet is required to achieve the target CF (i.e., $C_D = 0$), whereas if $V < 1$, extra buffer (D) is needed (i.e., $C_D \neq 0$). Let $\mathcal{B}[x.w_j]$ represent the binary fraction of the decimal value $x.w_j$, which indicates the contribution of A^i, where $x \in R$, $x = \mathcal{W}[i,j]$ and w_j is the corresponding weight value. Hence, $|R|$ indicates the number of droplets of input fluids (different x values) with related (arbitrary) CFs of A chosen in R to achieve $T' = \sum_{\forall x,\, x \in R} x$. The binary fraction for C_D can be represented as $\mathcal{B}[C_D] = \big(1 - \sum_{\forall x,\, x \in R} \mathcal{B}[x.w_j]\big)$ (obtained by binary addition and subtraction). Let $\beta_{d'}(D)$ be the total number of 1s in the d'-bit binary fraction for buffer D, that is, $\mathcal{B}[C_D]$. Hence, $\beta_{d'}(D)$ indicates the total number of buffer droplets used to construct the mixing tree. Thus, the sum total $(|R| + \beta_{d'}(D))$ is the total number of input droplets of sample/reagents and buffer solution (D), that is, the total number of input droplets of reactant fluids.

Problem Statement. Find a subset R from the set Q (i.e., $R \subset Q$), such that $(|R| + \beta_{d'}(D))$ is minimized, where $T' = \sum_{\forall x,\, x \in R} x$ and $|T' - T| \leq 0.5$ (i.e., $T' = T \pm 0.5$).

We conjecture that this problem is *NP-hard* [Garey and Johnson 1979]. The subset-sum problem (SSP) [Garey and Johnson 1979] can be reduced to this problem. It is known that SSP is *weak NP-Complete* and it admits a pseudo-polynomial algorithm [Garey and Johnson 1979; Kleinberg and Tardos 2005] based on dynamic programming. A solution (optimal or suboptimal) to this problem provides a dilution/mixing tree of height d' and the corresponding selection of R can be represented with a binary matrix $\mathcal{X}$ of size $(N + 1) \times (d' + 1)$, where $\mathcal{X}[i,j] \in \{0, 1\}$. If $\mathcal{W}[i,j]$ is selected in R, $\mathcal{X}[i,j] = 1$, otherwise $X[i,j] = 0$. Hence, $\mathcal{X}[i]$ is a vector of $(d' + 1)$ bits representing the binary fraction for input A^i (with binary point between $\mathcal{X}[i]\,[0]$ and $\mathcal{X}[i]\,[1]$) and $\mathcal{X}[N + 1]$ represents the binary fraction for C_D, that is, $\mathcal{B}[C_D]$.

We propose a heuristic Extended Generalized Dilution Algorithm (*EGDA*), written as Algorithm 2, which is based on the pseudo-polynomial time dynamic programming approach, and its time complexity is bounded above by a polynomial function of two variables $|Q|$ and T. In practical cases, $|Q| = N.(d + 1)$. Thus, the complexity of our algorithm is $\mathcal{O}(N.d.T)$, where N is the number of input concentrations available, d is the accuracy level, and T is determined by the target CF, which can be at most 2^d. Similarly, N can also be at most $(2^d + 1)$. Note that $d = 10$ implies that the error in target CF will be at most $\frac{1}{2048}$, which is extremely small for all practical purposes. Thus, for real applications, $d < 10$ and the two input parameters N and T can be treated as constants. Hence, the complexity will not grow exponentially. Moreover, the proposed pseudo-polynomial algorithm for dynamic programming first constructs a sorted array from a set Q of positive numbers. Therefore, while obtaining a valid solution (i.e., one choice of R), it selects the elements in R one-by-one from Q such that the sum of selected elements remains within T. In other words, the algorithm implicitly prunes the search graph on the fly without enumerating all possible solutions. Hence, the proposed algorithm *EGDA* does not suffer from runtime explosion for all practical purposes.

ALGORITHM 2: *EGDA* $\left(\langle b_1, b_2, \ldots, b_N \rangle, C_A, d\right)$

1: Set $b_{N+1} = 0$ (for buffer solution, D) and dilution/mixing tree $\mathcal{T} = \Phi$.
2: Represent b_i as $\frac{B_i}{2^d}$ (for all i) and C_A as $\frac{T}{2^d}$, where B_is and T are positive real numbers.
3: **for** $i = 1$ to $N + 1$ **do**
4: Set $\mathcal{W}[i][0] = B_i$.
5: **end for**
6: Construct $\mathcal{W}$ by $(d + 1)$ G.P. terms with $\mathcal{W}[i][0]$ as the first term and $\frac{1}{2}$ as the common ratio.
7: Obtain set Q with all the numbers (integers or reals) of $\mathcal{W}$ except the $(N + 1)^{th}$ row.
8: For each number $x \in Q$, associate an weight $w_x = \frac{1}{2^j}$, if x is j^{th} G.P. term in $\mathcal{W}$.
9: Find R from Q, where $R \subset Q$ (Subset-Sum Problem), by using the pseudo-polynomial time dynamic programming approach [Kleinberg and Tardos 2005]. A solution to this problem is a binary matrix $\mathcal{X}$ (i.e., R) denoting the selection of numbers from $\mathcal{W}$ (i.e., Q), such that $|T - T'| < 0.5$, where $T' = \sum_{\forall x,\, x \in R} x$. More than one solutions can be obtained.
10: **for all** the solutions **do**
11: Compute $V = \sum_{\forall x,\, x \in R} w_x$.
12: **if** $V \leq 1$ **then**
13: The solution is valid.
14: **end if**
15: **end for**
16: **for all** the valid solutions **do**
17: Compute $d' = max(j)\ s.t.\ \mathcal{X}[i][j] = 1,\ \forall i,j$.
18: Compute $m = |R| + \beta_{d'}(D) = \left(\sum_{\forall i,j} \mathcal{X}[i,j] - 1\right)$, where $\mathcal{X}[N+1] = \left(1 - \sum_{i=0}^{N} \mathcal{X}[i]\right)$ and $\beta_{d'}(D) = \sum_{\forall j} \mathcal{X}[N+1][j]$, as $\mathcal{X}[i]$ is the binary fraction denoting selection of numbers from $\mathcal{W}$ for fluid A^i.
19: Compute M_{lb}.
20: Keep the optimal solution(s) depending on the optimality criteria.
21: **end for**
22: Keep the binary matrix $\mathcal{X}$ for any one of the optimal solution(s).
23: Construct the dilution/mixing tree $\mathcal{T}$ from $\mathcal{X}$ using *MinMix* [Thies et al. 2008].

The total number of $(1:1)$ mix-split steps (T_{ms}) in the dilution/mixing tree for a valid solution (binary matrix) can be computed as $T_{ms} = \left(\sum_i \sum_j \mathcal{X}[i,j] - 1\right)$. However, multiple valid solutions (binary matrices) can be obtained. Depending on the optimality criteria (minimization of T_{ms}, d', M_{lb}, or $I_{>1}$) several solutions (binary matrices) may be obtained. For an optimization problem, if we get multiple solutions (binary matrices) with the same optimum value of the parameter, we refer to the situation as a tie. Any one of them can be used to construct a dilution/mixing tree by arbitrarily breaking the ties. The proposed algorithm *EGDA* determines the dilution/mixing tree for generalized dilution. Here, we use *MinMix* [Thies et al. 2008] in Step 23 to construct the dilution/mixing tree from the binary fractions obtained in Step 22 of Algorithm 2. However, other algorithms such as *RMA* [Roy et al. 2011b], *RSM* [Hsieh et al. 2012], *MTCS* [Kumar et al. 2013] or *CoDOS* [Liu et al. 2013] may also be used to determine the tree from the target ratio.

For the example $\mathcal{W}$ of Figure 12(a), four different selections of R from Q and the corresponding binary matrices are shown in Figure 14. No tree can be constructed from the binary matrix of Figure 14(b), indicating an invalid solution. The binary matrices

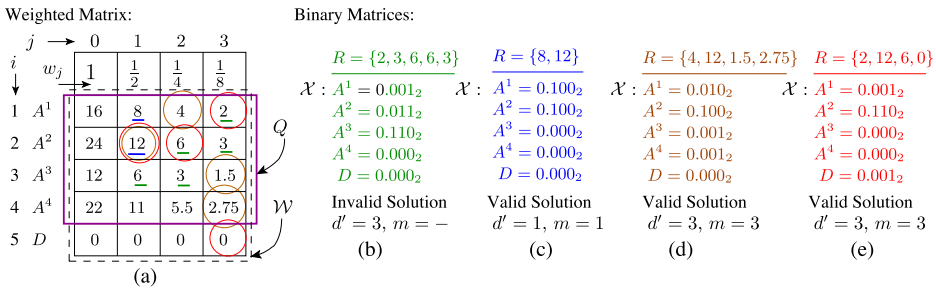


Fig. 14. (a) Weighted matrix shown in Figure 12(b). (b)–(e) Different binary matrices obtained by finding R from Q for $C_A = \frac{20}{32}$.

of Figure 14(c) and 14(d) yield the trees as shown in Figure 12(d) and (e), respectively. The binary matrix of Figure 14(e) provides a suboptimal dilution tree with $d' = 3$ and $T_{ms} = 3$.

4.2.3. Experimental results. We have simulated the proposed algorithm for generalized dilution considering the target *CF* C_A within the range between 0 and $\max_i\{b_i\}$, $\forall i$ ($i =$ 1 to N). The number of input *CF*s (expect buffer D) is varied from 3 to 6 and the input *CF*s are represented as $b_i = \frac{B_i}{2^d}$ with different d (4 and 6). Hence, C_A is represented as $C_A = \frac{T}{2^d}$. Let d' be the height of the dilution/mixing tree.

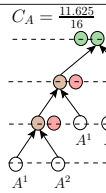
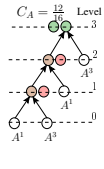
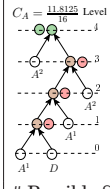
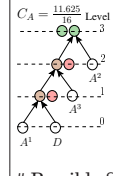
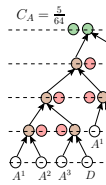
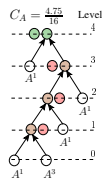
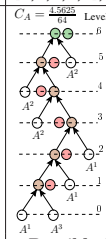
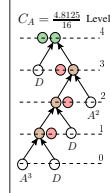
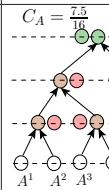
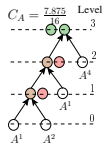
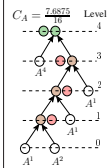
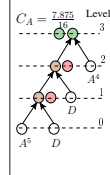
In Table VI, we present the comparative results for three examples of generalized dilution of a fluid A determined by *EGDA* with different optimization criteria, such as minimizing d', T_{ms}, M_{lb}, and $I_{>1}$, respectively. For each example, the total number of solutions and one such dilution/mixing tree obtained by *EGDA* along with its performance parameters are provided in the fifth, sixth, seventh and eighth column of Table VI, when the optimization criterion is to minimize d', T_{ms}, M_{lb} or $I_{>1}$, respectively. For Ex. 1, when the height of the dilution/mixing tree (d') is minimized, *EGDA* provides 6 solutions (dilution/mixing trees), whereas, we obtain two solutions when the total number of (1 : 1) mix-split steps (T_{ms}) is minimized. Similarly, when M_{lb} is minimized, *EGDA* provides 13 solutions (dilution/mixing trees), whereas, if $I_{>1}$ is minimized, we obtain one solution.

5. CONCLUSIONS

In this article, we have provided several new theoretical results and the first algorithmic solutions to the generalized mixing and dilution problems, where the input fluids may be available with random concentration factors. Given such an arbitrary input stock, it was a challenge earlier to optimize the various parameters while preparing a sample on a digital microfluidic biochip. The proposed techniques may be used to address some of these concerns.

We have presented a reachability analysis that characterizes under what input conditions, a given target mixture of several fluids is possible to prepare using a (1 : 1) microfluidic mixer. Our approach to generalized mixing can be used to control the mixing time, cost of reactants, accuracy in the target concentration, and also helps in stock planning of biochemical fluids needed in an assay. We have also provided solutions and experimental results to two variants of the generalized dilution problem, when two or more random *CF*s of the same sample are available. These results will be useful for on-chip implementation of a wide variety of sample preparation procedures, and provide a mechanism to recycle the waste droplets. They also settle several open questions

Table VI. Dilution/mixing Trees Obtained by *EGDA* with Different Optimization Criteria for Some Examples

Ex.	Input *CF*s; $\langle b_1, b_2, \ldots, b_N, 0\rangle$	C_A	d	Tree with Optimal d'; $I[A^1,\ldots,A^{N+1},tot]$; $\langle d', T_{ms}, W, I_{>1}, M_{lb}\rangle$	Tree with Optimal T_{ms}; $I[A^1,\ldots,A^{N+1},tot]$; $\langle d', T_{ms}, W, I_{>1}, M_{lb}\rangle$	Tree with Optimal M_{lb}; $I[A^1,\ldots,A^{N+1},tot]$; $\langle d', T_{ms}, W, I_{>1}, M_{lb}\rangle$	Tree with Optimal $I_{>1}$; $I[A^1,\ldots,A^{N+1},tot]$; $\langle d', T_{ms}, W, I_{>1}, M_{lb}\rangle$
Ex.1	$\langle \frac{7}{16}, \frac{14}{16}, \frac{15}{16}, \frac{0}{16}\rangle$	$\frac{12}{16}$	4	 # Possible Soln. = 6 [2,2,1,0,5]; ⟨3,4,3,2,2⟩	 # Possible Soln. = 2 [2,0,2,0,4]; ⟨3,3,2,2,1⟩	 # Possible Soln. = 13 [2,2,0,1,5]; ⟨4,4,3,2,1⟩	 # Possible Soln. = 1 [1,1,1,1,4]; ⟨3,3,2,0,1⟩
Ex.2	$\langle \frac{1}{64}, \frac{4}{64}, \frac{61}{64}, \frac{0}{64}\rangle$	$\frac{5}{64}$	6	 # Possible Soln. = 31 [3,2,1,2,8]; ⟨4,7,6,3,2⟩	 # Possible Soln. = 18 [4,0,1,0,5]; ⟨4,4,3,3,1⟩	 # Possible Soln. = 198 [3,3,1,0,7]; ⟨6,6,5,4,1⟩	 # Possible Soln. = 8 [0,1,1,3,5]; ⟨4,4,3,0,1⟩
Ex.3	$\langle \frac{3}{16}, \frac{6}{16}, \frac{9}{16}, \frac{12}{16}, \frac{15}{16}, \frac{0}{16}\rangle$	$\frac{8}{16}$	4	 # Possible Soln. = 70 [2,1,1,1,1,0,6]; ⟨3,5,4,1,2⟩	 # Possible Soln. = 26 [2,1,0,1,0,0,4]; ⟨3,3,2,1,1⟩	 # Possible Soln. = 159 [3,1,0,1,0,0,5]; ⟨4,4,3,2,1⟩	 # Possible Soln. =79 [0,0,0,1,1,2,4]; ⟨3,3,2,0,1⟩

in droplet-based algorithmic microfluidics raised earlier in the literature [Thies et al. 2008].

This study also opens up several interesting new directions, for instance, to formulate an improved method to handle three or more related inputs, to extend the algorithms for a general type of $(k : \ell)$ mixer modules (where $k, \ell \geq 2$, $k \neq \ell$), and to design a suitable biochip architecture to support them for expediting automatic sample preparation.

APPENDICES

A.1. Example Target Ratios for Reachability Study

In Table VII, we present results on target mixture reachability for some examples. The input stock is shown in column 2, target ratio and accuracy level d in column 3; the scaled *CF*s in binary fraction are shown in column 4. In column 5, we show whether or not the reachability conditions are satisfied; column 6 gives a pointer to the corresponding mixing tree if the target is reachable, and reports the number of $(1 : 1)$ mix-split steps T_{ms} needed to produce the target mixture.

For example, consider that we have eight fluids x_is (for $i = 1$ to 8), supplied with a *CF* of 100% each. We need to prepare a target mixture with a ratio $\{c_1 : c_2 : c_3 : c_4 : c_5 :$

Table VII.

Results for some example target ratios, that is, reachability, accuracy (d), and the total mumber of (1 : 1) mix-split steps (T_{ms}) (if the target is reachable)

Example	Input CFs $b_1, \ldots, b_N, 0$	Target mixture $\langle c_1 : \ldots : c_N : c_{N+1} \rangle^*$ (accuracy level, d)	Scaled c'_is $\left(c'_i = \frac{c_i}{b_i}\right)$ (binary fraction)	Reachability of target mixture [condition satisfied]	Mixing tree (d and T_{ms}) (if reachable)
Ex.1	40% x_1 and 56% x_2 (Each diluted with D) ($N = 2$)	$\langle 41\%\ x_1, 56\%\ x_2,$ and $3\%\ D \rangle$ ($d = 7$)	—	Not Reachable $\because 0.41 > (0.4 + \frac{1}{2^7})$ [by Case (i)]	—
		$\langle 40.5\%\ x_1,$ and $59.5\%\ D \rangle$ ($d = 7$)	—	Reachable $\because 0.4 < 0.405 < (0.4 + \frac{1}{2^7})$ [by Case (ii)]	40% x_1 itself is the target mixture.
		$\langle 40.5\%\ x_1,$ $56.5\%\ x_2$ and $3\%\ D \rangle$ ($d = 7$)	—	Not Reachable $\because 0.4 < 0.405 < (0.4 + \frac{1}{2^7})$ and $0.56 < 0.565 < (0.56 + \frac{1}{2^7})$ [by Case (iii)]	—
		$\langle 40.5\%\ x_1, 53\%\ x_2$ and $6.5\%\ D \rangle$ ($d = 7$)	$\mathcal{B}[c'_1] = 1.00000011_2$ $\mathcal{B}[c'_2] = 0.111100101_2$	Not Reachable $\because \frac{c_1}{b_1} + \frac{c_1}{b_2} = 1.9589 > 1.0$ and $\mathcal{B}[c'_1] + \mathcal{B}[c'_2] = 1.1111_2 > 1.0$ [by Case (iv)]	—
Ex.2	80% x_1, 70% x_2 and 45% x_3 (Each diluted with D) ($N = 3$)	$\langle 20\%\ x_1,$ $30\%\ x_2,$ $41\%\ x_3$ and $9\%\ D \rangle$ (any $d > 1$)	$\mathcal{B}[c'_1] = 0.0100_2$ $\mathcal{B}[c'_2] = 0.0110_2$ $\mathcal{B}[c'_3] = 0.1110_2$	Not Reachable $\sum_{i=1}^{N} \frac{c_i}{b_i} > 1.0$ and $\sum_{i=1}^{N} \mathcal{B}[c'_i] = 1.1_2 > 1.0$ [by Theorem 3.1]	—
Ex.3	80% A, 70% B and 45% C (Each diluted with D) ($N = 3$)	$\langle 20\%\ A,$ $30\%\ B,$ $9\%\ C$ and $41\%\ D \rangle$ ($d = 4$)	$\mathcal{B}[c'_1] = 0.0100_2$ $\mathcal{B}[c'_2] = 0.0110_2$ $\mathcal{B}[c'_3] = 0.0011_2$	Reachable $\sum_{i=1}^{N} \frac{c_i}{b_i} < 1.0$ and $\sum_{i=1}^{N} \mathcal{B}[c'_i] = 0.1101_2 < 1.0$ $\mathcal{B}[D] = 0.0011_2$ [Excess D] [by Theorem 3.1]	Figure 4 $d = 4$ and $T_{ms} = 6$
Ex.4	100% x_is (for $i = 1$ to 8) ($N = 8$)	$\langle c_1 = c_2 = 11.5\%,$ $c_3 = c_4 = 11.5\%,$ $c_5 = c_6 = 13.5\%,$ $c_7 = c_8 = 13.5\% \rangle$ ($d = 7$)	$\mathcal{B}[c'_i] = 0.0001110_2$ (for $i = 1$ to 4) $\mathcal{B}[c'_i] = 0.0010001_2$ (for $i = 5$ to 8)	Reachable $\sum_{i=1}^{N} \frac{c_i}{b_i} = 1.0$ and $\sum_{i=1}^{N} \mathcal{B}[c'_i] = 0.1111100_2 < 1.0$ $\mathcal{B}[D] = 0.0000100_2$ [Excess D] [by Theorem 3.1]	Figure 15(a) $d = 7$ and $T_{ms} = 20$
Ex.5	100% x_is (for $i = 1$ to 3) ($N = 3$)	$\langle 25\%\ x_1,$ $12.5\%\ x_2$ and $62.5\%\ x_3 \rangle$ ($d = 3$)	$\mathcal{B}[c'_1] = 0.010_2$ $\mathcal{B}[c'_2] = 0.001_2$ $\mathcal{B}[c'_3] = 0.101_2$	Reachable $\sum_{i=1}^{N} \frac{c_i}{b_i} = 1.0$ and $\sum_{i=1}^{N} \mathcal{B}[c'_i] = 1.0$ [No Excess D] [by Theorem 3.1]	Figure 15(b) $d = 3$ and $T_{ms} = 3$
Ex.6	100% x_is (for $i = 1$ to 3) ($N = 3$)	$\langle 71.8\%\ x_1,$ $3.1\%\ x_2$ and $25\%\ x_3 \rangle$ ($d = 5$)	$\mathcal{B}[c'_1] = 0.10111_2$ $\mathcal{B}[c'_2] = 0.00001_2$ $\mathcal{B}[c'_3] = 0.01000_2$	Reachable $\sum_{i=1}^{N} \frac{c_i}{b_i} < 1.0$ and $\sum_{i=1}^{N} \mathcal{B}[c'_i] = 1.0$ [No Excess D] [by Theorem 3.1]	Figure 15(c) $d = 5$ and $T_{ms} = 5$

Table VII. Continued.

Example	Input *CF*s $b_1, \ldots, b_N, 0$	Target mixture $\langle c_1 : \ldots : c_N : c_{N+1} \rangle^*$ (accuracy level, d)	Scaled c'_is $\left(c'_i = \frac{c_i}{b_i}\right)$ (binary fraction)	Reachability of target mixture [condition satisfied]	Mixing tree (d and T_{ms}) (if reachable)
Ex.7	80% A, 70% B and 45% C (Each diluted with D) ($N = 3$)	$\langle$20% A, 30% B, 14.5% C and 35.5% $D\rangle$ ($d = 4$)	$\mathcal{B}[c'_1] = 0.0100_2$ $\mathcal{B}[c'_2] = 0.0110_2$ $\mathcal{B}[c'_3] = 0.0101_2$	Reachable $\sum_{i=1}^{N} \frac{c_i}{b_i} = 1.0008 > 1.0$ and $\sum_{i=1}^{N} \mathcal{B}[c'_i] = 0.1111_2 < 1.0$ $\mathcal{B}[D] = 0.0001_2$ [Excess D] [by Theorem 3.1]	Figure 5 $d = 4$ and $T_{ms} = 5$

* c_{N+1} is the *CF* of D in target mixture, that is, $c_{N+1} = \left[1.0 - \sum_{i=1}^{N} c_i\right]$.

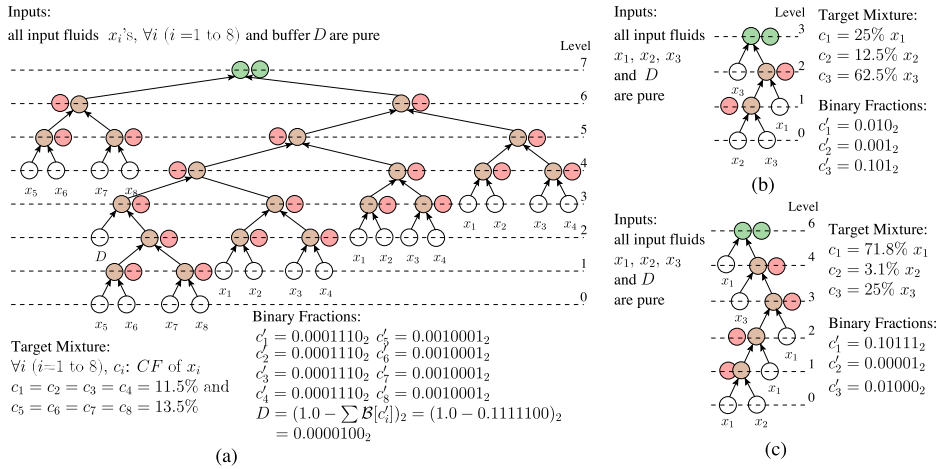


Fig. 15. Mixing tree obtained by *GMA* for a target mixture from a supply of input fluids. Examples: (a) where, additional buffer solution D is required, and (b)–(c), where buffer solution is not required.

$c_6 : c_7 : c_8\}$ such that $c_1 = c_2 = c_3 = c_4 = 11.5\%$ and $c_5 = c_6 = c_7 = c_8 = 13.5\%$. The corresponding mixing tree for $d = 7$ as obtained by *GMA* is shown in Figure 15(a). Note that it requires one additional buffer droplet (D), since $\mathcal{B}[D] = 0.0000100_2$. Details of this example are shown as Ex. 4 in Table VII. For another example (Ex. 5 in Table VII), we assume that three input fluids x_1, x_2, x_3 are supplied as 100% each, and a target mixture with the ratio 25% x_1 : 12.5% x_2 : 62.5% x_3, is to be produced. In this case, $\mathcal{B}[c'_1] = 0.010_2$, $\mathcal{B}[c'_2] = 0.001_2$ and $\mathcal{B}[c'_3] = 0.101_2$; hence $\sum_{i=1}^{N} \frac{c_i}{b_i} = 1.0$ and $\sum_{i=1}^{N} \mathcal{B}[c'_i] = 1.0$. Figure 15(b) shows the corresponding mixing tree obtained by *GMA*. Note that it does not require any additional buffer droplet. Figure 15(c) shows another example (Ex. 6 in Table VII) of preparing a target mixture of 71.8% x_1 : 3.1% x_2 : 25% x_3, for which the mixing tree is shown in Figure 15(c).

A.2. Pseudocode of *GDA*

The pseudocode for the *GDA* is written as Algorithm 3.

ALGORITHM 3: *GDA* (C_ℓ, C_h, C_t, d)

1: Compute $C'_t = \dfrac{C_t - C_\ell}{C_h - C_\ell}$.
2: Express C'_t as a d-bit binary fraction $\mathcal{B}[C'_t] = \{0.s_1 s_2 \ldots s_{i(d-1)} s_d\}$ (rounded off after looking at the $(d+1)^{th}$ bit), where $s_i \in \{0, 1\}$.
3: Set $i = 1$ and $Z_d = \Phi$.
4: Ignore all the 0-bits from the right of $\mathcal{B}[C'_t]$ till the first 1-bit by every time incrementing i by 1.
5: List the first mix-split step into the sequence, i.e., $Z_d = Z_d \bigcup \{X, mix\_split(C_\ell, C_h)\}$..
6: Obtain first intermediate *CF* as $X = \frac{C_\ell + C_h}{2}$ and set $i = i + 1$.
7: **while** $i \leq d$ **do**
8: **if** in $\mathcal{B}[C'_t]$ $s_i = 1$ **then**
9: List next mix-split cycle into the sequence, i.e., $Z_d = Z_d \bigcup \{X, mix\_split(X, C_h)\}$.
10: Obtain next intermediate *CF* as $X = \frac{X + C_h}{2}$ and set $i = i + 1$.
11: **else**
12: List next mix-split cycle into the sequence, i.e., $Z_d = Z_d \bigcup \{X, mix\_split(X, C_\ell)\}$.
13: Obtain next intermediate *CF* as $X = \frac{X + C_\ell}{2}$ and set $i = i + 1$.
14: **end if**
15: **end while**
16: Construct the dilution tree $\mathcal{T}$ from the sequence Z_d of $(1:1)$ mix-split steps.

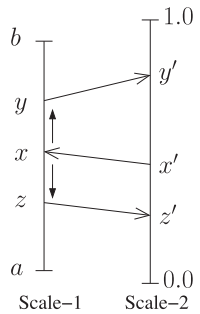


Fig. 16. Mapping of *CF*s between two different scales.

A.3. Proof of Theorem 4.1

PROOF. We assume that the highest (lowest) *CF* in Scale-1 is C_h (C_ℓ), whereas, in Scale-2 it is 1.0 (0.0). In order to run *GDA*, a mapping of *CF* C_t is required from Scale-1 (C_ℓ to C_h) to C'_t in Scale-2 (0.0 to 1.0) (see Figure 16). Since, $C'_t = \dfrac{C_t - C_\ell}{C_h - C_\ell}$, it follows that $C_t = (C_h - C_\ell)C'_t + C_\ell$. Now, if $C_t = C_\ell$, then $C'_t = 0.0$, and if $C_t = C_h$, then $C'_t = 1.0$. Thus, the boundary conditions of Scale-1 are satisfied at the two extremities of Scale-2.

To prove the correctness of *GDA*, we have to show that the mapping is valid after execution of arbitrary number of mix-split steps during the dilution process.

Assume that, at any instant, a *CF* of x in Scale-1 is mapped x' in Scale-2. After mixing it with the highest *CF*, we have to show that the new *CF* in Scale-2, that is, $y' = \dfrac{x' + 1.0}{2}$ is also the mapping of the new *CF* in Scale-1, that is, $y = \dfrac{x + C_h}{2}$. So, $x = (C_h - C_\ell)x' + C_\ell$ and after mixing it with C_h, we will move to the point $y = \dfrac{x + C_h}{2} = \dfrac{1}{2}\Big(((C_h - C_\ell)x' + C_\ell) + C_h\Big)$. Hence, $y' = \dfrac{y - C_\ell}{C_h - C_\ell}$. By substituting the expression for y in this equation and after simplification we can get $y' = \dfrac{x' + 1.0}{2}$, which

is the resulting *CF* obtained by mixing x' with C_h. Thus, the new *CF*s in the two scales obtained by mixing droplet of higher *CF* with the current one are equivalent.

Again assume that, at any arbitrary instant, x in Scale-1 is mapped to x' in Scale-2; after mixing it with the lowest *CF*, we have to show that the new *CF* in Scale-2, that is, $z' = \frac{x' + 0.0}{2}$ is also the mapping of the new *CF* in Scale-1, that is, $z = \frac{x + C_\ell}{2}$. So, $x = (C_h - C_\ell)x' + C_\ell$ and after mixing it with C_ℓ we get the new *CF* as $z = \frac{x + C_\ell}{2} = \frac{1}{2}(C_h - C_\ell)x' + C_\ell$. Hence, $z' = \frac{z - C_\ell}{C_h - C_\ell}$, and by substituting the expression for z in it and after simplification we get $z' = \frac{x' + 0.0}{2}$, which is the resulting *CF* obtained by mixing x' with C_ℓ. Thus, the new *CF*s in the two scales obtained by mixing the lower *CF* with the current one are equivalent.

Hence, after executing d $(1 : 1)$ mix-split steps, *GDA* correctly produces the target *CF* C_t from an input supply of two *CF*s C_h and C_ℓ. It is also easy to observe that scaling preserves the error bound as well. □

REFERENCES

M. Abdelgawad and A. R. Wheeler. 2009. The digital revolution: A new paradigm for microfluidics. *Adv. Mater. 21*, 920–925.

S. Bhattacharjee, A. Banerjee, and B. B. Bhattacharya. 2012. Multiple dilution sample preparation using digital microfluidic biochips. In *Proceedings of the International Symposium on Electronic System Design*. 188–192.

S. Bhattacharjee, A. Banerjee, and B. B. Bhattacharya. 2014. Sample preparation with multiple dilutions on digital microfluidic biochips. *IET Comput. Digital Tech. 8*, 1, 49–58.

S. Bhattacharjee, A. Banerjee, T.-Y. Ho, K. Chakrabarty, and B. B. Bhattacharya. 2013. Algorithms for producing linear dilution gradient with digital microfluidics. *CoRR* abs/1307.1251.

K. Chakrabarty and T. Xu. 2010. *Digital Microfluidic Biochips: Design and Optimization*.

D. Chatterjee, B. Hetayothin, A. R. Wheeler, D. J. King, and R. L. Garrell. 2006. Droplet-based microfluidics with nonaqueous solvents and solutions. *Lab-on-a-Chip 6*, 199–206.

T.-W. Chiang, C.-H. Liu, and J.-D. Huang. 2013. Graph-based optimal reactant minimization for sample preparation on digital microfluidic biochips. In *Proceedings of the International Symposium on VLSI Design, Automation, and Test*. 1–4.

S. K. Cho, H. M., and C.-J. Kim. 2003. Creating, transporting, cutting, and merging liquid droplets by electrowetting-based actuation for digital microfluidic circuits. *J. Microelectromechanical Syst. 12*, 1, 70–80.

N. J. Cira, J. Y. Ho, M. E. Dueck, and D. B. Weibel. 2012. A self-loading microfluidic device for determining the minimum inhibitory concentration of antibiotics. *Lab-on-a-Chip 12*, 6, 1052–1059.

T. A. Dinh, S. Yamashita, and T.-Y. Ho. 2014. A network-flow-based optimal sample preparation algorithm for digital microfluidic biochips. In *Proceedings of the Asia and South Pacific Design Automation Conference*. 225–230.

R. B. Fair. 2007. Digital microfluidics: Is a true lab-on-a-chip possible? *Microfluid. Nanofluid. 3*, 245–281.

R. B. Fair, A. Khlystov, T. D. Tailor, V. Ivanov, R. D. Evans, P. B. Griffin, V. Srinivasan, V. K. Pamula, M. G. Pollack, and J. Zhou. 2007. Chemical and biological applications of digital-microfluidic devices. *IEEE Des. Test Comput. 24*, 1, 10–24.

Y. Fouillet, D. Jary, C. Chabrol, P. Claustre, and C. Peponnet. 2008. Digital microfluidic design and optimization of classic and new fluidic functions for lab on a chip systems. *Microfluid. Nanofluid. 4*, 3, 159–165.

M. R. Garey and D. S. Johnson. 1979. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company.

E. J. Griffith, S. Akella, and M. K. Goldberg. 2006. Performance characterization of a reconfigurable planar-array digital microfluidic system. *IEEE Trans. CAD 25*, 2, 345–357.

K. E. Herold and A. Rasooly. 2009. *Lab-on-a-Chip Technology (Vol. 1): Fabrication and Microfluidics*. Caister Academic Press.

Y.-L. Hsieh, T.-Y. Ho, and K. Chakrabarty. 2012. A reagent-saving mixing algorithm for preparing multiple-target biochemical samples using digital microfluidics. *IEEE Trans. CAD 31*, 11, 1656–1669.

Y.-L. Hsieh, T.-Y. Ho, and K. Chakrabarty. 2014. Biochip synthesis and dynamic error recovery for sample preparation using digital microfluidics. *IEEE Trans. CAD 33*, 2, 183–196.

J.-D. Huang, C.-H. Liu, and T.-W. Chiang. 2012. Reactant minimization during sample preparation on digital microfluidic biochips using skewed mixing trees. In *Proceedings of the IEEE International Conference on Computer-Aided Design*. 377–384.

J.-D. Huang, C.-H. Liu, and H.-S. Lin. 2013. Reactant and waste minimization in multitarget sample preparation on digital microfluidic biochips. *IEEE Trans. CAD 32*, 10, 1484–1494.

T.-W. Huang and T.-Y. Ho. 2011. A two-stage ILP-based droplet routing algorithm for pin-constrained DMF biochips. *IEEE Trans. CAD 30*, 2, 215–228.

T.-W. Huang, C.-H. Lin, and T.-Y. Ho. 2010. A contamination aware droplet routing algorithm for the synthesis of digital microfluidic biochips. *IEEE Trans. CAD 19*, 11, 1682–1695.

M. J. Jebrail and A. R. Wheeler. 2009. Digital microfluidic method for protein extraction by precipitation. *Anal. Chim. Acta 81*, 1, 330–335.

J. Kleinberg and E. Tardos. 2005. *Algorithm Design*. Addison-Wesley.

S. Kumar, S. Roy, P. P. Chakrabarti, B. B. Bhattacharya, and K. Chakrabarty. 2013. Efficient mixture preparation on digital microfluidic biochips. In *Proceedings of the IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*. 205–210.

K. Lee, C. Kim, B. Ahn, R. Panchapakesan, A. R. Full, L. Nordee, J. Y. Kang, and K. W. Oh. 2009. Generalized serial dilution module for monotonic and arbitrary microfluidic gradient generators. *Lab-on-a-Chip 9*, 709–717.

C. C.-Y. Lin and Y.-W. Chang. 2011. Cross-contamination aware design methodology for pin-constrained digital microfluidic biochips. *IEEE Trans. CAD 30*, 6, 817–828.

C.-H. Liu, H.-H. Chang, T.-C. Liang, and J.-D. Huang. 2013. Sample preparation for many-reactant bioassay on dmfbs using common dilution operation sharing. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 615–621.

V. N. Luk and A. R. Wheeler. 2009. A Digital Microfluidic Approach to Proteomic Sample Processing. *Analytica Chimica Acta 81*, 11, 4524–4530.

Y. Luo, K. Chakrabarty, and T.-Y. Ho. 2013. Error recovery in cyberphysical digital microfluidic biochips. *IEEE Trans. CAD 32*, 1, 59–72.

Y. Luo, T.-Y. Ho, and K. Chakrabarty. 2012. Dictionary-based error recovery in cyberphysical digital-microfluidic biochips. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*. 369–376.

E. Maftei, P. Pop, and J. Madsen. 2012. Routing-based synthesis of digital microfluidic biochips. *Des. Automation Embed. Syst. 16*, 1, 19–44.

E.M. Miller and A. R.Wheeler. 2009. Digital bioanalysis. *Anal. Bioanal. Chem. 393*, 2, 419–426.

D. Mitra, S. Roy, K. Chakrabarty, and B. B. Bhattacharya. 2012. On-chip sample preparation with multiple dilutions using digital microfluidics. In *Proceedings of the IEEE International Symposium on VLSI*. 314–319.

H. Ren, V. Srinivasan, and R. B. Fair. 2003. Design and testing of an interpolating mixing architecture for electrowetting-based droplet-on-chip chemical dilution. In *Proceedings of the International Conference on Solid-State Sensors, Actuators and Microsystems*. 619–622.

S. Roy, B. B. Bhattacharya, P. P. Chakrabarti, and K. Chakrabarty. 2011b. Layout-aware solution preparation for biochemical analysis on a digital microfluidic biochip. In *Proceedings of the International Conference on VLSI Design*. 171–176.

S. Roy, B. B. Bhattacharya, and K. Chakrabarty. 2010. Optimization of dilution and mixing of biochemical samples using digital microfluidic biochips. *IEEE Trans. CAD 29*, 11, 1696–1708.

S. Roy, B. B. Bhattacharya, and K. Chakrabarty. 2011a. Waste-aware dilution and mixing of biochemical samples with digital microfluidic biochips. In *Proceedings of the Conference and Exhibition on Design, Automation and Test in Europe*. 1059–1064.

S. Roy, B. B. Bhattacharya, S. Ghoshal, and K. Chakrabarty. 2013a. High-throughput dilution engine for sample preparation on digital microfluidic biochips. *IET Comput. Digital Tech.*

S. Roy, B. B. Bhattacharya, S. Ghoshal, and K. Chakrabarty. 2013b. On-chip dilution from multiple concentrations of a sample fluid using digital microfluidics. In *Proceedings of the International Symposium on VLSI Design and Test*. 1–9.

S. Roy, P. P. Chakrabarti, S. Kumar, B. B. Bhattacharya, and K. Chakrabarty. 2013c. Routing-aware resource allocation for mixture preparation in digital microfluidic biochips. In *Proceedings of the International Symposium on VLSI*. 1–6.

R. Sista, Z. Hua, P. Thwar, A. Sudarsan, V. Srinivasan, A. E. Eckhardt, M. G. Pollack, and V. K. Pamula. 2008. Development of a digital microfluidic platform for point of care testing. *Lab-on-a-Chip 8*, 12, 2091–2104.

S. Sugiura, K. Hattori, and T. Kanamori. 2010. Microfluidic serial dilution cell-based assay for analyzing drug dose response over a wide concentration range. *Anal. Chem. 82*, 19, 8278–8282.

J. Talsma. 2013. Drug Shortages Still at Crisis Levels. http://drugtopics.modernmedicine.com/drug-topics/news/drug-shortages-still-crisis-levels?page=full.

H. Y. Tan, W. K. Loke, Y. T. Tan, and N.-T. Nguyen. 2008. A lab-on-a-chip for detection of nerve agent sarin in blood. *Lab-on-a-Chip 8*, 6, 885–891.

W. Thies, J. P. Urbanski, T. Thorsen, and S. Amarasinghe. 2008. Abstraction layers for scalable microfluidic biocomputing. *Natural Computing 7*, 2, 255–275.

C. L. Ventola. 2011. The drug shortage crisis in the United States: Causes, impact, and management strategies. *Pharm. Therapeutics 36*, 11, 740–742, 749–757.

H. A. Yusuf, S. J. Baldock, P. R. Fielden, N. J. Goddard, S. Mohr, and B. J. T. Brown. 2009. Systematic linearisation of a microfluidic gradient network with unequal solution inlet viscosities demonstrated using glycerol. *Microfluid. Nanofluid. 8*, 5, 587–598.

Received November 2013; revised March 2014; accepted April 2014