UNIVERSITY OF CALIFORNIA
RIVERSIDE

Towards a Theory of Droplet-Mixing Graphs in Microfluidics

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Miguel Coviello Gonzalez

September 2019

Dissertation Committee:

Professor Marek Chrobak, Chairperson
Professor Stefano Lonardi
Professor Philip Brisk
Professor William Grover

The Dissertation of Miguel Coviello Gonzalez is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I sincerely thank CONACYT and UC MEXUS for their full-ride scholarship during my PhD studies. Without their support, this dissertation would not have been possible.

I also thank the National Science Foundation for their funding to the project *AitF: EXPL: Algorithmic Fluid Concentration Management for Programmable Microfluidics*, with award number *CCF-1536026*, which this dissertation is part of.

In addition, I express my deepest appreciation to my advisor and committee chair Marek Chrobak, for his guidance and assistance throughout the entire PhD program.

Finally, I also thank the rest of my dissertation committee members, Philip Brisk, Stefano Lonardi and William Grover, for their great support and invaluable advice.

To God, for giving me life, health and intelligence to achieve my goals.

To my parents, Francisco Jose Coviello Marcano and Maricela Gonzalez Tolosa, for all the sacrifices they made to make me who I am.

To my wife, Nadia Lorena Valdez Guevara, for supporting me thoroughly during this long and challenging phase.

To my friends, family and everybody else that believed in me. Specially, my brothers Jonatan Tadeo Coviello Gonzalez and Marcelo Coviello Gonzalez.

Thank you!

ABSTRACT OF THE DISSERTATION

Towards a Theory of Droplet-Mixing Graphs in Microfluidics

by

Miguel Coviello Gonzalez

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, September 2019
Professor Marek Chrobak, Chairperson

In this work, we study the problem of fluid mixing in microfluidic chips. The motivation for studying this problem comes from the process of sample preparation for chemical, biological, medical and environmental experiments, which often require preparation of fluid mixtures with desired concentrations.

We assume that fluids are manipulated in discrete units called droplets. The input set of droplets consists of two distinct fluids: the reactant, which is the fluid of interest, and the buffer fluid that is used to dilute it. The goal is to produce a target set of droplets with prespecified reactant concentrations. In the model we study, the mixing process in a microfluidic chip can be abstractly represented as a mixing graph.

A mixing graph is a collection of micro-mixers (nodes) connected by micro-channels (edges) that converts an input set of droplets I into a set of output droplets T, by applying a sequence of 1:1 mixing operations. This graph may also produce some waste, which are superfluous droplets of fluid not used in the target set. Computational complexity of most natural questions regarding such mixing graphs remain open. For example, it is not even known whether it is decidable for a given target set to be produced without waste. Current work in the literature contains only heuristic approaches that compute mixing graphs while attempting to optimize certain objectives, including minimizing waste, reactant usage, the depth of the graphs, and more.

Our first contribution is an efficient algorithm for computing mixing graphs for single-droplet targets. Our algorithm produces significantly less waste than state-of-the-art algorithms in an experimental comparison. We also provide a bound on its worst-case performance that is significantly better than those for earlier algorithms. Our second result concerns the variant of the problem where the objective is to design a mixing graph that perfectly mixes a collection of input droplets with arbitrary concentrations. We provide a complete characterization of input sets for which such graphs exist, and an efficient algorithm to construct these graphs. In addition, we provide several other results about properties of mixing graphs and about computational complexity of computing mixing graphs of fixed depth.

Contents

List of Figures

Chapter 1

Introduction

1.1 Microfluidic Mixing Chips

Microfluidics is an area of science and engineering dedicated to the study and manipulation of very small (picoliter to nanoliter [24]) amounts of fluids. Microfludic devices have found applications in inkjet printing [4, 10, 41], microlens arrays [13, 18], super-resolution imaging [39], bioanalysis [57], and cell biology research [29]. One of the most remarkable achievements of microfluidics is the creation of *lab-on-chip* (LoC), which can replace bulky and costly biochemical systems in labs, hospitals, and research centers [25]. Liu *et al.* in [38] defined LoC as a tiny device integrating a variety of laboratory functions, such as dispensing, mixing, filtering, separating and detecting. During the past years, LoC has gained a lot of attention due to its promising applications in cancer discovery [32,37,53], health environment monitoring [40], protein analysis [28, 31, 72], drug discovery [9, 66] and human physiological sample analysis [60, 62]. Furthermore, the importance of LoC devices will soon scale up with the introduction of *cloud laboratories* [19], which give researchers access to state-of-the-art equipment and data analysis tools, allowing them to carry out their experiments remotely.

LoC devices manipulate the fluids along their chambers by applying physical effects such as hydrodynamic pressure gradients, capillarity, magnetic forces, dielectrophoresis, electrophoresis, electrowetting and acoustic waves [58, 63].

One of the most fundamental functions of LoC devices is *mixing* of different fluids, where the objective is to produce desired volumes of pre-specified mixtures of some given fluids. There are several different technologies that can be used to manufacture microfluidic chips for fluid mixing. One of the technologies involves *fluid-based* mixing chips, where continuous volumes of fluids are manipulated throughout the chip. In other fluid mixing technologies, discrete volumes of fluids, where the unit of fluid is called *droplet*, are mixed by tiny components called micro-mixers.

In this work, we study droplet-based mixing chips. One of the motivations for this study is that, although such mixing chips are simple to model, there are still many theoretical gaps regarding the use of algorithms for the computation of efficient mixing sequences for a mixing chip. In the following sections we describe the above mentioned technologies for fluid mixing in LoC mixing devices, and further discuss our adopted model and the problem statement.

1.1.1 Fluid-based Mixing Chips

In fluid-based technologies for mixing chips, continuous flows of fluids are manipulated throughout the chips. This is commonly achieved by having fluid dispensers continuously inject fluids into the chips at a given rate. These fluid dispensers can regulate the rate at which they inject fluids by adjusting the pressure on their pumps. In addition, micro-channels (which fluids flow through) have their own hydrodynamic resistances, which decrease the flow rate of the fluids passing through them.

Since fluids are continuously flowing, the typical approach is to let fluids mix along the way while they are being transported, instead of having micro-mixers mixing the fluids at specific locations. Certain fluid-based mixing chips bear some similarities to electric network circuits. (See for example Appendix A.3.)

Oh *et al.* in [44] classified fluid-based mixing chips as either concentration-dependent or flow-dependent. Later, they further classified concentration-dependent LoC mixing chips into five network categories based on the architecture and similarities of their approaches: proportional, T-typed, serial, combinatorial and pyramidal. (See Appendix A for examples of each type of concentration-dependent fluid-based mixing chip.)

The mathematics behind fluid-based technologies depends on many non-trivial factors, including the physical design of the mixing chip, the physical properties of the fluids, and the physical effects applied to the fluids. Hence, fluid-based technologies are commonly represented by continuous models. However, the focus of this work is on discrete models, so issues that arise in fluid-based technologies are outside the scope of this work.

1.1.2 Droplet-based Mixing Chips

In droplet-based technologies, LoC mixing devices manipulate discrete volumes of fluids called droplets. This is achieved by either having the fluid dispenser system provide the exact volume of fluid per droplet, or by internally dividing the fluids into droplets within the mixing chip. (Since we are interested in purely theoretical analysis, it is safe to assume that these mixing chips are provided with exact units of fluid per droplet.)

There are also different technologies for droplet-based mixing chips. However, in this work we focus on chips that involve a collection of tiny components called *micro-mixers*, which are connected by *micro-channels*. In such chips, input droplets are injected into the chip by a fluid dispenser system, then they travel through appropriate micro-channels, through a sequence of micro-mixers where they are subjected to mixing operations, and are eventually discharged into output reservoirs. These droplet-based mixing chips may also produce a by-product of superfluous fluid, called *waste*, that is discarded.

Typically, droplet-based mixing chips are categorized by *mixing models*. The most common mixing model is the 1:1-model, for its simplicity and because mixing chips of this model are relatively easy to manufacture. In this model, a micro-mixer has exactly two input and two output channels. It receives one droplet from each input, mixes them perfectly, and produces two identical droplets on its outputs.

Other mixing models include the a:a-model and the a:b-model, where a droplets of equal concentration are mixed with some other a or b droplets, respectively, of equal concentration. However, in this work we focus on mixing chips under the 1:1-model only.

1.1.3 Dilution

Dilution is a special case of fluid mixing that involves only two fluids, called *buffer* and *reactant*. The goal is to produce a desired target set of fluid volumes with pre-specified concentrations of reactant by diluting it with buffer. For example, in clinical diagnostics (see [62]), blood, serum, plasma and urine may be used as reactants and phosphate buffered saline as a buffer.

Recall that in a mixing chip under the 1:1-mixing model, a micro-mixer takes two input droplets, mixes them perfectly, and produces two output droplets with equal concentration. Therefore, if the input droplets have (reactant) concentrations a, b, then the produced droplets will have concentration $\frac{1}{2}(a+b)$. It follows that all droplets flowing through the chip have concentrations of the form $c/2^d$, where c and $d \geq 0$ are integers. This simply means that their binary representations are finite; note that these numbers are simply *binary numbers*. In this representation, the number d, called *precision*, is the number of fractional bits; assuming c is odd when $d \geq 1$.

In addition to mixing chips under the 1:1-mixing model, in this work we also focus on dilution. Throughout the rest of the dissertation, we will represent droplets by their reactant concentrations (which uniquely determines their buffer concentration, since both add up to one).

1.1.4 Mixing Graphs

The mixing process in a droplet-based mixing chip can be naturally represented by a directed acyclic graph G that we call a *mixing graph*. The edges of G represent micro-channels. Source vertices represent droplet dispensers, internal vertices represent micro-mixers, and sink vertices represent output reservoirs. In a mixing graph represented by the 1:1-mixing model, source vertices have in-degree 0 and out-degree 1, internal vertices have in-degree and out-degree 2, and sink vertices have in-degree 1 and out-degree 0.

Given a set I of input droplets injected into the source nodes, a mixing graph G converts it into the set Q of droplets in its sink nodes. Graph G_1 in Figure 1.1 illustrates a mixing graph for a 1:1 droplet-based mixing chip that, given input set $I = \{0, 0, 1, 1, 1, 1\}$, converts it into the output set $Q = \{\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{13}{16}, \frac{13}{16}, \frac{7}{8}\}$.

Typically, given an input set I and an output set T of droplets, the goal is to design a mixing graph that converts I into T. However, the computed mixing graph might also produce a superfluous set W of droplets, called *waste*, such that $W = Q - T$, where Q is the output set of droplets. (This is trivially true when I has more droplets than T.) For example, consider $I = \{0, 0, 1, 1, 1, 1\}$ and $T = \{\frac{1}{8}, \frac{1}{8}, \frac{13}{16}, \frac{13}{16}\}$. Then graph G_2 in Figure 1.1 converts I into $T \cup W$, for $W = \{\frac{1}{4}, \frac{7}{8}\}$.

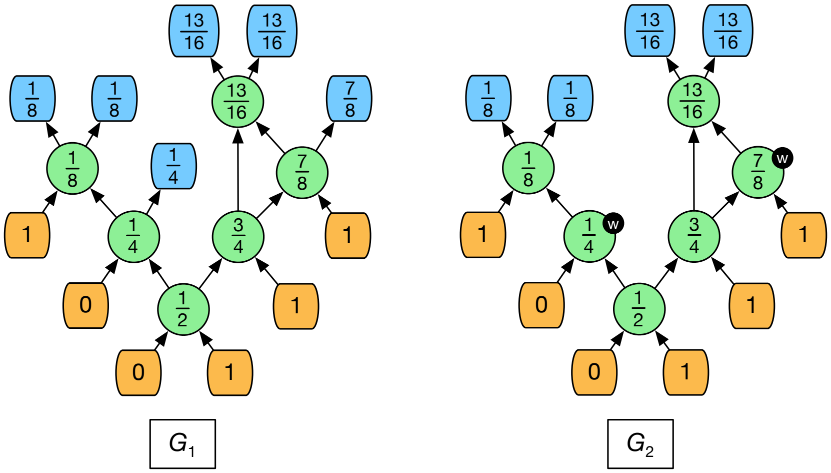

Figure 1.1: Two mixing graph examples for a given input set $I = \{0, 0, 1, 1, 1, 1\}$. Mixing graph G_1 converts the input set I into the droplet set $\{\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{13}{16}, \frac{13}{16}, \frac{7}{8}\}$. Numbers on the micro-mixers (internal nodes) represent the droplet's reactant concentrations produced by these micro-mixers. If only some of the produced droplets are needed, the remaining droplets are designated as waste. This is illustrated by mixing graph G_2, which produces the droplet set $\{\frac{1}{8}, \frac{1}{8}, \frac{13}{16}, \frac{13}{16}\}$. Small black circles labeled "w" on micro-mixers represent droplets of waste.

The most fundamental algorithmic question involving mixing chips asks for designing mixing graphs. Typically, we seek not just any correct graph, but one that minimizes some measure of cost: for example, the reactant volume, waste, or the number of mixing operations.

1.1.5 Problem Statement

In this work, we focus on designing mixing graphs for droplet-based mixing chips under the 1:1-mixing model. Specifically, we focus on mixing graphs that minimize waste. As a starting point, consider the following problem:

MixReachability: Given an input set I and a target set T of droplets, design a mixing graph that converts I into T without waste (if at all possible).

If there is a mixing graph that converts I into T (exactly, without waste) then we say that T is *mix-reachable*, or just *reachable*, from I. For example, let $I = \left\{\frac{1}{16}, \frac{3}{16}, \frac{7}{16}, \frac{7}{32}\right\}$ and $T = \left\{\frac{1}{4}, \frac{1}{4}, \frac{13}{64}, \frac{13}{64}\right\}$. Figure 1.2 illustrates a mixing graph that converts I into T, exactly. Therefore, T is reachable from I. (See, also, G_1 in Figure 1.1 for a target set $T = \left\{\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{13}{16}, \frac{13}{16}, \frac{7}{8}\right\}$ that is reachable from $I = \{0, 0, 1, 1, 1, 1\}$.)

For T to be reachable from I, clearly, I and T must have the same cardinality and equal reactant volumes. However, these conditions are not sufficient. For example, $T = \left\{\frac{1}{4}, \frac{3}{4}\right\}$ is not reachable from $I = \{0, 1\}$, because producing $\frac{1}{4}$ from I requires at least two buffer droplets and one reactant droplet, but T itself contains only two droplets. (A mixing graph for target set $T = \left\{\frac{1}{4}, \frac{3}{4}\right\}$ with two waste droplets is shown in Figure 1.3.)

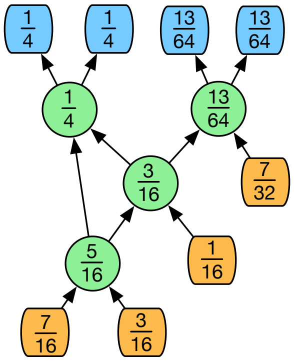

Figure 1.2: Mixing graph that converts the input set $I = \left\{\frac{1}{16}, \frac{3}{16}, \frac{7}{16}, \frac{7}{32}\right\}$ into the target set $T = \left\{\frac{1}{4}, \frac{1}{4}, \frac{13}{64}, \frac{13}{64}\right\}$.

As another example, consider $I = \left\{\frac{1}{16}, \frac{1}{8}, \frac{3}{8}, \frac{7}{8}\right\}$ and $T = \left\{\frac{1}{16}, \frac{3}{16}, \frac{7}{16}, \frac{3}{4}\right\}$. First, to produce target droplet $\frac{3}{4}$, we eventually have to mix droplet $\frac{7}{8} \in I$, because it is the only droplet in I with a concentration higher than $\frac{3}{4}$. Now, the second highest concentration in I is $\frac{3}{8}$, therefore the highest concentration we can obtain by mixing $\frac{7}{8}$ is $\frac{5}{8}$, when we mix it with $\frac{3}{8}$. However, $\frac{5}{8} < \frac{3}{4}$ and thus a droplet with concentration $\frac{3}{4}$ cannot be produced from I. Hence, T is not reachable from I.

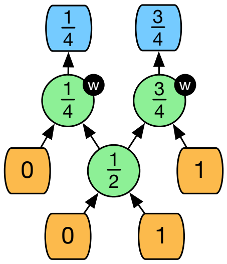

Figure 1.3: Mixing graph for target set $T = \left\{\frac{1}{4}, \frac{3}{4}\right\}$.

In typical applications, the input set I consists of pure buffer and reactant droplets (that is, only 0's and 1's). We denote this variant by MixProducibility, and target sets reachable from such input sets are called *mix-producible*, or just *producible*; see set $\{\frac{1}{8}, \frac{1}{8}, \frac{1}{4}, \frac{13}{16}, \frac{13}{16}, \frac{7}{8}\}$ in Figure 1.1 for an example of a producible set. If waste is allowed, then, naturally, MixProducibility can be extended to an optimization problem where the objective is to design a mixing graph that generates T while minimizing waste. Alternative objective functions have been studied: for example minimizing the reactant waste, minimizing the number of micro-mixers, and others.

MixProducibility is not likely to be computationally easier than MixReachability. (For example, via a simple linear mapping, any algorithm that solves MixProducibility can also solve the variant of MixReachability where the input set has droplets of *any* two given concentrations.) To the best of our knowledge, the computational complexity of MixReachability and MixProducibility remains open. In fact, even for single-droplet targets, we do not know whether the minimum-waste function can be effectively computed. These theoretical gaps also motivate our work.

The contributions of this work are mainly related to MixReachability and MixProducibility. In particular, they can be classified in the following areas:

- Algorithms for waste-minimization
- Computational complexity
- Characterization of reachable and producible sets

An overview of our contributions is given in Section 1.3.

1.2 Literature Review

Most of the previous papers in the literature study designing mixing graphs using heuristic approaches. Earlier studies focused on producing *single-droplet targets*, where only one droplet of some desired concentration is needed. This line of research was pioneered by Thies *et al.* [65], who proposed an algorithm called `Min-Mix` that constructs a mixing graph for a single-droplet target. Roy *et al.* [50] developed a single-droplet target algorithm called `DMRW` that considered waste reduction and the number of mixing operations. Huang *et al.* [26] and Chiang *et al.* [3] proposed single-droplet algorithms designed to minimize reactant usage.

Many applications, however, require target sets with droplets of multiple concentrations (see for example [23,61,62,72,73]). Typically, target sets arising in practice involve concentration values that form arithmetic or geometric sequences (referred to, respectively, as "linear" and "logarithmic" in some literature – see, for example [32]), but the special form of such sets does not seem to facilitate the design of mixing graphs. For multiple-concentration targets, Huang *et al.* [27] proposed an algorithm called `WARA`, which is an extension of Algorithm `REMIA` from [26]. Mitra *et al.* [43] modeled the problem of producing multiple concentrations as an instance of the Asymmetric TSP on a de Brujin graph.

The papers cited above describe heuristic algorithms with no formal performance guarantees. Dinh *et al.* [7] took a more rigorous approach. They modeled the problem as an equal-split flow problem [42] on a "universal" graph that contains all possible mixing graphs of maximum depth d as subgraphs, where d is the maximum precision in the target set T. By assigning appropriate capacity and cost values to edges, the problem of extracting a mixing subgraph that minimizes waste can be represented as an integer linear program, resulting in an algorithm that is doubly exponential in d. Unfortunately, contrary to the claim in [7], their algorithm does not necessarily produce mixing graphs with minimum waste. The reason is that there are target sets with maximum precision d that require mixing graphs of depth larger than d to be produced without any waste. In particular, target sets of the form $T = \left\{ 2^{-d}\, , \, 2^d + 1 : 1 - 2^{-d} \right\}$ are producible only by mixing graphs of depth at least $2d$.

We next show that producing a target set $T = \left\{ 2^{-d}\, , \, 2^d + 1 : 1 - 2^{-d} \right\}$, where d is the maximum precision in T, without any waste, requires depth at least $2d$. The argument is as follows. Consider a micro-mixer that produces droplet of concentration 2^{-d} in T. This mixer actually produces *two* droplets with concentration 2^{-d} and is at least depth d, because it takes d steps to dilute 1 to concentration 2^{-d}. The fluid in the second droplet with concentration 2^{-d} (the one not in T) produced by this mixer must also end up in some droplets of T and its concentration will increase to $1 - 2^{-d}$. The *buffer concentration* in this droplet is $1 - 2^{-d}$ and it takes at least $d - 1$ steps to reduce it to 2^{-d}, in order to obtain a droplet with reactant concentration $1 - 2^{-d}$. We can therefore conclude that the depth of any mixing graph for T is at least $2d$. Figure 1.4 shows an example for the target set $T = \left\{ \frac{1}{8}\, , \, 17 : \frac{7}{8} \right\}$ (one droplet with concentration $\frac{1}{8}$ and seventeen droplets with concentration $\frac{7}{8}$) with precision $d = 3$, for which a mixing graph of depth $2d = 6$ is constructed.

In the following sections we present a more in-depth description of the literature mentioned above.

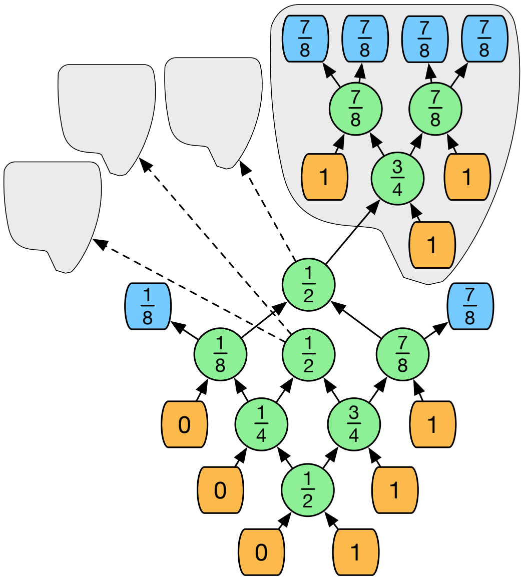

Figure 1.4: A (partial) mixing graph of depth 6 that produces $T = \left\{ \frac{1}{8}, 17 : \frac{7}{8} \right\}$ with precision $d = 3$. Each of the three empty-shaded areas on the left represent the exact same subgraph used in the rightmost shaded area, which takes as input one droplet of concentration $\frac{1}{2}$ and produces four droplets of concentration $\frac{7}{8}$.

1.2.1 Single-Droplet Targets

In this section, we present the work in the literature on designing mixing graphs that produce a single-droplet target with a specified concentration.

1.2.1.1 Algorithm `Min-Mix`

In [65], Thies *et al.* pioneered research on mixing (dilution) optimization for microfluidic mixing chips by proposing an algorithm called `Min-Mix` that produces a mixing graph for a single-droplet target with a desired concentration. Their algorithm is sometimes also referred to as the *bit-scanning algorithm* in the literature.

Algorithm `Min-Mix` computes a mixing graph based on the binary representation of the target droplet concentration. Starting at the least-significant bit, the mixing process mixes a buffer droplet with a reactant droplet at depth 1. Then, if the second least-significant bit is one, a reactant droplet is mixed at depth 2, otherwise a buffer droplet is mixed. The process continues until the most-significant bit has been processed and the target droplet has been produced. The mixing graph produced by the algorithm is always a tree; in fact, it consists of a path with leaves representing fluid dispensers attached to it.

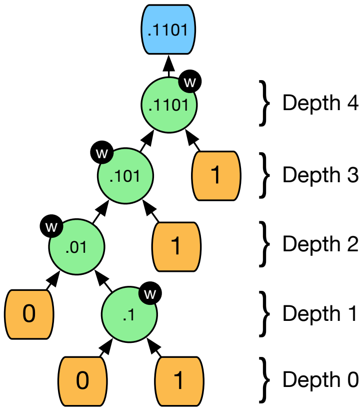

Figure 1.5: Mixing graph example for Algorithm `Min-Mix` with target concentration $\frac{13}{16}$, or 0.1101 in binary. The number of waste droplets equals the depth of the graph: in this case 4. (For simplicity, droplet concentrations are shown in binary.)

In Figure 1.5, we illustrate an example of a mixing graph produced by Algorithm `Min-Mix`. Given target droplet with reactant concentration $\frac{13}{16}$, whose binary representation is 0.1101, the mixing graph produces concentrations 0.1, 0.01, 0.101 and 0.1101 at depth 1, 2, 3 and 4 respectively.

Although this algorithm minimizes the number of mixing operations necessary to produce the target droplet (therefore the name, `Min-Mix`), it produces a lot of waste – one waste droplet per depth in the graph (depths 1, 2, 3 and 4 in Figure 1.5). However, this approach has been the basis of some other algorithms that we discuss next.

1.2.1.2 Algorithm `DMRW`

Roy *et al.* in [50] developed a single-droplet target algorithm called *dilution and mixing with reduced wastage* (`DMRW`), which considered waste reduction in addition to the number of mixing operations. This algorithm uses k:k micro-mixers, for appropriate values of k that are determined by the algorithm. (We can think of such k:k micro-mixers as a collection of k "parallel" 1:1 micro-mixers.)

Given the target concentration c_t and two initial fluids with concentration $c_l$and c_r, respectively, such that $0 \leq c_l < c_t < c_r \leq 1$, the goal is to find a sequence of k:k-mixing operations that produce c_t from c_l and c_r, with an error at most $\frac{1}{2^\gamma}$. Algorithm `DMRW` is based on binary search. At each step, concentration c_m is produced by mixing c_l with c_r, where $c_m = \frac{1}{2}(c_l + c_r)$. Thus, there is only one k:k-mixer per step. If c_m is smaller than c_t, then c_l is set to c_m. Otherwise, c_r is set to c_m instead. If $|c_m - c_t| < \frac{1}{2^\gamma}$, the algorithm stops and outputs c_m. The number of iterations is at most γ, because the concentration interval between c_l and c_r (which includes c_t) halves after each mixing operation.

If the initial concentrations are $c_l = 0$ and $c_r = 1$, then the algorithm computes the exact value of c_t after d mixing operations, where d is the precision of c_t. This is because in the i^{th} step we have $|c_t - c_m| < 2^{-i}$, and c_m has precision i. So after d steps we have $|c_t - c_m| < 2^{-d}$ and both c_t and c_m have precision d, which implies that they must be equal.

We must still specify the number of droplets that each mixer needs to produce. This value can be computed by backtracking, starting from the target node. When a node v producing a concentration c is considered, and the nodes processed earlier require ℓ droplets with concentration c, then v will represent a k:k micro-mixer (or k 1:1 micro-mixers) for

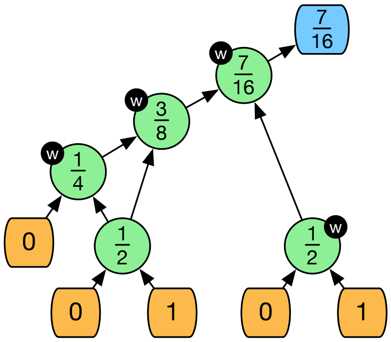

Figure 1.6: Mixing graph example for Algorithm `DMRW` with target concentration $\frac{7}{16}$. This example shows two 1:1 micro-mixers, instead of one 2:2 micro-mixer, for producing three droplets of concentration $\frac{1}{2}$.

$k = \lceil \ell/2 \rceil$. If ℓ is odd, then v will produce one droplet of waste. Figure 1.6 illustrates the mixing graph computed by Algorithm `DMRW` for the target concentration $\frac{7}{16}$.

1.2.1.3 Algorithm `REMIA`

In [26], Huang *et al.* proposed an algorithm for single-droplet targets called `REMIA`, which uses 1:1-mixers and was designed to minimize reactant volume. Their approach consists of two phases: an *interpolated dilution* phase and an *exponential dilution* phase.

Let a *prime concentration value* (pcv) denote a concentration value with exactly one bit 1 in its binary representation; that is, a number of the form 2^{-j}, for some non-negative integer j. First, in the interpolated dilution phase they compute a pcv mixing tree, which is a mixing graph with a tree-like structure. This mixing tree generates the target droplet starting from a droplet set of pcvs, instead of buffer and reactant droplets. Thus, every leaf in a pcv mixing tree is a pcv node.

Let c_t be the concentration of the target droplet. In the first step, we express c_t as $c_t = 2^{-i} + (c_t - 2^{-i})$, where i is the most-significant non-zero bit in c_t. We can obtain c_t by mixing one droplet with concentration 2^{-i+1} and one with concentration $2 \cdot (c_t - 2^{-i})$. So, in the pcv mixing tree, we will have two nodes with concentrations 2^{-i+1} and $2 \cdot (c_t - 2^{-i})$ respectively, pointing to node c_t. The node with concentration 2^{-i+1} is a pcv, so we put it aside as a leaf. On the other hand, the node with concentration $2 \cdot (c_t - 2^{-i})$ may or may not be a pcv. If it is a pcv, we put it aside as well, and the interpolation process ends. If not, it becomes the new target concentration c_t, and another round of the interpolated dilution phase begins. G_1 in Figure 1.7 shows the mixing graph obtained by the interpolated dilution phase for target concentration $\frac{43}{64}$.

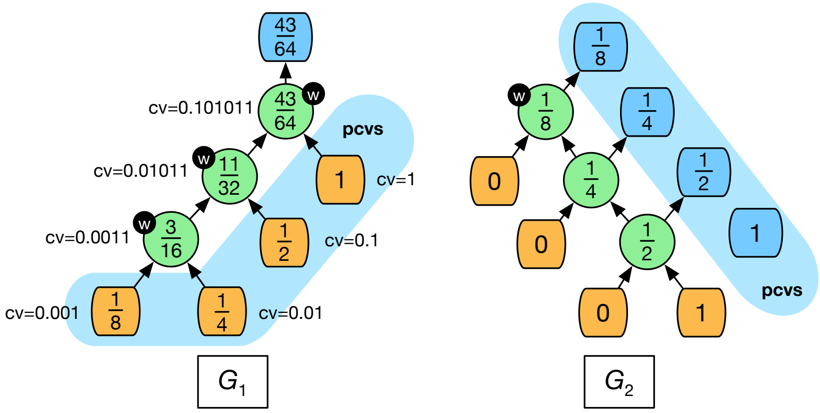

Figure 1.7: Example of Algorithm `REMIA` for the target concentration $\frac{43}{64}$. G_1 and G_2 are the mixing graphs obtained by the interpolated dilution and exponential dilution phases, respectively.

Next, in the exponential dilution phase, the set of all pcv concentrations found in the interpolated dilution phase becomes our target set of concentrations, and a new mixing graph is computed. The set of pcvs is processed in order of their increasing concentration values. Let c_i and c_j be the next pcvs in the pcv target set respectively. If $c_i = 1$, then c_i represents a reactant droplet and we just add a reactant node to our new mixing graph.

Otherwise, we know that c_i can be generated by mixing a pcv droplet with concentration $2 \cdot c_i$ with a buffer droplet. So we add a mixer node with concentration c_i with one input being a buffer node and another being a node with concentration $2 \cdot c_i$. We also add the node with concentration $2 \cdot c_i$ to the target set of our pcvs.

Now, if $c_i = c_j$, then the two output droplets from the target set of pcvs will be produced, and we remove both from the target set. Otherwise, we only remove c_i and one other droplet of concentration c_i becomes waste. (In this case, c_j will be considered again later in the process.) The process continues until the target set of pcvs becomes empty. G_2 in Figure 1.7 shows the mixing graph obtained by the exponential dilution phase for the target set of pcvs in G_1.

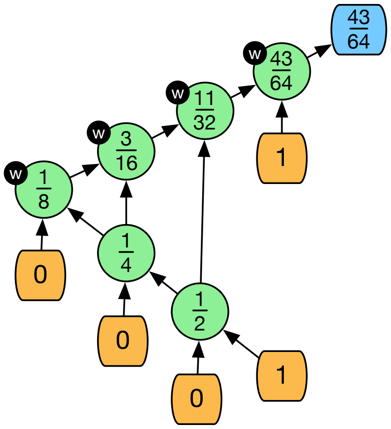

Figure 1.8: Complete mixing graph produced by Algorithm `REMIA` for concentration $\frac{43}{64}$.

The complete mixing graph is obtained by combining the graphs from the interpolated dilution and exponential dilution phases: each leaf node from the interpolated dilution phase is identified with the node of the exponential dilution tree with the same pcv label. Figure 1.8 illustrates the final mixing graph of Algorithm `REMIA` for the target concentration

$\frac{43}{64}$. (Note that the complete mixing graph produced by Algorithm REMIA might not be a tree anymore.)

1.2.1.4 Algorithm GORMA

In [3], Chiang *et al.* proposed a branch-and-bound algorithm called GORMA, or *graph-based optimal reactant minimization algorithm*, for single-droplet targets. This algorithm consists of two main processes: *complete mixing tree* (CMT) enumeration and *maximal droplet sharing.* A CMT is basically a mixing graph that is a binary tree, where leaves are either buffer or reactant nodes, and where the root represents the target droplet. Also, in a CMT, the precision of the concentration of any node (except the root) is smaller than that of its immediate ancestor. Because of the 1:1-mixers, every non-leaf node in a CMT has waste associated to it.

Algorithm GORMA can be divided into two phases. In the first phase, an initial CMT B for c_t is constructed. Then, in the second phase, using B as starting point, "all" CMTs for c_t are enumerated. We summarize each of these phases below. Let c_x be an arbitrary concentration. We define a *descending concentration pair* of c_x as a pair of concentrations (c_u, c_v) such that $c_x = \frac{1}{2}(c_u + c_v)$ and where the precision of both c_u and c_v is smaller than the precision of c_x. We also define a *shareable node pair* (u, v), with concentrations c_u and c_v, respectively, as a pair of nodes such that $c_u = c_v$, and where both u and v have waste associated with them. Then, an *unshareable mixing graph* is defined as a mixing graph without any shareable node pair.

Constructing B for c_t. First, add a root node t (for c_t) to B, then associate to it a list of every possible descending concentration pair for c_t and further mark each such pair as unprocessed. Then, take the first unprocessed concentration pair (c_u, c_v) and processes it as follows: mark (c_u, c_v) as processed and add nodes u and v, for c_u and c_v respectively, to B, with edges (u, t) and (v, t) (where u and v are t's left and right immediate descendants, respectively). Finally, recursively construct the initial CMTs for u and v; u's CMT first. (No initial CMTs are computed for buffer and reactant nodes, since they are leaf nodes.)

Enumerating all CMTs for c_t. Let B be the CMT constructed above for c_t. First, we convert B into an unshareable mixing graph by using the maximal droplet sharing process, which basically consists of identifying shareable node pairs (u, v) in B, and replacing v and v's *exclusive descendants* with a node corresponding to u's waste, until there are no more shareable node pairs in B.

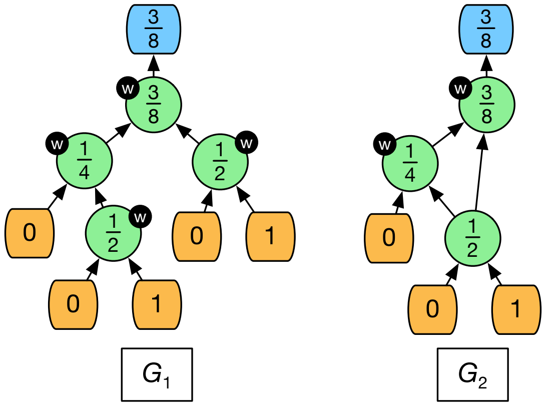

Figure 1.9: Algorithm GORMA droplet sharing example. G_1 illustrates the initial CMT while G_2 illustrates the resulting graph after sharing one of the droplets of concentration $\frac{1}{2}$.

Figure 1.9 shows an example of droplet sharing in an arbitrary CMT. G_1 shows the initial CMT. Then, G_2 can be obtained from G_1 by using the waste droplet of $\frac{1}{2}$, in the left sub-tree of node $\frac{3}{8}$, to replace the corresponding node (and its corresponding exclusive descendants) in the right sub-tree of node $\frac{3}{8}$.

Next, the CMT enumeration process continues using B as reference. Let x be the last modified node in B with unprocessed descending concentration pairs. (If there is no such node x, then all feasible CMTs have been enumerated and the CMT with minimum waste is used as the final mixing graph.) Then, using x as a root, corresponding CMTs are recursively computed. (However, regarding descending concentration pairs for x, only those marked as unprocessed are considered.) As every feasible CMT must be enumerated, the enumeration of a CMT for x leads to the re-enumeration of previously-enumerated CMTs. For example, let $y \in B$ be an ancestor of x, such that y's left sub-tree contains x. Then, if a CMT rooted at x is enumerated, y's right child CMT must also be re-enumerated.

To avoid unnecessary re-enumerations, Chiang *et al.* proposed the following branch-and-bound technique: instead of re-enumerating y's right child CMT on the spot, a dummy node is used to temporarily replace it. Let the CMT containing the dummy node be D. Only when D's unshareable mixing graph is better (with respect to reactant usage) than the current best CMT is y's right child CMT re-enumerated. Otherwise, the dummy node replacement is rolled-back and the enumeration process continues.

Figure 1.10 illustrates a CMT enumeration example in progress. G_1 shows the current CMT. Then, G_2 shows node $\frac{3}{8}$ processing descending concentration pair $(\frac{5}{16}, \frac{7}{16})$. On G_3, as node $\frac{3}{8}$ is in the left sub-tree of node $\frac{17}{32}$, node $\frac{11}{16}$ re-enumerates its initial descending concentration pair, which is $(\frac{3}{8}, 1)$.

Chiang *et al.* claimed that the order in that shareable node pairs are processed is irrelevant to the final outcome. However, in Figure 1.11 we show that this is not true. If node $\frac{3}{8}$ (on the right sub-tree of node $\frac{15}{32}$) replaces node $\frac{3}{8}$ (on the left sub-tree of node $\frac{15}{32}$)

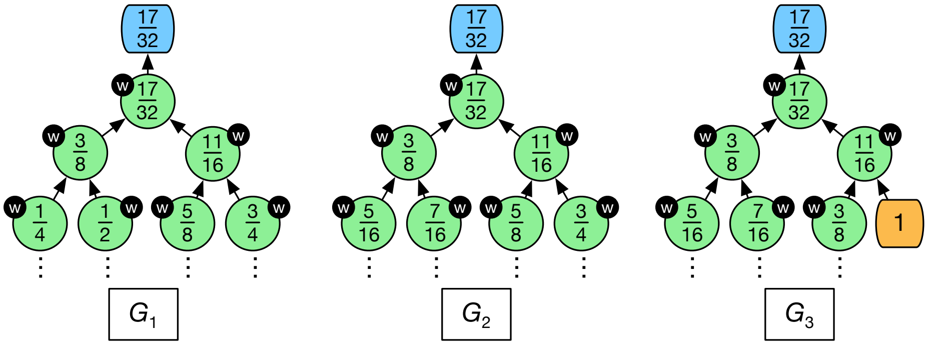

Figure 1.10: Algorithm GORMA CMT enumeration example. G_1 shows the current CMT. G_2 shows node $\frac{3}{8}$ processing descending concentration pair $(\frac{5}{16}, \frac{7}{16})$. G_3 shows node $\frac{11}{16}$ processing descending concentration pair $(\frac{3}{8}, 1)$, again caused by the update on G_2.

in the first droplet sharing operation, then a droplet sharing operation involving nodes of $\frac{3}{4}$ will be lost, leading to an unshareable mixing graph that uses 3 reactant droplets, while the optimal uses only 2. Thus, the order in which shareable node pairs are chosen and the order in which nodes are replaced matters.

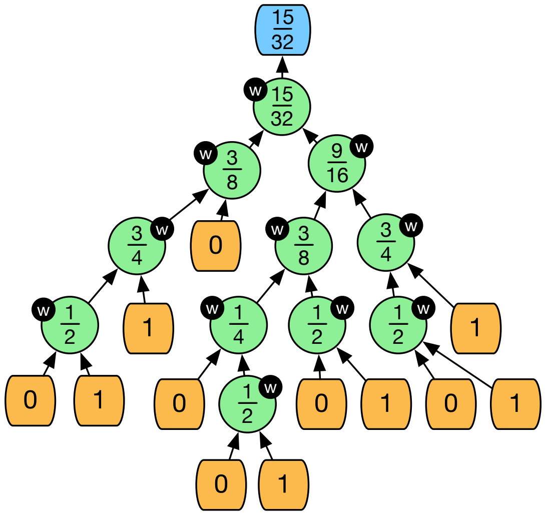

Figure 1.11: Algorithm GORMA droplet sharing counter example. If node $\frac{3}{8}$ is chosen as the first droplet sharing pair, the optimal unshareable mixing graph might not be found.

1.2.2 Multiple-Concentrations Targets

In this section, we present the work in the literature on designing mixing graphs that produce many droplets with different specified concentrations.

1.2.2.1 Algorithm `WARA`

Algorithm `WARA`, proposed by Huang *et al.* in [27], is an extension of Algorithm `REMIA` (see Section 1.2.1.3). It is based on the idea of *waste recycling*, which is implemented by adding two more phases called *droplet sharing* (as in Algorithm `GORMA`) and *droplet replacement*.

First, Algorithm `WARA` computes a pcv mixing tree for each target droplet using Algorithm `REMIA`. Initially, every non-leaf node in a mixing tree has waste associated with it, thus every non-leaf node in a mixing tree is a *reusable* node. Also, every leaf in a mixing tree is a pcv node.

Second, in the droplet sharing phase, Algorithm `WARA` computes an unshareable mixing graph G by applying maximal droplet sharing (as Algorithm `GORMA` does) to the set of pcv mixing trees produced by Algorithm `REMIA`.

Third, in the droplet replacement phase, Algorithm `WARA` finds *replacement candidate pairs* (u, v) for z such that $u, v, z \in G$, z is a descendant of neither u nor v, $c_z = \frac{1}{2}(c_u + c_v)$ and both u and v are reusable. Thus, the output produced by mixing u and v can safely replace z and its exclusive descendants. Once there are no more replacement candidate pairs satisfying the above constraints, new replacement candidate pairs are explored by allowing either u or v to be a pcv.

Algorithm WARA introduces two sorting criteria for replacement candidate pair processing. Let G' be obtained from G after processing a replacement candidate pair (u, v) in G. Define the *reactant minimization gain* as

$$gain(u, v) = ERU(G) - ERU(G')$$

where $ERU(G)$ (for *essential reactant usage*) is the sum of the concentration values of all the leaves in G. Also, define the *uniqueness* of (u, v) as:

$$uniq(u, v) = min(ac(u), ac(v))$$

where $ac(u)$ is the number of replacement candidate pairs that contain u, if u is not a pcv, and $ac(u) = \infty$, if u is a pcv. (Similarly for $ac(v)$.) In order to minimize reactact usage, during the droplet replacement phase, the algorithm chooses replacement candidate pairs with high reactant minimization gain and high uniqueness.

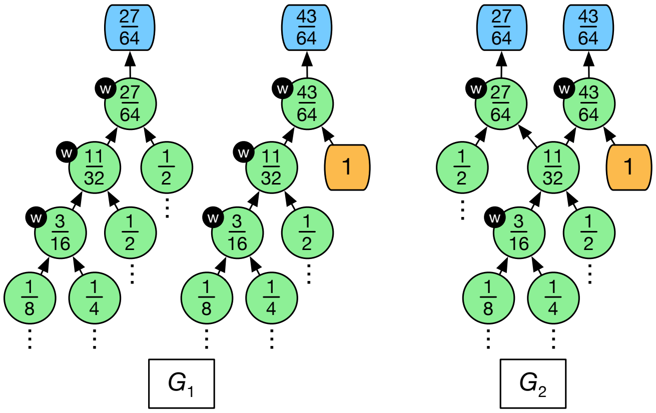

Figure 1.12: Algorithm WARA droplet sharing example. G_2 can be obtained from G_1 by replacing node of $\frac{11}{32}$ in the sub-tree rooted from $\frac{43}{64}$, by the corresponding node of $\frac{11}{32}$ in the sub-tree rooted from $\frac{27}{64}$.

Let H be the graph produced after the droplet replacement phase. Next, Algorithm WARA produces the set of pcvs that are leaves in H by using Algorithm REMIA's exponential dilution phase. Call this graph H'. Finally, Algorithm WARA combines H and H': each leaf node from H is identified with the node of H' with the same pcv label. Figure 1.12 shows an example of droplet sharing. Figure 1.13 shows an example of droplet replacement where in G_1, node $\frac{11}{32}$ is reusable and node $\frac{1}{16}$ is a leaf pcv node, and their mixing can replace (partially) the sub-tree rooted at $\frac{13}{64}$.

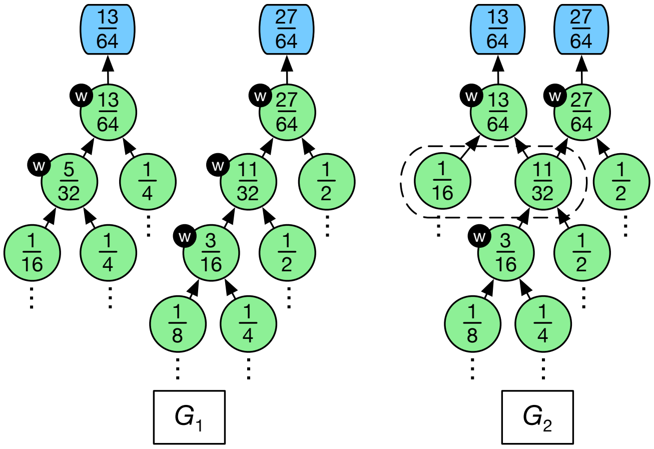

Figure 1.13: WARA droplet replacement example. Nodes of $\frac{11}{16}$ and $\frac{11}{32}$ in G_1 can be used to replace the sub-tree rooted at $\frac{13}{64}$. The resulting mixing graph is shown by G_2.

1.2.2.2 Equal-Split Flow Model

In [7], Dinh *et al.* proposed an approach (ESFM) to computing mixing graphs for multiple target concentrations that is based on modeling the problem as the equal-split flow problem [42]. The idea is to construct a "universal" flow graph that contains all possible mixing graphs as subgraphs. By assigning appropriate capacity and cost values to edges, minimum-cost equitable flows in this graph will determine a mixing graph for a given target set of concentrations. The problem of computing such equitable flows can be represented as an integer linear program (ILP). The objective function of this ILP can be customized to optimize reactant usage or waste.

Assign cost values $cost_r$ and $cost_b$ to a reactant droplet and buffer droplet, respectively. The optimization problem addressed in [7] can then be formulated as follows: given the target set $T = \{(q_1 : c_1), (q_2 : c_2), \ldots, (q_m : c_m)\}$, where q_i represents the number of droplets of c_i for all i, find a mixing graph that generates T and minimizes the cost function

$$\mathcal{F} = u_r \cdot cost_r + u_b \cdot cost_b.$$

where u_r and u_b denote the number of reactant and buffer droplets used by the mixing graph, respectively. By setting $cost_r = 1$ and $cost_b = 0$ the reactant usage is being minimized. Setting $cost_r = 1$ and $cost_b = 1$ will minimize the total number of droplets used, and thus also the waste.

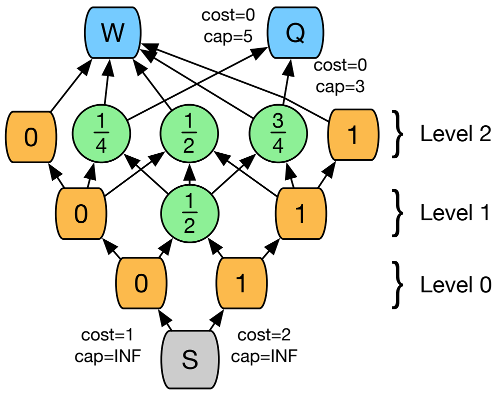

Figure 1.14: ESFM network structure for target set $T = \{(5 : \frac{1}{4}), (3 : \frac{3}{4})\}$, with $cost_b = 1$, $cost_r = 2$ and $d = 2$. Edges without label have zero cost and infinite capacity.

Let d be the precision of the target set T. The flow network used by Dinh *et al.* has $d+1$ levels numbered $l = 0, 1, ..., d$. For each level l and each concentration c with precision at most l, the network contains a node denoted $v_{l,c}$. For any two nodes $v_{l,a}$ and $v_{l,b}$ at level

l, and for $c = \frac{1}{2}(a+b)$, the network has directed edges $(v_{l,a}, v_{l+1,c})$ and $(v_{l,b}, v_{l+1,c})$ to level $l+1$, with infinite capacities and zero cost. In addition, we have three more nodes. One of them is the source node S with edges $(S, v_{0,0})$, $(S, v_{0,2^d})$ with infinite capacities and costs $cost_b$ and $cost_r$, respectively. We also have the waste node W and the sink node Q. Node W has incoming edges $(v_{d,c}, W)$ with infinite capacity and zero cost from every node $v_{d,c}$ at level d. Node Q has an incoming edge (v_{d,c_i}, Q) with capacity q_i and zero cost, for each pair $(q_i : c_i) \in T$. Figure 1.14 shows the network structure for target set $T = \{(5 : \frac{1}{4}), (3 : \frac{3}{4})\}$, with $cost_b = 1$, $cost_r = 2$ and $d = 2$. Edges without labels have zero cost and infinite capacity.

Next, we define the constraints. Let $f(u,v)$ be the variable representing the flow on an edge (u,v). We include all usual constraints for flow problems, namely capacity constraints and flow preservation constraints. In addition, we have equitable flow constraints, defined as follows: for any three nodes $v_{l,a}$, $v_{l,b}$ and $v_{l+1,c}$ such that $c = \frac{1}{2}(a+b)$, we require that $f(v_{l,a}, v_{l+1,c}) = f(v_{l,b}, v_{l+1,c})$. These constraints ensure that the fluids with concentrations a and b arriving from $v_{l,a}$ and $v_{l,b}$ have the same volumes, so that they can be mixed (using 1:1-mixers represented by $v_{l+1,c}$) to obtain a fluid with concentration c.

We now give a complete `ILP` from [7]. Let $cap(u,v)$ be the capacity of edge (u,v). Using the network flows notation, their `ILP` looks as follows:

$$
\begin{array}{llcll}
\textbf{minimize} & \mathcal{F} & = & f(S, v_{0,0}) \cdot cost_b + f(S, v_{0,2^d}) \cdot cost_r & \\
\textbf{subject to} & f(u,v) & \leq & cap(u,v), & \text{for } (u,v) \in E \text{ (capacity constraint)} \\
 & f(v_{d,c_i}, Q) & = & q_i, & \text{for } (q_i : c_i) \in T \text{ (target concentrations)} \\
 & \sum\limits_{(u,v)\in E} f(u,v) & = & \sum\limits_{(v,w)\in E} f(v,w), & \text{for } v \in V - (S \cup W \cup Q) \text{ (flow conservation)} \\
 & f(v_{l,a}, v_{l+1,c}) & = & f(v_{l,b}, v_{l+1,c}) & \text{for } 0 \leq l < d \text{ and } c = \frac{a+b}{2} \text{ (equitable flow)}
\end{array}
$$

All flow values are required to be non-negative integers. (V and E above denote the set of vertices and edges in the network, respectively.)

Any integral equitable flow that is a solution of the above `ILP` represents a mixing graph, by interpreting the nodes in the network as mixers. More specifically, for any pair $v_{l-1,a}$ and $v_{l-1,b}$ such that $c = \frac{1}{2}(a+b)$, if $f(v_{l-1,a}, v_{l,c}) = (v_{l-1,b}, v_{l,c}) = k$, we introduce a k:k-mixer with inputs from two mixers corresponding to $v_{l-1,a}$ and $v_{l-1,b}$. (Recall that k:k-mixers are equivalent to k parallel 1:1-mixers.)

As we proved in Section 1.2, however, this approach actually has a flaw: while the `ILP` correctly produces a mixing graph, this mixing graph may not necessarily be optimal for a given target set. The reason is that an optimal mixing graph for a target set with precision d may have depth larger than d, while [7] tacitly assumes that depth d is sufficient.

1.2.2.3 Asymmetric Traveling Salesman Model

In [43], Mitra *et al.* approached the problem of multiple-concentrations targets by modeling it as an *Asymmetric Traveling Salesman Problem on a de Brujin graph* (`ATSP-DBG`). Let T be a target set of droplets with specified concentrations. The idea is to produce T by a linear sequence of mixing operations using 1:1-mixers, where at each step the current droplet is mixed either with buffer or reactant. (Thus, in this approach, no fluid storage is required.) At each intermediate mixing step, one droplet is sent to the next mixer, while the other one is either discarded (becomes waste) or used as output (becomes a target droplet). Under the assumption that the mixing graph has this form, the objective in [43] is to minimize the number of mixing operations.

Let d be the precision of the target set T. All concentrations are manipulated as d-bit binary strings. If concentrations with higher precision arise, they are rounded-off to d bits (e.g. for $d = 4$, mixing 0.1011 with 0 produces concentration 0.0101). This introduces an error bounded above by $\frac{1}{2^d}$.

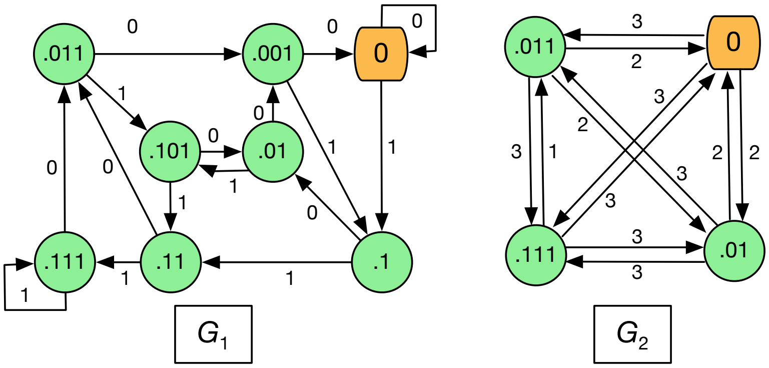

Figure 1.15: Example of `ATSP-DBG` for $T = \{\frac{1}{4}, \frac{3}{8}, \frac{7}{8}\}$ with $d = 3$. G_1 shows a graph that is isomorphic to the de Brujin graph $\mathcal{B}(2, d)$. G_2 shows the directed weighted graph for which a traveling salesman tour is computed.

The process of mixing with either reactant or buffer can be represented by a directed graph $G = (V, E)$ as follows: For each concentration c_u with precision at most d, G contains the corresponding node u. We introduce an edge (u, v) with label 0 if mixing a droplet with concentration c_u with a buffer droplet produces concentration c_v. Similarly, we introduce an edge (u, v) with label 1 if mixing a droplet with concentration c_u with a reactant droplet produces concentration c_v. (In both cases, rounding can occur, as explained earlier.) G_1 in Figure 1.15 illustrates an example of this construction when $d = 3$.

It can be shown that G is isomorphic to the d^{th} order binary de Brujin graph $\mathcal{B}(2, d)$, and thus it contains a Hamiltonian cycle. This in turns implies that there is a path in G that visits all vertices corresponding to the target set T; or, in other words, that T can be generated (approximately, as we allow rounding) by repeated mixing with buffer and reactant only. The path that corresponds to T, however, could contain many additional vertices, representing additional mixers and generating additional waste.

To optimize their mixing graphs, [43] construct a smaller graph G'. G' is a complete weighted directed graph. For each pair of concentrations c_u and c_v in T, graph G' has a directed edge (u, v) with weight $dist(c_u, c_v)$, where $dist(c_u, c_v)$ is the number of 1:1-mixers needed to generate c_v from c_u, by mixing only with buffer and reactant, and with rounding (as described earlier). The value of $dist(c_u, c_v)$ can be determined by scanning the bits of c_u and c_v from left to right and from right to left, respectively, to find the longest substring matching and by subtracting its length from d (note that in general $dist(c_u, c_v)$ may not be equal to $dist(c_v, c_u)$). Optimizing the number of mixing operations is equivalent to computing a minimum cost Hamiltonian path in G', in other words to solving the traveling salesman problem on G'. To this end, apply a heuristic algorithm.

G_2 in Figure 1.15 shows an example of the construction of G' for $T = \{\frac{1}{4}, \frac{3}{8}, \frac{7}{8}\}$. Figure 1.16 shows the final mixing graph for the example provided in Figure 1.15. One drawback of the approach in [43] is that every mixing operation that does not produce a target droplet produces a waste droplet. (Also, we remark that Mitra *et al.* estimated an absolute error to be at most $\frac{1}{2^{d+1}}$, which is not quite correct, since the error can propagate after each mixing operation and can accumulate to $\frac{1}{2^d}$.)

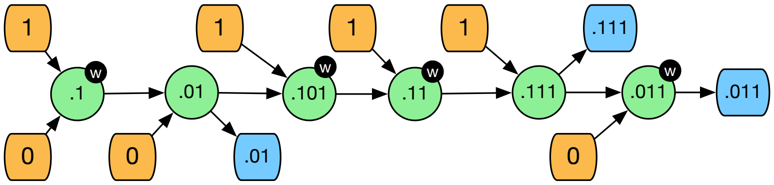

Figure 1.16: Mixing graph for ATSP-DBG when $T = \{\frac{1}{4}, \frac{3}{8}, \frac{7}{8}\}$ with $d = 3$.

1.3 Overview of Contributions

In this section we give an overview of our contributions. As mentioned in Section 1.1.5, regarding MixReachability and MixProducibility, our contributions can be divided into the following areas: (i) algorithms for waste-minimization, (ii) computational complexity, and (iii) characterization of reachable and producible sets.

To this end, we first summarize our contributions for (i). In Section 1.3.1 we overview a waste-minimization algorithm for the single-droplet target case of MixProducibility, which produces significantly less waste than state-of-the-art algorithms in an experimental comparison. Then, in Section 1.3.2 we overview our results for a sub-problem of MixReachability, where the objective is to design a mixing graph that perfectly mixes the input droplets. Furthermore, we show that this algorithm can be extended to provide a minimum-waste mixing graph for MixProducibility when the target set consists of a sufficiently-large number of same-concentration droplets.

Next, in Section 1.3.3.1 we overview our results for (ii), which result in an $\mathbb{NP}$-hardness proof for a restricted variant of MixReachability, where mixing graphs have fixed depth. Then, in Section 1.3.3.2, we outline a necessary condition for both reachable and producible sets. Although this condition is not a sufficient condition in our droplet model, it allows us to characterize special cases of MixProducibility in Section 1.3.3.3. Sections 1.3.3.2 and 1.3.3.3 comprises our contributions regarding (iii).

1.3.1 Waste-Minimization Algorithm for Single-Droplet Targets

To the best of our knowledge, the computational complexity of MixReachability and MixProducibility remains open. In fact, it is not even known whether the minimum-waste function is computable at all. This is true even for the single-droplet target case. Nonetheless, we designed an efficient waste-minimization algorithm for the single-droplet case of MixProducibility and provided a bound on its worst-case performance, which is significantly better than state-of-the-art algorithms, as shown in an experimental comparison. We give a bound on its running time, as a function of its output, which is the size of the produced graph. (This is because the output size is at least as large, and usually larger, than the input size, which is equal to d – the number of bits in the target concentration.) The algorithm's running time is linear in the size of the computed graph, whose size is $O(d^2)$. These contributions are presented in Chapter 2.

1.3.2 Algorithm for Perfect-Mixing

In this work, we focus on a sub-problem of MixReachability that we refer to as PerfectMixability, where the objective is to design a mixing graph that perfectly mixes the input droplets, producing only droplets of equal concentration. For this problem, we give a complete characterization of input sets for which such graphs exist and an efficient algorithm to construct these graphs.

We further extend the algorithm to construct minimum-waste mixing graphs for MixProducibility when the target sets consist of equal-concentration droplets with large multiplicities. The resulting algorithm is very convenient for applications that require large volumes of equal-concentration droplets. For example, in drug manufacturing, where mass production plays an essential role, large amounts of droplets with the same concentrations are required in early stages of the production. In such an application, waste minimization is a crucial component of the entire manufacturing process, thus highlighting the importance of our contribution. This algorithm is presented in Chapter 3.

1.3.3 Other Contributions

In this section, we overview other minor contributions. Specifically, an $\mathbb{NP}$-hardness proof for a variant of MixReachability where mixing graphs have fixed depth, a necessary condition for reachable and producible sets, and decidability proofs for special target sets.

1.3.3.1 $\mathbb{NP}$-Hardness of a Restricted Variant

As mentioned earlier, the computational complexity of MixReachability and MixProducibility remains open. Nevertheless, in this work we obtained an $\mathbb{NP}$-hardness proof for a variant of MixReachability where mixing graphs have fixed depth. In essence, we provide proofs for mixing graphs of depth 1 and 2, and then show the construction of an $\mathbb{NP}$-hardness proof for mixing graphs of fixed depth. These proofs and construction are given in detail in Section 4.1.

1.3.3.2 A Necessary Condition for Mixing Reachability

Our contributions include a necessary condition for reachable sets. We refer to this condition as *dominance relation*. In other words, let I and T be arbitrary initial and target sets, respectively. If T is reachable from I, then this dominance relation holds. In fact, in the fluid-based model, where we are allowed to mix arbitrary volumes of fluids, this dominance relation is also a sufficient condition.

However, in our droplet model, dominance relation is not a sufficient condition. Therefore, as an open problem, it remains to show a sufficient condition for the droplet model. These contributions are presented in Section 4.2.

1.3.3.3 Decidability of Special Cases

These contributions are solely focused on MixProducibility. We give decidability proofs for special types of configurations. In particular, we give decidability proofs for configurations with at most four droplets, configurations with precision 2 and configurations containing only pcvs. We also give a sufficient condition for producing configurations containing pcvs and one non-pcv. (Recall that a pcv denotes a concentration value with exactly one bit 1 in its binary representation.) These contributions are presented in Section 4.3.

Interestingly, the difficulty of such decidability proofs comes from the absence of bounds on the depth of mixing graphs as well as the precision of intermediate concentrations produced by these graphs. (This is the reason behind the flaw in Dinh *et al.* model mentioned in Section 1.2.2.2.) Therefore, determining such bounds is a key step in proving decidability.

Chapter 2

Waste-Minimization Algorithm for Single-Droplet Targets

In this chapter, regarding MixProducibility, we present a new efficient algorithm RPRIS (for *Recursive Precision Reduction with Initial Shift*) for designing mixing graphs for single-droplet targets, with the objective to minimize waste. Our algorithm was designed to provide improved worst-case waste estimate. Its main idea is quite natural: recursively, at each step it reduces the precision of the target droplet by 2, while only adding one waste droplet when adjusting the mixing graph during backtracking.

While designed with worst-case performance in mind, RPRIS significantly outperforms algorithms Min-Mix and DMRW in our experimental study, producing on average about 50% less waste than Min-Mix and between 21 and 25% less waste than DMRW, with the percentage increasing with the precision d of the target droplet. (It also produces about 40% less waste than REMIA.) Additionally, when compared to ESFM, RPRIS produces on average only about 7% additional waste. (See Section 1.2 in Chapter 1 for a description of Algorithms `Min-Mix`, `DMRW` and `ESFM`.)

Unlike earlier work in this area, that was strictly experimental, we introduce a performance measure for waste minimization algorithms and show that `RPRIS` has better worst-case performance than `Min-Mix` and `DMRW`. This measure is based on two attributes d and γ of the target concentration t. As defined earlier, d is the precision of t, and γ is defined as the number of equal leading bits in t's binary representation, not including the least-significant bit 1. For example, if $t = .00001011$ then $\gamma = 4$, and if $t = .1111$ then $\gamma = 3$. (Both d and γ are functions of t, but we skip the argument t, as it is always understood from context.) In the discussion below we provide more intuition and motivations for using these parameters.

We show that Algorithm `RPRIS` produces at most $\frac{1}{2}(d+\gamma)+2$ droplets of waste (see Theorem 2 in Section 2.4). In comparison, Algorithm `Min-Mix` from [65] produces exactly d droplets of waste to produce t, independently of the value of t. This means that the waste of `RPRIS` is about half that of `Min-Mix` for almost all concentrations t. (More formally, for a uniformly chosen random t with precision d the probability that the waste is larger than $(\frac{1}{2} - \epsilon)d$ vanishes when d grows, for any $\epsilon > 0$.) As for Algorithm `DMRW`, its average waste is better than that of `Min-Mix`, but its worst-case bound is still $d - \Omega(1)$ even for for small values of γ (say, when $t \in [\frac{1}{4}, \frac{3}{4}]$), while Algorithm `RPRIS`' waste is at most $d/2 + O(1)$ in this range.

In regard to time performance, for the problem of computing mixing graphs it would be reasonable to express the time complexity of an algorithm as a function of its output, which is the size of the produced graph. (This is because the output size is at least as large, and usually larger, than the input size, which is equal to d – the number of bits of t.) Algorithm RPRIS runs in time that is linear in the size of the computed graph, and the graphs computed by Algorithm RPRIS have size $O(d^2)$.

Discussion. To understand better our performance measure for waste, observe that the optimum waste is never smaller than $\gamma + 1$. This is because if the binary representation of t starts with γ 0's then any mixing graph has to use $\gamma + 1$ input droplets 0 and at least one droplet 1. (The case when the leading bits of t are 1's is symmetric.) For this reasons, a natural approach is to express the waste in the form $\gamma + f(d - \gamma)$, for some function $f()$. In Algorithm RPRIS we have $f(x) \approx \frac{1}{2}(x)$. It is not known whether smaller functions $f()$ can be achieved.

Ideally, one would like to develop "approximation" algorithms for waste minimization, that measure waste performance in terms of the additive or multiplicative approximation error, with respect to the optimum value. This is not realistic, however, in the current state-of-the-art, as no estimates for the optimum value are known; in fact, it is not even known whether the optimum value is computable.

2.1 Preliminaries

We use notation $\mathsf{prec}(c)$ for the precision of concentration c, that is the number of fractional bits in the binary representation of c. (All concentration values will have finite binary representation.) In other words, $\mathsf{prec}(c) = d \in \mathbb{Z}_{\geq 0}$ such that $c = a/2^d$ for an odd $a \in \mathbb{Z}$.

We will deal with sets of droplets, some possibly with equal concentrations. We define a *configuration* as a multiset of droplet concentrations. Let A be an arbitrary configuration. By $|A| = n$ we denote the number of droplets in A. We will often write a configuration as $A = \{f_1 : a_1, f_2 : a_2, ..., f_m : a_m\}$, where each a_i represents a different concentration and f_i denotes the multiplicity of a_i in A. (If $f_i = 1$, then, we will just write "a_i" instead of "$f_i : a_i$".) Naturally, we have $\sum_{i=1}^{m} f_i = n$.

We defined mixing graphs in the introduction. A mixing graph can be thought of, abstractly, as a linear mapping from the source values (usually 0's and 1's) to the sink values. Yet in this work, for convenience, we will assume that the source concentration vector is part of a mixing graph's specification, and that all sources, micro-mixers, and sinks are labeled by their associate concentration values.

We now define an operation of graph coupling. Consider two mixing graphs G_1 and G_2. Let T_1 be the output configuration (the concentration labels of the sink nodes) of G_1 and I_2 be the input configuration (the concentration labels of the source nodes) for G_2. To construct the *coupling* of G_1 and G_2, denoted $G_2 \bullet G_1$, we identify inlet edges of the sinks of G_1 with labels from $T_1 \cap I_2$ with outlet edges of the corresponding sources in G_2. More precisely, repeat the following steps as long as $T_1 \cap I_2 \neq \emptyset$: (1) choose any $a \in T_1 \cap I_2$, (2) choose any sink node t_1 of G_1 labeled a, and let (u_1, t_1) be its inlet edge, (3) choose any source node s_2 of G_2 labeled a, and let (s_2, v_2) be its outlet edge, (4) remove t_1 and s_2 and their incident edges, and finally, (5) add edge (u_1, v_2). The remaining sources of G_1 and G_2 become sources of $G_2 \bullet G_1$, and the remaining sinks of G_1 and G_2 become sinks of $G_2 \bullet G_1$. See Figure 2.1 for an example.

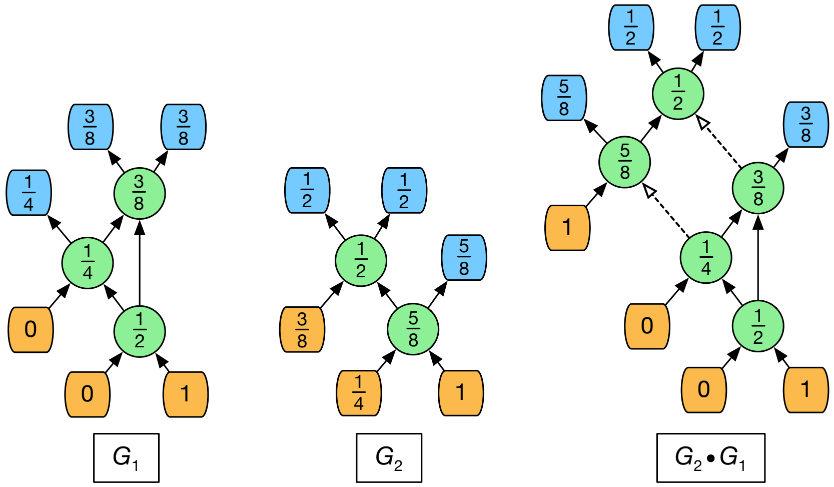

Figure 2.1: Coupling of two mixing graphs G_1 and G_2. $G_2 \bullet G_1$ is obtained by identifying inlet edges of two sinks of G_1, one labelled $\frac{1}{4}$ and one $\frac{3}{8}$, with the outlet edges of the corresponding sources of G_2. These new edges are shown as dotted arrows.

Next, we define converter graphs. An $(i:\alpha, j:\beta)$-*converter* is a mixing graph that produces a configuration of the form $T = \{i:\alpha, j:\beta\} \cup W$, where W denotes a set of waste droplets, and whose input droplets have concentration labels either 0 or 1. As an example, graph $G_2 \bullet G_1$ in Figure 2.1 can be interpreted either as a $(\frac{5}{8}, 2:\frac{1}{2})$-converter that produces a waste droplet of concentration $\frac{3}{8}$ or as a $(2:\frac{1}{2}, \frac{3}{8})$-converter that produces a waste droplet of concentration $\frac{5}{8}$.

If needed, to avoid clutter, sometimes we will use a more compact graphical representation of mixing graphs by aggregating (not necessarily all) nodes with the same concentration labels into a single node, and with edges labeled by the number of droplets that flow through them. (We will never aggregate two micro-mixer nodes if they both produce a droplet of waste.) If the label of an edge is 1, then we will simply omit the label. See Figure 2.2 for an example of such a compact representation.

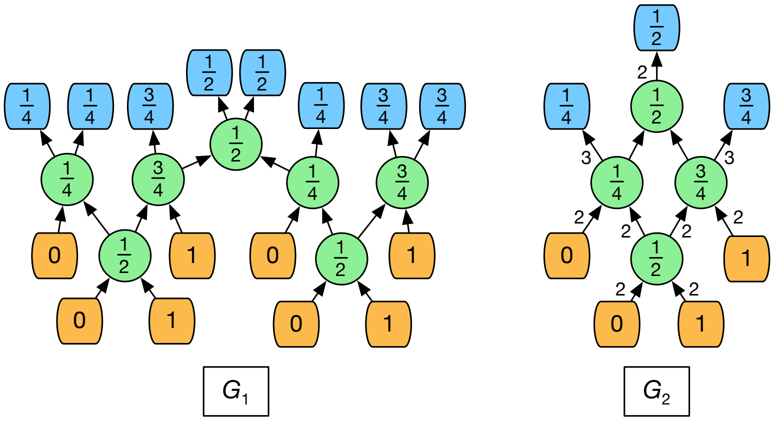

Figure 2.2: G_2 is a compact representation of G_1. All nodes in G_2 (except the last intermediate node with label $\frac{1}{2}$) represent an aggregation of at least two nodes from G_1.

2.2 Algorithm Description

In this section we describe our algorithm RPRIS for producing a single-droplet target of concentration t with precision $d = \mathsf{prec}(t)$. We first give the overall strategy and then we gradually explain its implementation. The core idea behind RPRIS is a recursive process that we refer to as *Recursive Precision Reduction*, that works (roughly) as follows. Let t_s be the concentration computed at the s^{th} recursive step with $d_s = \mathsf{prec}(t_s)$; initially, $t_0 = t$.

(rpr1) Replace t_s by another concentration value t_{s+1} with $d_{s+1} = d_s - 2$.

(rpr2) Recursively construct a mixing graph G_{s+1} for t_{s+1}.

(rpr3) Convert G_{s+1} into a mixing graph G_s for t_s, increasing waste by only one droplet.

The mixing graph produced by this process is G_0.

When we convert G_{s+1} into G_s in part (rpr3), the precision of the target increases by 2, but the waste only increases by 1, which gives us a rough bound of $d/2$ on the overall waste. However, the above process does not work for all concentration values; it only works when $t_0 \in [\frac{1}{4}, \frac{3}{4}]$. To deal with values outside this interval, we map t into t_0 so that $t_0 \in [\frac{1}{4}, \frac{3}{4}]$, next we apply Recursive Precision Reduction to t_0, and then we appropriately modify the computed mixing graph. This process is called *Initial Shift*.

We next describe these two processes in more detail, starting with Recursive Precision Reduction, followed by Initial Shift.

Recursive Precision Reduction (RPR). We start with concentration t_0 that, by applying Initial Shift (described next), we can assume to be in $[\frac{1}{4}, \frac{3}{4}]$.

Step (rpr1): computing t_{s+1}. We convert t_s into a carefully chosen concentration t_{s+1} for which $d_{s+1} = d_s - 2$. One key idea is to maintain an invariant so that at each recursive step, this new concentratiaon value t_{s+1} satisfies $t_{s+1} \in [\frac{1}{4}, \frac{3}{4}]$. To accomplish this, we consider five intervals $S_1 = [\frac{1}{8}, \frac{3}{8}]$, $S_2 = [\frac{1}{4}, \frac{1}{2}]$, $S_3 = [\frac{3}{8}, \frac{5}{8}]$, $S_4 = [\frac{1}{2}, \frac{3}{4}]$, and $S_5 = [\frac{5}{8}, \frac{7}{8}]$. Based on the value of t_s, we choose an interval S_k that contains t_s "in the middle". Specifically, we choose $S_k = [l, r]$ such that $t_s \in [l + \frac{1}{16}, r - \frac{1}{16}]$. We then compute $t_{s+1} = 4(t_s - l)$. Note that t_{s+1} satisfies both $t_{s+1} \in [\frac{1}{4}, \frac{3}{4}]$ (that is, our invariant) and $d_{s+1} = d_s - 2$. Figure 2.3 illustrates a graphical representation of intervals $S_1, S_2, \ldots, S_5$.

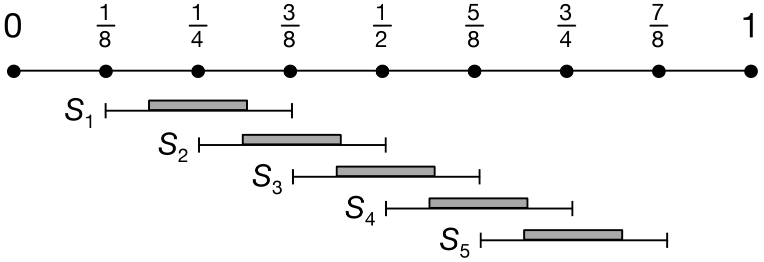

Figure 2.3: Graphical representation of intervals $S_1, S_2, \ldots, S_5$. The thick shaded part of each interval $S_k = [l, r]$ marks its "middle section" $[l + \frac{1}{16}, r - \frac{1}{16}]$. Each concentration within interval $[\frac{1}{4}, \frac{3}{4}]$ belongs to a middle section of some S_k.

Step (rpr3): converting G_{s+1} into G_s. Let G_{s+1} be the mixing graph obtained for t_{s+1} in step (rpr2), by invoking our procedure recursively. We modify G_{s+1} to obtain a graph G'_{s+1} which is then coupled with an appropriate converter C_{s+1}, to obtain mixing graph $G_s = G'_{s+1} \bullet C_{s+1}$. Figure 2.4 illustrates this conversion process.

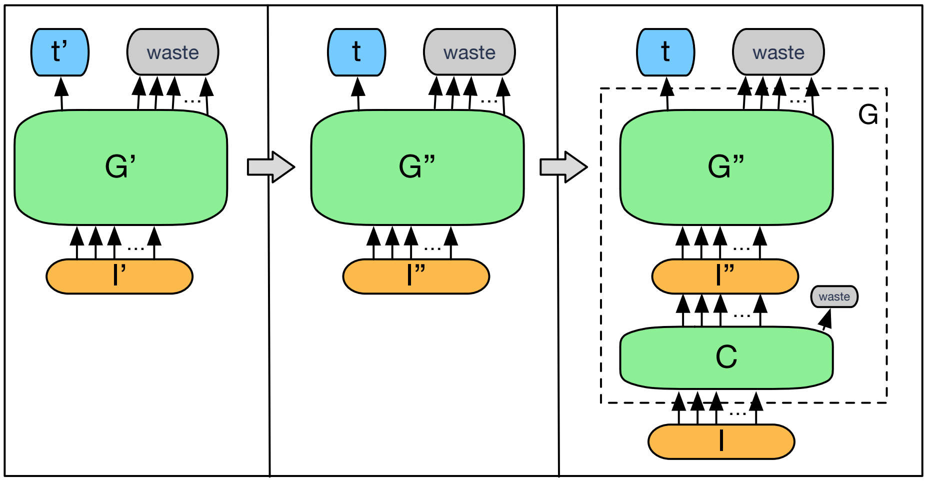

Figure 2.4: Conversion from G_{s+1} to G_s. The left image illustrates the computed mixing graph G_{s+1} with input labels I_{s+1} (consisting of only 0's and 1's) that produces t_{s+1} along with some waste. The middle figure illustrates G'_{s+1}, which is obtained from G_{s+1} by changing concentration labels. The last figure illustrates the complete mixing graph $G_s = G'_{s+1} \bullet C_{s+1}$ for t_s, shown within a dotted rectangle.

Next, we explain how to construct G'_{s+1}. G'_{s+1} consists of the same nodes and edges as G_{s+1}, only the concentration labels are changed. Specifically, every concentration label c from G_{s+1} is changed to $l + c/4$ in G'_{s+1}. Note that this is simply the inverse of the linear function that maps t_s to t_{s+1}. In particular, this will map the 0- and 1-labels of the source nodes in G_{s+1} to the endpoints l and r of the corresponding interval S_k.

The converter C_{s+1} used in G_s needs to have sink nodes with labels equal to the source nodes for G'_{s+1}. That is, if the labeling of the source nodes of G'_{s+1} is $I'_{s+1} = \{i : l, j : r\}$, then C_{s+1} will be an $(i : l, j : r)$-converter. As a general rule, C_{s+1} should produce at most one waste droplet, but there will be some exceptional cases where it produces two. (Nonetheless, we will show that at most one of such "bad" converter is used during the RPR process.) The construction of these converters is somewhat intricate, and is deferred to the next section.

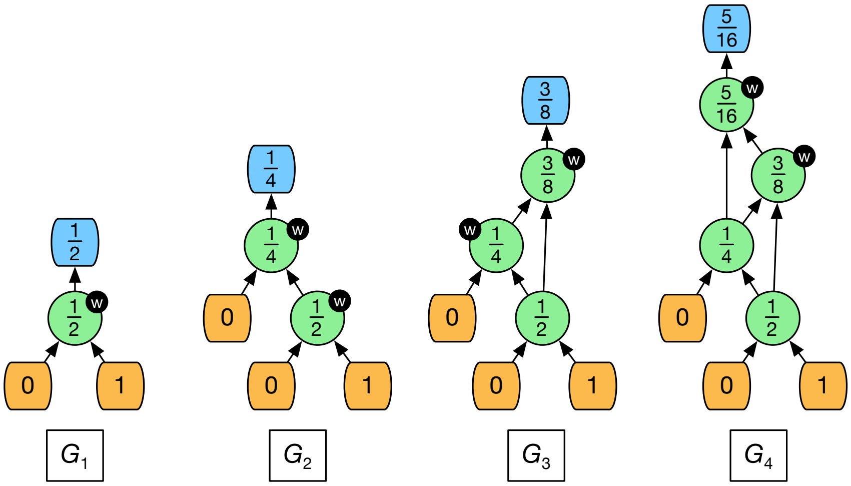

Figure 2.5: Base mixing graphs B_1, B_2, B_3 and B_4 for concentrations $\frac{1}{2}, \frac{1}{4}, \frac{3}{8}$ and $\frac{5}{16}$, respectively.

The base case. In the description of the RPR procedure above we tacitly assumed that $d_0 = \mathsf{prec}(t_0)$ is sufficiently large, so that we can apply the algorithm recursively to t_0. The base case for the recursion consists of only a few values for which we give explicit mixing graphs. Specifically, the base case involves concentrations $\mathcal{B} = \{\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{3}{8}, \frac{5}{8}, \frac{5}{16}, \frac{11}{16}\}$. (Concentrations $\frac{5}{16}$ and $\frac{11}{16}$ are not strictly necessary for correctness but are included in the base case to improve the waste bound.) Figure 2.5 illustrates the base mixing graphs for concentrations $\frac{1}{2}$, $\frac{1}{4}$, $\frac{3}{8}$, and $\frac{5}{16}$; the mixing graphs for the remaining concentrations are symmetric.

Initial Shift (IS). We now describe the IS procedure. At the fundamental level, the idea is similar to a single step of RPR, although the involved linear mappings and the converter are significantly different.

We can assume that $t < \frac{1}{4}$ (because for $t > \frac{3}{4}$ the process is symmetric). Thus the binary representation of t starts with $\gamma \geq 2$ fractional 0's. Since $2^{\gamma-1}t \in [\frac{1}{4}, \frac{1}{2})$, we could use this value as the result of the initial shift, but to improve the waste bound we refine this choice as follows: If $2^{\gamma-1}t \in (\frac{3}{8}, \frac{1}{2})$ then let $t_0 = 2^{\gamma-1}t$ and $\sigma = 1$. Otherwise, we have $2^{\gamma-1}t \in [\frac{1}{4}, \frac{3}{8}]$, in which case we let $t_0 = 2^{\gamma}t$ and $\sigma = 0$. In either case, $t_0 = 2^{\gamma-\sigma}t \in [\frac{1}{4}, \frac{3}{4}]$ and $d_0 = d - \gamma + \sigma$.

Let G_0 be the mixing graph obtained by applying the RPR process to t_0. It remains to show how to modify G_0 to obtain the mixing graph G for t. This is analogous to the process shown in Figure 2.4. We first construct a mixing graph G'_0 that consists of the same nodes and edges as G_0, only each concentration label c is replaced by $c/2^{\gamma-\sigma}$. In particular, the label set of the source nodes in G'_0 will have the form $I'_0 = \{i : 0, j : 1/2^{\gamma-\sigma}\}$.

We then construct a $(i : 0, j : 1/2^{\gamma-\sigma})$-converter C_0 and couple it with G'_0 to obtain G; that is, $G = G'_0 \bullet C_0$. This C_0 is easy to construct: The 0's don't require any mixing, and to produce the j droplets $1/2^{\gamma-\sigma}$ we start with one droplet 1 and repeatedly mix it with 0's, making sure to generate at most one waste droplet at each step. More specifically, after z steps we will have j_z droplets with concentration $1/2^z$, where $j_z = \lceil j/2^{\gamma-\sigma-z} \rceil$. In step z, mix these j_z droplets with j_z 0's, producing $2j_z$ droplets with concentration $1/2^{z+1}$. We then either have $j_{z+1} = 2j_z$, in which case there is no waste, or $j_{z+1} = 2j_z - 1$, in which case one waste droplet $1/2^{z+1}$ is produced. Overall, C_0 produces at most $\gamma - \sigma$ waste droplets.

2.3 Construction of Converters

In this section we detail the construction of our converters. Let t_s denote the concentration at the s^{th} recursive step in the RPR process. We can assume that $t_s \in [\frac{1}{4}, \frac{1}{2}]$, because the case $t_s \in (\frac{1}{2}, \frac{3}{4}]$ is symmetric. Recall that for a t_s in this range, in Step (rpr1) we will chose an appropriate interval S_k, for some $k \in \{1, 2, 3\}$. Let $S_k = [l, r]$ (that is, $l = k \cdot \frac{1}{8}$ and $r = l + \frac{1}{4}$). For each such k and all $i, j \geq 1$ we give a construction of an $(i : l, j : r)$-converter that we will denote $C^k_{i,j}$. Our main objective here is to design these converters so that they produce as little waste as possible — ideally none.

2.3.1 $(i : \frac{1}{4}, j : \frac{1}{2})$-Converters $C^2_{i,j}$

We start with the case $k = 2$, because in this case the construction is relatively simple. We show how to construct, for all $i, j \geq 1$, our $(i : \frac{1}{4}, j : \frac{1}{2})$-converter $C^2_{i,j}$ that produces at most one droplet of waste. These converters are constructed via an iterative process. We first give initial converters $C^2_{i,j}$, for some small values of i and j, by providing specific graphs. All other converters are obtained from these initial converters by repeatedly coupling them with other mixing graphs that we refer to as *extenders*.

Let $J^2_{\text{init}} = \{(i,j)\}_{i,j\in\{1,2\}}$. The initial converters $C^2_{i,j}$ are defined for the four index pairs $(i,j) \in J^2_{\text{init}}$. Figure 2.6 illustrates the initial converters $C^2_{2,1}, C^2_{1,2}$ and two extenders X^2_1, X^2_2. Converter $C^2_{1,2}$ produces one waste droplet and converter $C^2_{2,1}$ does not produce any waste. Converter $C^2_{1,1}$ can be obtained from $C^2_{2,1}$ by designating one of the $\frac{1}{4}$ outputs as waste. Converter $C^2_{2,2}$ is defined as $C^2_{2,2} = X^2_1 \bullet C^2_{1,1}$, and produces one waste droplet of $\frac{1}{2}$. (Note that, by the definition of coupling, $C^2_{2,2}$ is simply a disjoint union of $C^2_{1,1}$ and X^2_1.)

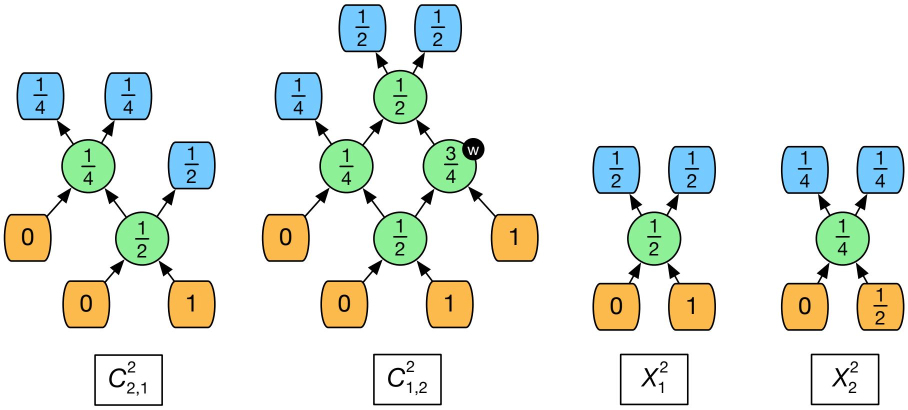

Figure 2.6: Initial converters and extenders for the case $I = \left\{i : \frac{1}{4}, j : \frac{1}{2}\right\}$.

The construction of other converters $C^2_{i,j}$ is based on the following observation: Suppose that we already have constructed some $C^2_{i,j}$. Then (i) $X^2_1 \bullet C^2_{i,j}$ is a $C^2_{i,j+2}$ converter that produces the same waste as $C^2_{i,j}$, and (ii) provided that $j \geq 2$, $X^2_2 \bullet C^2_{i,j}$ is a $C^2_{i+2,j-1}$ converter that produces the same waste as $C^2_{i,j}$.

Let now $i, j \geq 1$ with $(i,j) \notin J^2_{\text{init}}$ be arbitrary. To construct $C^2_{i,j}$, using the initial converters and the above observation, express the integer vector (i,j) as $(i,j) =$

$(i',j')+\phi(0,2)+\psi(2,-1)$, for some $i',j' \in J^2_{\text{init}}$ and integers $\psi = \lceil \frac{i}{2} \rceil - 1$ and $\phi = \lceil \frac{j+\psi}{2} \rceil - 1$. Then $C^2_{i,j}$ is constructed by starting with $C^2_{i',j'}$ and coupling it ϕ times with X^2_1 and then ψ times with X^2_2. (This order of coupling is not unique but is also not arbitrary, because each extender X^2_2 requires a droplet of concentration $\frac{1}{2}$ as input.) Since X^2_1 and X^2_2 do not produce waste, $C^2_{i,j}$ will produce at most one waste droplet.

2.3.2 $(i : \frac{3}{8}, j : \frac{5}{8})$-Converters $C^3_{i,j}$

Next, for each pair $i, j \geq 1$ we construct an $(i : \frac{3}{8}, j : \frac{5}{8})$-converter $C^3_{i,j}$. These converters are designed to produce one droplet of waste. ($C^3_{1,1}$ will be an exception, see the discussion below). Our approach follows the scheme from Section 2.3.1: we start with some initial converters, which then can be repeatedly coupled with appropriate extenders to produce all other converters. Since concentrations $\frac{3}{8}$ and $\frac{5}{8}$ are symmetric (as $\frac{5}{8} = 1 - \frac{3}{8}$), we will only show the construction of converters $C^3_{i,j}$ for $i \geq j$; the remaining converters can be computed using symmetric mixing graphs.

Let $J^3_{\text{init}} = \{(i,1)\}_{i \in \{1,2,\ldots,9\}} \cup \{(2,2)\}$. The initial converters $C^2_{i,j}$ are defined for all index pairs $(i,j) \in J^3_{\text{init}}$. Figure 2.7 shows converters $C^3_{3,1}, C^3_{4,1}, \ldots, C^3_{7,1}$ and $C^3_{9,1}$. Converter $C^1_{8,1}$ can be obtained from $C^3_{9,1}$ by designating an output of $\frac{3}{8}$ as waste. Converter $C^3_{2,2}$ is almost identical to X^3_1 in Figure 2.8; except that the source labels $\frac{3}{8}$ and $\frac{5}{8}$ are replaced by 0 and 1, respectively (the result of mixing is still $\frac{1}{2}$, so other concentrations in the graph are not affected). Converters $C^3_{1,1}$ and $C^3_{2,1}$ are obtained from $C^3_{2,2}$ by designating outputs of $\{\frac{3}{8}, \frac{5}{8}\}$ and $\frac{5}{8}$, respectively, as waste. Note that all initial converters except for $C^3_{1,1}$ produce at most one droplet of waste.

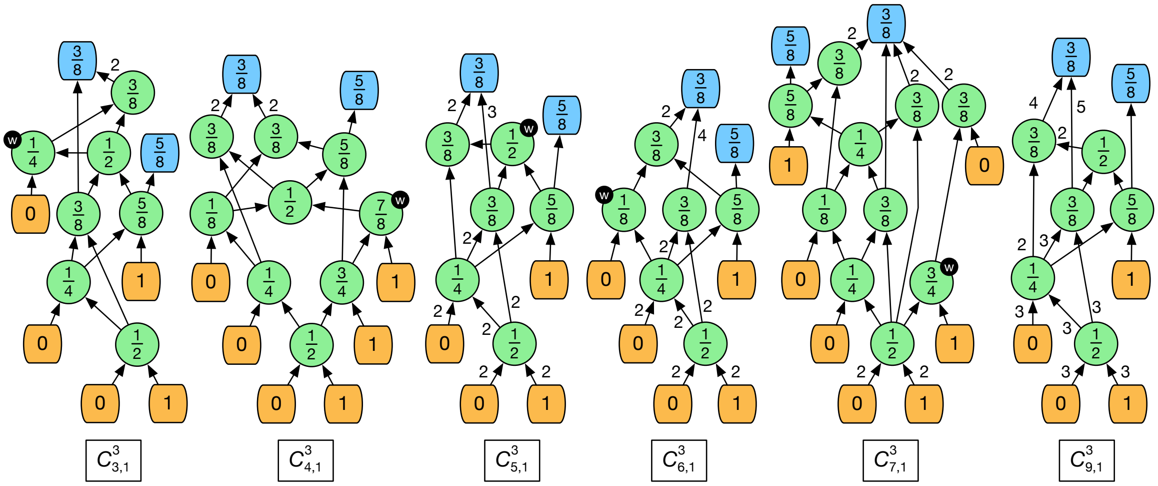

Figure 2.7: Initial converters for the case $I = \{i : \frac{3}{8}, j : \frac{5}{8}\}$.

Now, consider extenders X^3_1 and X^3_2 in Figure 2.8. The construction of other converters $C^3_{i,j}$ follows the next observation: Assume that we have already constructed some $C^3_{i,j}$, with $i \geq j$. Then (i) $X^3_1 \bullet C^3_{i,j}$ is a $C^3_{i+1,j+1}$ converter that produces the same waste as $C^3_{i,j}$, and (ii) $X^3_2 \bullet C^3_{i,j}$ is a $C^3_{i+8,j}$ converter that produces the same waste as $C^3_{i,j}$.

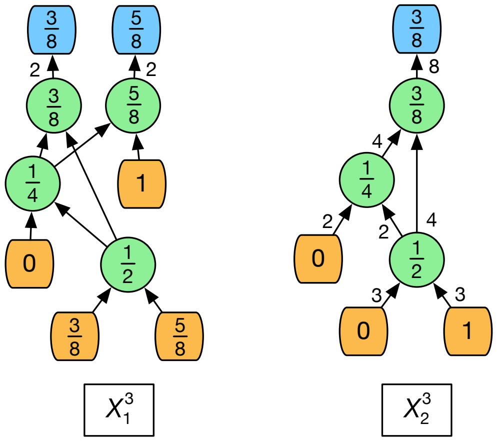

Figure 2.8: X^3_1 and X^3_2 extenders for the case $I = \{i : \frac{3}{8}, j : \frac{5}{8}\}$.

Consider now arbitrary $i \geq j \geq 1$ with $(i,j) \notin J^3_{\text{init}}$. To construct $C^3_{i,j}$, using the initial converters and the above observation, express the integer vector (i,j) as $(i,j) = (i',j') + \phi(1,1) + \psi(8,0)$, for some integers $\phi, \psi \geq 0$, and $(i',j') \in J^3_{\text{init}} - \{(1,1)\}$. Then $C^3_{i,j}$ is constructed by starting with $C^3_{i',j'}$ and coupling it ϕ times with X^3_1 and then ψ times with X^3_2 (in arbitrary order). Since X^3_1 and X^3_2 do not produce waste (and we do not use the initial converter $C^3_{1,1}$), $C^3_{i,j}$ will produce at most one waste droplet.

Overall, all converters $C^3_{i,j}$, except for $C^3_{1,1}$ produce at most one waste droplet. Converter $C^3_{1,1}$ produces two droplets of waste; however, as we later show in Section 2.4, it is not actually used in the algorithm.

2.3.3 $(i : \frac{1}{8}, j : \frac{3}{8})$-Converters $C^1_{i,j}$

In this section, for each pair $i,j \geq 1$ we construct an $(i : \frac{1}{8}, j : \frac{3}{8})$-converter $C^1_{i,j}$. Most of these converters produce at most one droplet of waste, but there will be four exceptional coverters with waste two. (See the comments at the end of this section.) The idea of the construction follows the same scheme as in Sections 2.3.1 and 2.3.2: we start with some initial converters and repeatedly couple them with appropriate extenders to obtain other converters.

Let $J^1_{\text{init}} = \{(i,j)\}_{i,j\in\{1,2,3\}} \cup \{(4,2),(2,5)\}$. The initial converters $C^1_{i,j}$ are defined for all index pairs $(i,j) \in J^1_{\text{init}}$. Converters $C^1_{2,2}$, $C^1_{2,3}$, $C^1_{2,5}$, $C^1_{3,1}$, $C^1_{3,3}$ and $C^1_{4,2}$ are shown in Figure 2.9. Converters $C^1_{1,1}$, $C^1_{1,2}$ and $C^1_{2,1}$ are obtained from $C^1_{2,2}$ by designating outputs of $\{\frac{1}{8}, \frac{3}{8}\}$, $\frac{1}{8}$ and $\frac{3}{8}$, respectively, as waste. Converter $C^1_{1,3}$ is obtained from $C^1_{2,3}$ by designating an output of $\frac{1}{8}$ as waste, and $C^1_{3,2}$ is obtained from $C^1_{4,2}$ by designating an output of $\frac{1}{8}$ as

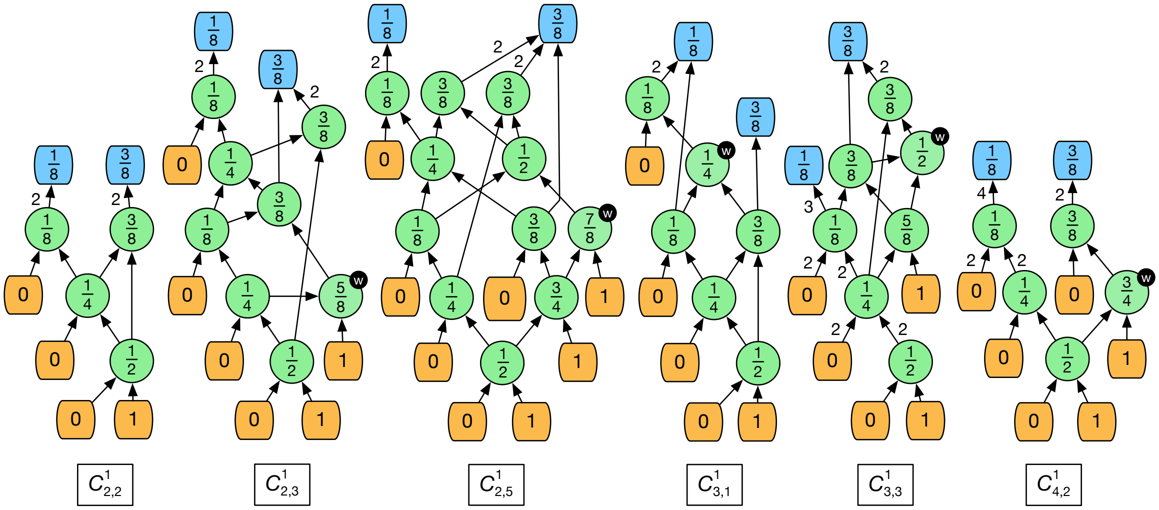

Figure 2.9: Initial converters for the case $I = \{i : \frac{1}{8}, j : \frac{3}{8}\}$.

waste. Thus, among the initial converters, $C^1_{1,1}$, $C^1_{1,3}$ and $C^1_{3,2}$ each produces two droplets of waste; all other converters have at most one droplet of waste.

Next, we provide an observation leading to the construction of other converters $C^1_{i,j}$. Consider extenders X^1_1 and X^1_2 in Figure 2.10 and assume that we have already constructed some $C^1_{i,j}$. Then, (i) provided that $j \geq 2$, $X^1_1 \bullet C^1_{i,j}$ is a $C^1_{i+3,j-1}$ converter that produces the same waste as $C^1_{i,j}$, and (ii) provided that $i \geq 2$, $X^1_2 \bullet C^1_{i,j}$ is a $C^1_{i-1,j+3}$ converter that produces the same waste as $C^1_{i,j}$. We also need the following, less obvious observation:

Observation 1 *If $i, j \geq 1$ and $(i,j) \notin J^1_{init} \cup \{(6,1)\}$, then $(i,j) = (i',j') + \phi(-1,3) + \psi(3,-1)$, for some integers $\phi, \psi \geq 0$, and $(i',j') \in J^1_{init} - \{(1,1),(1,3),(3,2)\}$.*

Proof. Let $i, j \geq 1$ and $(i,j) \notin J^1_{\text{init}} \cup \{(6,1)\}$. We note first that we can represent (i,j) as $(i,j) = (\tilde{i},\tilde{j}) + \tilde{\phi}(-1,3) + \tilde{\psi}(3,-1)$, for $(\tilde{i},\tilde{j}) \in J^1_{\text{init}} - \{(2,5),(4,2)\}$ and integers $\tilde{\phi}, \tilde{\psi} \geq 0$.

If $(\tilde{i}, \tilde{j}) \notin \{(1,1),(1,3),(3,2)\}$ then we are done. Otherwise, we show how to modify the values of parameters $\tilde{i}$, $\tilde{j}$, $\tilde{\phi}$ and $\tilde{\psi}$ so that they satisfy the condition in the observation.

<u>Case 1</u>: $(\tilde{i}, \tilde{j}) = (1,1)$. For this case, $\tilde{\phi}, \tilde{\psi} \geq 1$ must hold, as otherwise we would get a contradiction with $i, j \geq 1$. Therefore, we can write (i,j) as $(i,j) = (3,3) + (\tilde{\phi} - 1)(-1,3) + (\tilde{\psi} - 1)(3,-1)$.

<u>Case 2</u>: $(\tilde{i}, \tilde{j}) = (1,3)$. For this case, $\tilde{\psi} \geq 1$ must hold, because $i \geq 1$. Therefore, we can write (i,j) as $(i,j) = (4,2) + \tilde{\phi}(-1,3) + (\tilde{\psi} - 1)(3,-1)$.

<u>Case 3</u>:$(\tilde{i}, \tilde{j}) = (3,2)$. For this case, it is sufficient to prove that $\tilde{\phi} \geq 1$, since we could then write (i,j) as $(i,j) = (2,5) + (\tilde{\phi} - 1)(-1,3) + \tilde{\psi}(3,-1)$. To show that $\tilde{\phi} \geq 1$ we argue by contradiction, as follows. Suppose that $\tilde{\phi} = 0$. Then $(i,j) = (3,2) + \tilde{\psi}(3,-1)$. For $\tilde{\psi} \in \{0,1\}$ this contradicts that $(i,j) \notin J^1_{\text{init}} \cup \{(6,1)\}$, and for $\tilde{\psi} \geq 2$ it contradicts that $j \geq 1$. ■

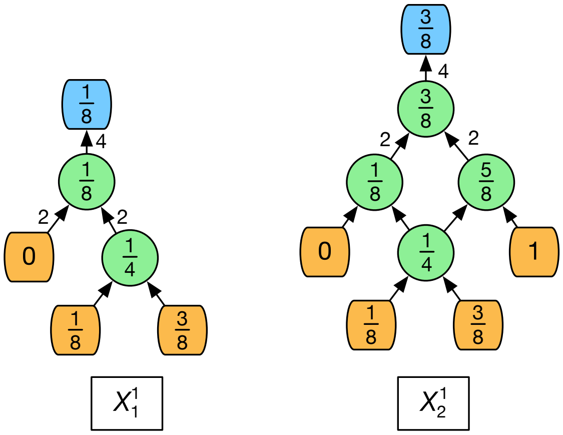

Figure 2.10: X_1^1 and X_2^1 extenders for the case $I = \{i : \frac{1}{8}, j : \frac{3}{8}\}$.

Using the observations above, for any pairs $i, j \geq 1$ we can construct converter $C^1_{i,j}$ as follows. If $(i,j) = (6,1)$ we let $C^1_{6,1} = X^1_1 \bullet C^1_{3,2}$ (so $C^1_{6,1}$ has two droplets of waste). If $(i,j) \neq (6,1)$, we construct $C^1_{i,j}$ by starting with $C^1_{i',j'}$ and repeatedly coupling it with ϕ copies of X^1_2 and ψ copies of X^1_1, choosing a suitable order of couplings to ensure that each intermediate converter has at least one output $\frac{1}{8}$ and at least one $\frac{3}{8}$. (For example, if $j' = 1$ then we begin by coupling X^1_2 first.) As X^1_1 and X^1_2 do not produce any waste, these $C^1_{i,j}$'s will each produce at most one droplet of waste.

Overall, the converters $C^1_{i,j}$ we construct have at most one droplet of waste, with the exception of the following four: $C^1_{1,1}$, $C^1_{1,3}$, $C^1_{3,2}$ and $C^1_{1,6}$. (It is easy to prove that for these converters waste 2 cannot be avoided.) As we show later in Section 2.4, of these four converters only $C^1_{1,3}$ is actually used in the RPR process of Algorithm RPRIS, and it is used at most once.

2.4 Performance Bounds

In this section we provide the analysis of Algorithm RPRIS, including the worst-case bound on produced waste, a bound on the size of computed mixing graphs, and the running time.

Bound on waste. We first estimate the number of waste droplet of Algorithm RPRIS. Let G be the mixing graph constructed by RPRIS for a target concentration t with its corresponding values $d = \mathsf{prec}(t)$ and γ (as defined at the beginning of the chapter). Below we prove the following theorem.

Theorem 2 *The number of waste droplets in G is at most $\frac{1}{2}(d+\gamma)+2$.*

To prove Theorem 2, we show that the total number of sink nodes in G is at most $\frac{1}{2}(d+\gamma-\sigma)+3$, for corresponding $\sigma \in \{0,1\}$. (This is sufficient, as one sink node is used to produce t).

Following the algorithm description in Section 2.2, let $G = G'_0 \bullet C_0$. From our construction of C_0 (at the end of Section 2.2), we get that C_0 contributes at most $\gamma-\sigma$ sink nodes to G. (Each waste droplet produced by C_0 represents a sink node in G.) Therefore, to prove Theorem 2 it remains to show that G'_0 contains at most $\frac{1}{2}(d-\gamma+\sigma)+3$ sink nodes. This is equivalent to showing that G_0, computed by process RPR for t_0 (and used to compute G'_0), contains at most $\frac{1}{2}d_0+3$ sink nodes, where $d_0 = \mathsf{prec}(t_0) = d-\gamma+\sigma$. Lemma 3 next proves this claim.

Lemma 3 *The number of sink nodes in G_0 is at most $\frac{1}{2}d_0+3$.*

Proof. Let t_b be the concentration used for the base case of the RPR process and $d_b = \mathsf{prec}(t_b) \leq d_0$ its precision. We prove the lemma in three steps. First, we show that (i) the number of sink nodes in the mixing graph computed for t_b is at most three. (In particular, this gives us that the lemma holds if $t_0 = t_b$.) Then, we show that (ii) if $t_0 \neq t_b$ then the number of converters used in the construction of G_0 is no more than $\frac{1}{2}d_0-1$, and (iii) that at most one of such converter contains two waste sink nodes. All sink nodes of G_0 are either in its base-case graph or in its converters, so combining claims (i), (ii) and (iii) gives a complete proof for Lemma 3.

The proof of (i) is by straightforward inspection. By definition of the base case, $t_b \in \mathcal{B} = \left\{\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{3}{8}, \frac{5}{8}, \frac{5}{16}, \frac{11}{16}\right\}$. The mixing graphs for base concentrations are shown in

Figure 2.5. (The graphs for $\frac{3}{4}$, $\frac{5}{8}$, and $\frac{11}{16}$ are symmetric to B_2, B_3, and B_4.) All these graphs have at most 3 sink nodes.

Next, we prove part (ii). In each step of the RPR process we reduce the precision of the target concentration by 2 until we reach the base case, which gives us that the number of converters is exactly $\frac{1}{2}(d_0 - d_b)$. It is thus sufficient to show that $d_b \geq 2$, as this immediately implies (ii). Indeed, the assumption that $t_0 \neq t_b$ and the definition of the base case implies that $d_0 \geq 4$. (This is because the algorithm maintains the invariant that its target concentration is in $[\frac{1}{4}, \frac{3}{4}]$ and all concentrations in this interval with precision at most 3 are in $\mathcal{B}$.) This, and the precision of the target concentration decreasing by exactly 2 in each step of the recursion, imply that $d_b \in \{2, 3\}$ holds.

We now address part (iii). First we observe that converters $C^k_{1,1}$ are not used in the construction of G_0: If we did use $C^k_{1,1}$ in the construction of G_0 then the source labels for the next recursive step are $\{0, 1\}$. Hence, $t_b = \frac{1}{2}$. Now, let t_{b-1} be the concentration, and $S_k = [l, r]$ the interval, used to compute t_b. Since $t_b = \frac{1}{2}$, then $t_{b-1} = \frac{1}{2}(l + r)$. Therefore, by definition of S_k, $t_{b-1} \in \left\{\frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}\right\} \subset \mathcal{B}$, so Algorithm `RPRIS` would actually use a base case mixing graph for t_{b-1}, instead of constructing $C^k_{1,1}$ for t_b.

So, it is sufficient to consider $C^k_{i,j}$ converters that satisfy $i + j \geq 3$ with $i, j \geq 1$. Now, from Sections 2.3.1, 2.3.2 and 2.3.3, we observe that the only such converters that contain two waste sink nodes are $C^1_{1,3}, C^1_{3,2}$ and $C^1_{6,1}$. Claim 4 below shows that converters $C^1_{6,1}$ and $C^1_{3,2}$ are not used in the construction of G_0.

Regarding $C^1_{1,3}$, first we note that this converter has exactly six source nodes; see Figure 2.9, Section 2.3.3. This implies that $C^1_{1,3}$ can not be used more than once in the construction of G_0, since the number of source nodes at each recursive step in the RPR process is decreasing. (Note that there are symmetric converters $C^5_{3,1}$, $C^5_{2,3}$ and $C^5_{1,6}$ for $C^1_{1,3}$, $C^1_{3,2}$ and $C^1_{6,1}$, respectively, where superscript 5 is associated to interval S_5. Nevertheless, a similar argument holds.) Thus, step (iii) holds.

Claim 4 *Converters $C^1_{6,1}$ and $C^1_{3,2}$ are not used by Algorithm `RPRIS` in the construction of G_0 for t_0.*

We first present the following observations. Consider recursive step s of the RPR process, for which t_s is the target concentration. If a converter $C^1_{i,j}$ is used in this step, then $t_s \in (\frac{1}{4}, \frac{5}{16}]$ must hold; that is t_s is in the middle part of interval S_1 (see Figure 2.3 in Section 2.2). (Recall that, by our algorithm's invariant, $t_s \in [\frac{1}{4}, \frac{3}{4}]$. Also, note that $t_s \neq \frac{1}{4}$ since otherwise this would be a base case and the algorithm would use B_2 from Figure 2.5 instead.) Further, at the next step of the RPR process, $t_{s+1} = 4(t_s - \frac{1}{8})$ satisfies $t_{s+1} \in (\frac{1}{2}, \frac{3}{4}]$.

We now prove the claim by contradiction, using the above observations. Assume that either $C^1_{6,1}$ or $C^1_{3,2}$ were used in the construction of G_0. If $C^1_{6,1}$ was used in the construction of G_0, then the concentration labels of the source nodes at the next recursive step are $\{6 : 0, 1\}$, and thus, since $t_{s+1} > \frac{1}{2}$, there is not enough reactant available to produce t_{s+1}.

On the other hand, if $C^1_{3,2}$ was used in the construction of G_0, then the concentration labels of the source nodes at the next recursive step are $\{3 : 0, 2 : 1\}$. This implies that the next step is guaranteed not to be a base case, since all mixing graphs used for base case concentrations contain at most three source nodes, as illustrated in Figure 2.5. Now, as $t_{s+1} > \frac{1}{2}$, depending on the exact value of t_{s+1}, the chosen interval for t_{s+1} must be either $S_3 = [\frac{3}{8}, \frac{5}{8}]$, $S_4 = [\frac{1}{2}, \frac{3}{4}]$ or $S_5 = [\frac{5}{8}, \frac{7}{8}]$. We now consider these three cases.

Case 1: $t_{s+1} \in (\frac{1}{2}, \frac{9}{16}]$. Then the chosen interval is $S_3 = [\frac{3}{8}, \frac{5}{8}]$. The only $C^3_{i,j}$ converter with source concentration labels $\{3 : 0, 2 : 1\}$ is $C^3_{3,1}$ (see in Figure 2.7 in Section 2.3.2), whose sink nodes have concentration labels $\{\frac{1}{4}, 3 : \frac{3}{8}, \frac{5}{8}\}$. Therefore, the input configuration for the next recursive step will be a subset of $\{3 : 0, 1\}$, which does not have enough reactant to produce $4(t_{s+1} - \frac{3}{8}) > \frac{1}{2}$, thus contradicting the choice of S_3.

Case 2: $t_{s+1} \in (\frac{9}{16}, \frac{11}{16}]$. Then the chosen interval is $S_4 = [\frac{1}{2}, \frac{3}{4}]$. This instance is symmetric to interval S_2, having source concentration labels $\{2 : 0, 3 : 1\}$, instead of $\{3 : 0, 2 : 1\}$, and target concentration $t'_{s+1} = (1 - t_{s+1})$. Thus we proceed accordingly. Since every converter and extender in Section 2.3.1 adds at least the same number of source nodes with concentration label 0 as source nodes with concentration label 1, then no converter constructed by the algorithm will have source concentration labels $\{2 : 0, 3 : 1\}$. Hence, we have a contradiction with the choice of S_2 for t'_{s+1}, and thus also with the choice of S_4 for t_{s+1}.

Case 3: $t_{s+1} \in (\frac{11}{16}, \frac{3}{4}]$. Then the chosen interval is $S_5 = [\frac{5}{8}, \frac{7}{8}]$. The argument here is simple: to produce concentration $\frac{7}{8}$, at least three reactant droplets are needed, but the input configuration contains only two. Therefore, at the next recursive step, the algorithm will not have enough reactant droplets to construct a converter $C^5_{i,j}$ with $i, j \geq 1$, contradicting the choice of S_5 for t_{s+1}.

Finally, neither S_3, S_4 nor S_5 are chosen by our algorithm for t_{s+1}, contradicting $C^1_{3,2}$ being used for the construction of G_0. This completes the proof of Claim 4 and Lemma 3 (thus also completing the proof of Theorem 2). ■

Size of mixing graphs and running time. Let $G = G'_0 \bullet C_0$ be the mixing graph computed by Algorithm `RPRIS` for t; C_0 is constructed by process IS while G'_0 is obtained from G_0 (constructed by process RPR) by changing concentration labels appropriately. We claim that the running time of Algorithm RPRIS is $O(|G|)$, and that the size of G is $O(d^2)$, for $d = \mathsf{prec}(t)$. We give bounds for G_0 and C_0 individually, then we combine them to obtain the claimed bounds. (This is sufficient because the size of G'_0, as well as the running time to construct it, is asymptotically the same as that for G_0.)

First, following the description of process RPR in Section 2.2, suppose that at recursive step s, G_{s+1}, G'_{s+1} and converter $C_{s+1} = C^k_{i,j}$ are computed. (Note that the algorithm does not need to explicitly relabel G_{s+1} to get G'_{s+1} – we only distinguish G_{s+1} from G'_{s+1} for the purpose of presentation.) The size of $C^k_{i,j}$ is $O(i + j)$ and it takes time $O(i+j)$ to assemble it (as the number of required extenders is $O(i+j)$). Coupling C_{s+1} with G'_{s+1} also takes time $O(i + j)$, since I'_{s+1} (the input configuration for G'_{s+1}) has cardinality $O(i+j)$ as well. In other words, the running time of each recursive RPR step is proportional to the number of added nodes. Thus the overall running time to construct G_0 is $O(|G_0|)$.

Now, let t_0 be the target concentration for the RPR process, with $d_0 = \mathsf{prec}(t_0)$. Then, the size of G_0 is $O(d_0^2)$. This is because the depth of recursion in the RPR process is $O(d_0)$, and each converter used in this process has size $O(d_0)$ as well. The reason for this bound on the converter size is that, from a level of recursion to the next, the number of source nodes increases by at most one (with an exception of at most one step, as explained earlier in this section), and the size of a converter $C^k_{i,j}$ used at this level is asymptotically the same as the number of source nodes at this level. (I_s and I'_{s+1} in Figure 2.4 illustrate the idea.)

Regarding the bounds for C_0, we first argue that the running time to construct C_0 is $O(|C_0|)$. This follows from the construction given in Section 2.2; on step s there are

$2j_s$ droplets being mixed, which requires j_s nodes; thus the entire step takes time $O(j_s)$. We next show that the size of C_0 is $O(d^2)$. Let I_0 be the input configuration for G_0. From the analysis for G_0, we get that $|I_0| = O(d_0)$, so the last step in C_0 contains $O(d_0)$ nodes. Therefore, as the depth of C_0 is $\gamma - \sigma$, the size of C_0 is $O(\gamma d_0) = O(d_0^2)$.

Combining the bounds from G_0 and C_0, we get that the running time of Algorithm `RPRIS` is $O(|G|)$ and the size of G is $O(d^2)$. (The coupling of C_0 with G'_0 does not affect the overall running time, since it takes $O(d_0)$ time to couple them, as $|I_0| = O(d_0)$.)

2.5 Experimental Study

In this section we compare the performance of our algorithm with algorithms `Min-Mix`, `DMRW`, `REMIA` and `ESFM`. These algorithms were described in Section 1.2. However, for completeness, we still give brief descriptions of these algorithms so that the reader reviews the intuitions behind different approaches for constructing mixing graphs. Let $t \in (0, 1)$ be the target concentration and $d = \mathsf{prec}(t)$ its precision. Also, let $\mathsf{bin}(t)$ be t's binary representation with no trailing zeros.

`Min-Mix` **[65]:** This algorithm is very simple. It starts with $\tau = 0$ and mixes it with the bits of $\mathsf{bin}(t)$ in reverse order, ending with $\tau = t$. It runs in time $O(d)$ and produces d droplets of waste.

`DMRW` **[50]:** This algorithm is based on binary search. Starting with pivot values $l = 0$ and $r = 1$, the algorithm repeatedly "mixes" l and r and resets one of them to their average $\frac{1}{2}(l + r)$, maintaining the invariant that $t \in [l, r]$. After d steps we end up with $l = r = t$. Then the algorithm gradually backtracks to determine, for each intermediate pivot value, how many times this value was used in mixing, and based on this information it computes the required number of droplets. This information is then converted into a mixing graph.

REMIA [26]: This algorithm is based on two phases. In the first phase, the algorithm computes a mixing graph G' whose source nodes have concentration labels that have exactly one bit 1 in their binary representation; each such concentration represents each of the 1 bits in $\mathsf{bin}(t)$. Then, in the second phase, a mixing graph G'' (that minimizes reactant usage), whose sink nodes are basically a superset of the source nodes in G', is computed. Finally, G for t is obtained as $G' \bullet G''$. (Although REMIA targets reactant usage, its comparison to different algorithms in terms of total waste was also reported in [26]. Thus, for the sake of completeness, we included REMIA in our study.)

ESFM [7]: This algorithm constructs a “universal” mixing graph that contains all mixing graphs of depth d as subgraphs. It then formulates the problem of computing a mixing graph minimizing waste as an integer linear program (a restricted flow problem), and solves this program. This universal graph has size exponential in d, and thus the overall running time is doubly exponential in d.

We now present the results of our experiments. Each experiment consisted on generating all concentration values with precision d, for $d \in \{7, 8, 15, 20\}$, and comparing the outputs of each of the algorithms. The results for ESFM are shown only for $d \in \{7, 8\}$, since for $d \in \{15, 20\}$ the running time of ESFM is prohibitive.

Figure 2.11 illustrates the experiments for concentrations of precision 7 and 8. Figure 2.12 illustrates the experiments for concentrations of precision 15 and 20. In both figures, the data was smoothed using MATLAB's *smooth* function to reduce clutter and to bring out the differences in performance between different algorithms.

As can be seen from these graphs, RPRIS significantly outperforms algorithm Min-Mix, REMIA and DMRW: It produces on average about 50% less waste than Min-Mix (consistently with our bound of $\frac{1}{2}(d + \gamma) + 4$ on waste produced by RPRIS), and 40% less waste than REMIA. It also produces on average between 21 and 25% less waste than DMRW, with this percentage increasing with d. Additionally, when compared to ESFM, RPRIS produces on average only about 7% additional waste for $d = 7, 8$.

Among all of the target concentration values used in our experiments, there is not a single case where RPRIS is worse than either Min-Mix or REMIA. When compared to DMRW, RPRIS never produces more waste for precision 7 and 8. For precision 15, the percentage of concentrations where RPRIS produces more waste than DMRW is below 2%, and for precision 20 it is below 3.5%.

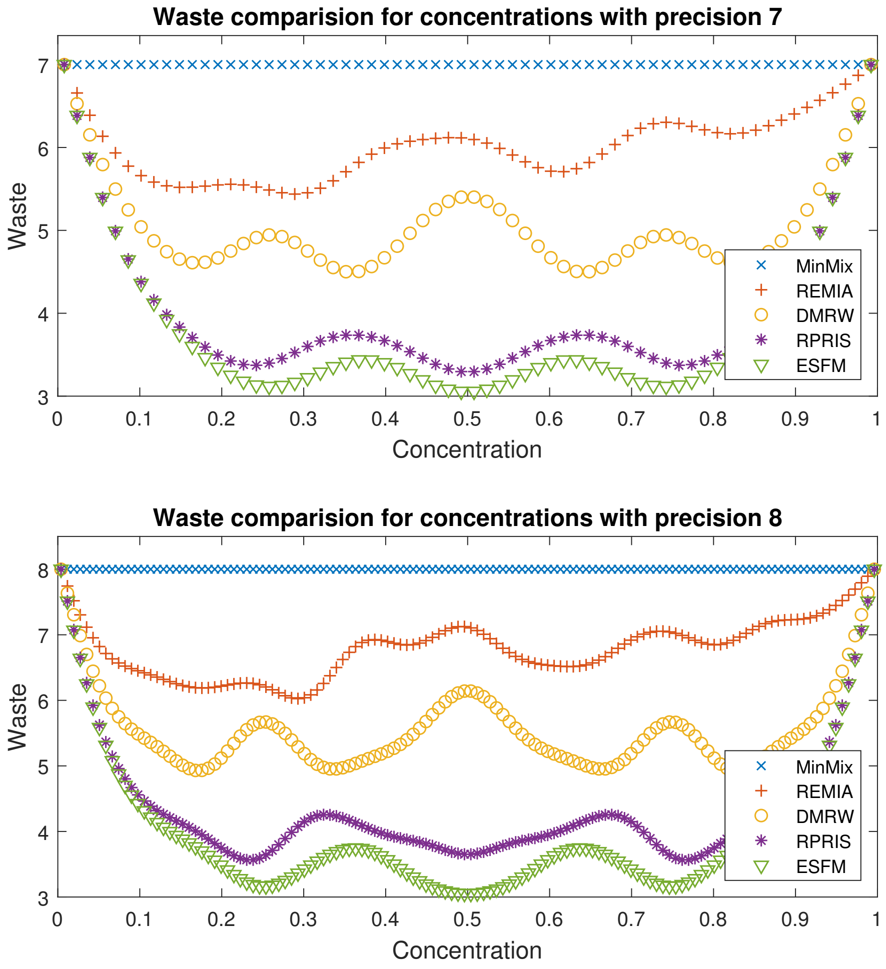

Figure 2.11: The number of waste droplets of algorithms `Min-Mix`, `DMRW`, `REMIA`, `ESFM`, and our algorithm `RPRIS`, for all concentrations with precision 7 (top figure) and 8 (bottom figure). All graphs are smoothed using `MATLAB`'s *smooth* function.

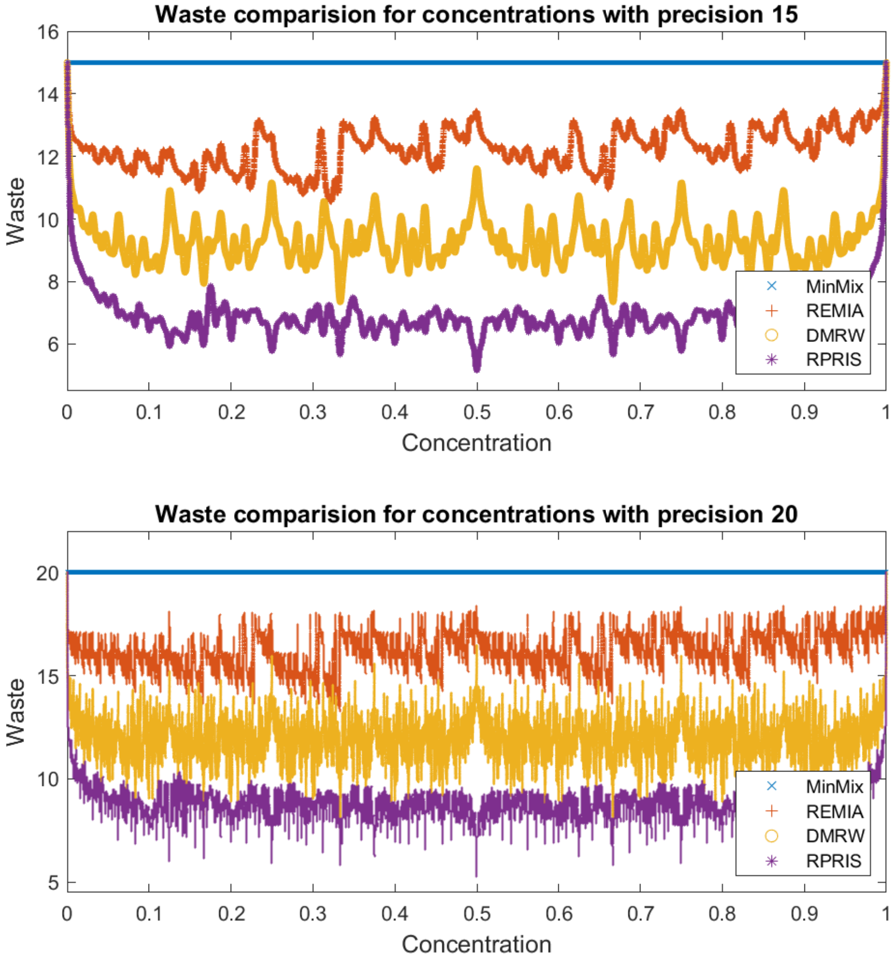

Figure 2.12: The number of waste droplets of algorithms `Min-Mix`, `DMRW`, `REMIA`, and our algorithm `RPRIS`, for all concentrations with precision 15 (top figure) and 20 (bottom figure). All graphs are smoothed using `MATLAB`'s *smooth* function.

Chapter 3

Algorithm for Perfect-Mixing

In this chapter we consider the following sub-problem of MIXREACHABILITY:

PERFECTMIXABILITY: Given a set C of n droplets with binary concentrations and binary average value $\mu = (\sum_{c \in C} c)/n$, is there a mixing graph that mixes C perfectly, converting C into the set of n droplets of concentration μ?

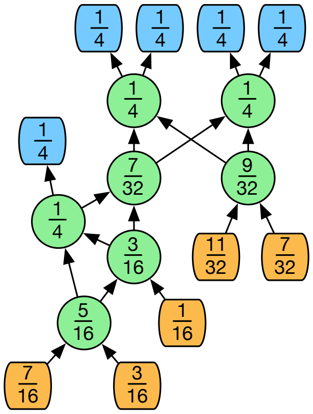

Figure 3.1: A mixing graph that perfectly mixes $C = \{\frac{1}{16}, \frac{3}{16}, \frac{7}{32}, \frac{11}{32}, \frac{7}{16}\}$.

Figure 3.1 shows an example of a perfect-mixing graph. As an example of a set that is not perfectly mixable, consider $D = \left\{0, \frac{3}{16}, \frac{9}{16}\right\}$. After any (non-zero) number of mixing operations the resulting set of concentrations will have the form $D' = \{a, a, b\}$ for $a \neq b$, so no finite mixing graph will convert D into its perfect mixture $\left\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right\}$.

We provide a complete characterization of perfectly mixable sets, and show that there is a polynomial-time algorithm that tests whether a given set is perfectly mixable, and if so, constructs a polynomial-size perfect-mixing graph for it.

We represent droplet sets as multisets of concentration values. First, we observe that without loss of generality we can assume that $C \cup \{\mu\} \sqsubset \mathbb{Z}$, for otherwise we can simply rescale all values by an appropriate power of 2. ($\mathbb{Z}$ is the set of integers; $\mathbb{Z}_{>0}$ and $\mathbb{Z}_{\geq 0}$ are the sets of positive and non-negative integers, respectively. Symbol $\sqsubset$ is used to specify a ground set of a multiset.) For any finite multiset $A \sqsubset \mathbb{Z}$ and $b \in \mathbb{Z}_{>0}$, we define A to be *b-congruent* if $x \equiv y \pmod{b}$ for all $x, y \in A$. (Otherwise we say that A is *b-incongruent*.)

We say that C satisfies *Condition (MC)* if, for each odd $b \in \mathbb{Z}_{>0}$, if C is b-congruent then $C \cup \{\mu\}$ is b-congruent as well, where $\mu = \mathsf{ave}(C)$. The following theorem summarizes our results.

Theorem 5 *Assume that $n \geq 4$ and $C \cup \{\mu\} \sqsubset \mathbb{Z}$, where $\mu = \mathsf{ave}(C)$. Then:*

(a) *C is perfectly mixable if and only if C satisfies Condition (MC).*

(b) *If C satisfies Condition (MC) then it can be perfectly mixed with precision at most 1 and in a polynomial number of steps. (That is, C has a perfect-mixing graph of polynomial size where all intermediate concentration values are half-integral.)*

(c) *There is a polynomial-time algorithm that tests whether C is perfectly mixable and, if so, computes a polynomial-size perfect-mixing graph for C.*

Part (b) implies that, in general (if the concentrations in $C \cup \{\mu\}$ are arbitrary binary values), C can be mixed perfectly with precision at most $d + 1$, where d is the maximum precision in $C \cup \{\mu\}$; in other words, at most one extra bit of precision is needed in the intermediate nodes of a perfect-mixing graph for C.

This extra 1-bit of precision in part (b) of Theorem 5 is necessary. For example, $C = \{0, 0, 0, 3, 7\}$ (for which $\mu = 2$) cannot be mixed with precision 0. If we mix 3 and 7, we will obtain multiset $\{0, 0, 0, 5, 5\}$ which is not perfectly mixable, as it violates Condition (MC). Any other mixing creates fractional values. However, C does have a mixing graph where the intermediate precision is at most 1 — see Figure 3.2.

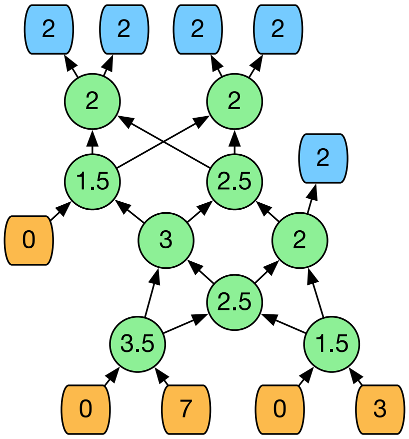

Figure 3.2: A perfect mixing graph for $C = \{0, 0, 0, 3, 7\}$ with precision 1.

The proof of Theorem 5 is given in several sections. The necessity of Condition (MC) in Theorem 5(a) is relatively simple to show; the proof appears in Section 3.2.

The proof that Condition (MC) is sufficient is more challenging. We first show in Section 3.3 (see Corollary 12) that, in essence, in Condition (MC) it is sufficient to consider only the values of b, no bigger than the maximum concentration in C, that are powers of prime factors of n. This property is used in Section 3.4 to show that any set C that satisfies Condition (MC) has a perfect-mixing graph, completing the proof of Theorem 5(a). The mixing graph constructed in Section 3.4 has precision at most 1, proving the frst part of Theorem 5(b). The second part, showing the existence of a perfect-mixing graph of polynomial size is established in Section 3.5. The proof of Theorem 5(c) is divided into two parts: that testing Condition (MC) can be done in polynomial time follows directly from Corollary 12 in Section 3.3, while a polynomial-time algorithm for constructing a perfect-mixing graph is implicit in our construction in Section 3.5.

3.1 Preliminaries

Let $\mathbb{Q}_0$ be the set of binary numbers. Following the notation from Chapter 2, by $\mathsf{prec}(c)$, where $c \in \mathbb{Q}_0$, we denote the precision of c, that is the number of fractional bits in the binary representation of c, assuming there are no trailing 0's. In other words, $\mathsf{prec}(c)$ is the smallest $d \in \mathbb{Z}_{\geq 0}$ such that $c = a/2^d$ for some $a \in \mathbb{Z}$. If $c = a/2^d$ represents actual fluid concentration, then we have $0 \leq a \leq 2^d$. However, it is convenient to relax this restriction and allow "concentration values" that are arbitrary binary numbers, even negative. In fact, as we show shortly, it will be convenient to work with integral values.

Similarly, by a *configuration* we mean a multiset of n binary numbers, called droplets or concentrations. In the literature, multisets are often represented by their characteristic functions (that specify the multiplicity of each element of the ground set). In this chapter we will generally use set-theoretic terminology, with its natural interpretation. For example, for a configuration C and a concentration a, $a \in C$ means that the multiplicity of a in C is strictly positive, while $a \notin C$ means that it's zero. Or, $C - \{a\} = C'$ means that the multiplicity of a in C' is one less than in C, while other multiplicities are the same. The number of droplets in C is denoted $|C| = n$, while the number of different concentrations is denoted $\|C\| = m$. As in Chapter 2, we will typically denote a configuration by $C = \{f_1 : c_1, f_2 : c_2, ..., f_m : c_m\} \sqsubset \mathbb{Q}_0$, where each c_i represents a (different) concentration value and f_i denotes the multiplicity of c_i in C, so that $\sum_{i=1}^{m} f_i = n$. Occasionally, if it does not lead to confusion, we may say "droplet c_i" or "concentration c_i", referring to some droplet with concentration c_i. If $f_i = 1$, we shorten "$f_i : c_i$" to just "c_i". If $f_i = 1$ we say that droplet c_i is a *singleton*, if $f_i = 2$ we say that droplet c_i is a *doubleton* and if $f_i \geq 2$ we say that droplet c_i is a *non-singleton*. By $\mathsf{sum}(C)$ we denote the sum of C, that is $\mathsf{sum}(C) = \sum_{c \in C} c$. $\mathsf{ave}(C) = \mathsf{sum}(C)/n$ is the average value of the concentrations in C and will be typically denoted by μ. (Later, we will typically deal with configurations C such that $C \cup \{\mu\} \sqsubset \mathbb{Z}$.)

In this work, since we are not concerned with the topological properties of mixing graphs, we will often identify a mixing graph G with a corresponding *mixing sequence*, which is a sequence (not necessarily unique) of mixing operations that converts C into its perfect mixture. In other words, a mixing sequence is a sequence of mixing operations in a topological ordering of a mixing graph.

Of course in a perfect-mixing graph (or sequence) G for C, all concentrations in G, including those in $C \cup \{\mu\}$, must have finite precision (that is, belong to $\mathbb{Q}_0$) which is at least $\max\{\mathsf{prec}(C), \mathsf{prec}(\mu)\}$. In addition to the basic question about finding a perfect-mixing graph for C, we are also interested in bounding the precision required to do so.

For $x \in \mathbb{Q}_0$, define multisets $C + x = \{c + x \mid c \in C\}$, $C - x = C + (-x)$, and $C \cdot x = \{c \cdot x \mid c \in C\}$. The next observation says that offsetting all values in C does not affect perfect mixability, as long as the offset value's precision does not exceed that of C or μ.

Observation 6 *Let $\mu = \mathsf{ave}(C)$ and $x \in \mathbb{Q}_0$. Also, let $d \in \mathbb{Z}_{>0}$ be such that $d \geq \max\{\mathsf{prec}(C), \mathsf{prec}(\mu), \mathsf{prec}(x)\}$. Then C is perfectly mixable with precision d if and only if $C' = C + x$ is perfectly mixable with precision d.*

Proof. ($\Rightarrow$) Suppose that G is a perfect-mixing sequence for C with precision d. Run the same sequence G on input C'. If some mixing step in G produces a value z when the input is C then on C' its value is $z + x$, and $\mathsf{prec}(z + x) \leq \max\{\mathsf{prec}(z), \mathsf{prec}(x)\} \leq d$. Thus the maximum precision in G for input C' is at most d.

($\Leftarrow$) The proof for this implication follows from noting that $\mu' = \mathsf{ave}(C') = \mu + x$, $\max\{\mathsf{prec}(C'), \mathsf{prec}(\mu')\} \leq d$, and by applying the above argument to $-x$ instead of x. ■

Observation 7 *Let $\mu = \mathsf{ave}(C)$, $\delta = \max\{\mathsf{prec}(C), \mathsf{prec}(\mu)\}$, $C' = C \cdot 2^\delta$ with $\mu' = \mathsf{ave}(C') = 2^\delta \mu$. (Thus $C' \cup \{\mu'\} \sqsubset \mathbb{Z}$.) Then C is perfectly mixable with precision $d \geq \delta$ if and only if C' is perfectly mixable with precision $d' = d - \delta$.*

Proof. ($\Rightarrow$) Let G be a perfect-mixing sequence for C, with precision d. Run the same sequence G on input C'. If some node in G produces a value z on input C, then its value on input C' will be $z2^\delta$, and $\mathsf{prec}(z2^\delta) = \max\{\mathsf{prec}(z) - \delta, 0\} \leq d - \delta = d'$.

($\Leftarrow$) Let G' be a perfect-mixing sequence for C' with precision d'. Run G' on input C. If some node in G' produces a value y on input C', then its value on input C will be $y/2^\delta$, and $\mathsf{prec}(y/2^\delta) \leq \mathsf{prec}(y) + \delta \leq d' + \delta = d$. ■

Integral configurations. Per Observation 7, we can restrict our attention to configurations with integer values and average, that is, we will be assuming that $C \cup \{\mu\} \sqsubset \mathbb{Z}$.

For $x \in \mathbb{Z}_{>0}$, if each $c \in C$ is a multiple of x, let $C/x = \{c/x \mid c \in C\}$. For integral configurations, we can extend Observation 7 to also multiplying C by an odd integer or dividing it by a common odd factor of all concentrations in C.

Observation 8 *Assume that $C \cup \{\mu\} \sqsubset \mathbb{Z}$ and let $x \in \mathbb{Z}_{>0}$ be odd.*

(a) *Let $C' = C \cdot x$. Then C is perfectly mixable with precision 0 if and only if C' is perfectly mixable with precision 0.*

(b) *Suppose that x is a divisor of all concentrations in $C \cup \{\mu\}$. Then C is perfectly mixable with precision 0 if and only if C/x is perfectly mixable with precision 0.*

Proof. Part (b) follows from (a), so we only prove part (a). Any sequence G of mixing operations for C can be applied to $C \cdot x$. By simple induction, if some intermediate value in G was an integer z, now its value will be $z \cdot x$, also an integer. This shows the ($\Rightarrow$) implication.

To justify the ($\Leftarrow$) implication, suppose that G' is a sequence of mixing operations for C' and that all concentrations in G' are integer. Since x is odd and all concentrations in C' are multiples of x, every concentration in G' will be also a multiple of x, including $\mathsf{ave}(C')$. Thus, if we run G' on C instead of C', if some node's concentration was $c \cdot x$, now it will be c. Thus, the ($\Leftarrow$) implication holds. ■

Input format and size. The input configuration C is specified by listing the concentration values of individual droplets, and its size is the total number of bits used in this representation. For the sake of concretness, we will assume that C is already rescaled to consist only of integers, and we will define the input size as $s(C) = \sum_{c \in C} \log(|c| + 2)$. This value is within a small constant factor of the actual number of bits representing C.

3.2 Necessity of Condition (MC)

In this section we prove that Condition (MC) in Theorem 5(a) is necessary for perfect mixability. So let $C \cup \{\mu\} \sqsubset \mathbb{Z}$, where $\mu = \mathsf{ave}(C)$, and assume that C is perfectly mixable. Let G be a graph (or a sequence) that mixes C perfectly. We want to prove that C satisfies Condition (MC).

Suppose that C is b-congruent for some odd $b \in \mathbb{Z}_{>0}$. Consider an auxiliary configuration $C' = C \cdot 2^{\delta}$, where δ is sufficiently large, so that all intermediate concentrations in G when applying G to C' are integral. This C' is b-congruent, and starting from C', G produces a perfect mixture of C', that is $\{n : \mu'\}$, for $\mu' = 2^{\delta}\mu$.

Since C' is b-congruent, there is $\beta \in \{0, ..., b-1\}$ such that for each $x \in C'$ we have $x \equiv \beta \pmod{b}$. We claim that this property is preserved as we apply mixing operations to droplets in C'. Indeed, suppose that we mix two droplets with concentrations $x, y \in C'$, producing two droplets with concentration z. Since $x \equiv \beta \pmod{b}$ and $y \equiv \beta \pmod{b}$, we have $x = \alpha b + \beta$ and $y = \alpha' b + \beta$, for some $\alpha, \alpha' \in \mathbb{Z}$, so $z = \frac{1}{2}(x+y) = (\frac{1}{2}(\alpha + \alpha'))b + \beta$. As b is odd (and z is integer), $\alpha + \alpha'$ must be even, and therefore $z \equiv \beta \pmod{b}$, as claimed.

Eventually G produces μ', so this must also hold for $z = \mu'$. This implies that $C' \cup \{\mu'\}$ is b-congruent.

Finally, since $C' \cup \{\mu'\}$ is b-congruent, for all $2^\delta x, 2^\delta y \in C' \cup \{\mu'\}$ it holds that $2^\delta x \equiv 2^\delta y \pmod{b}$. But this implies that $x \equiv y \pmod{b}$, because b is odd. So we can conclude that $C \cup \{\mu\}$ is b-congruent, thus proving that C satisfies Condition (MC).

3.3 Some Auxiliary Lemmas

Condition (MC) involves all odd $b \in \mathbb{Z}_{>0}$, so it does not directly lead to an efficient test for perfect mixability (Theorem 5(c)). Hence, we now show that only factors b of n that are odd prime powers need to be considered, which implies that perfect mixability can be tested in polynomial time. These properties will also play a role in the sufficiency proof of Condition (MC) in Theorem 5(a).

Lemma 9 *Let $b, c \in \mathbb{Z}_{>0}$ and $A \sqsubset \mathbb{Z}$. (a) If A is bc-congruent then A is also b-congruent. (b) If $\mathsf{gcd}(b, c) = 1$ and A is both b-congruent and c-congruent then A is bc-congruent.*

Proof. Part (a) is trivial, because $x \equiv y \pmod{bc}$ implies that $x \equiv y \pmod{b}$. Part (b) is also simple: Suppose that $x \equiv y \pmod{b}$ and $x \equiv y \pmod{c}$. This means that $b|(x - y)$ and $c|(x - y)$. This, since b, c are co-prime, implies that $(bc)|(x - y)$, which is equivalent to $x \equiv y \pmod{bc}$. ∎

Lemma 10 *If Condition (MC) holds for all $b \in \mathbb{Z}_{>0}$ that are a power of an odd prime then it holds for all odd $b \in \mathbb{Z}_{>0}$.*

Proof. Assume that condition (MC) holds for all b that are odd prime powers. Let $b' \in \mathbb{Z}_{>0}$ be odd, with factorization $b' = p_1^{\gamma_1} ... p_k^{\gamma_k}$, for different odd primes $p_1, ..., p_k$, and suppose that C is b'-congruent. Then, by Lemma 9(a), C is also $p_i^{\gamma_i}$-congruent for all i. Since condition (MC) holds for $p_i^{\gamma_i}$, this implies that $C \cup \{\mu\}$ is $p_i^{\gamma_i}$-congruent for all i. By repeated application of Lemma 9(b), we then obtain that $C \cup \{\mu\}$ is b'-congruent as well. So Condition (MC) holds for b'. ■

Lemma 11 *Let $b \in \mathbb{Z}_{>0}$ be odd. If $\gcd(b, n) = 1$ then Condition (MC) holds for b.*

Proof. Assume that C is b-congruent. By Observation 6, without loss of generality we can assume that all numbers in C are multiples of b. (Otherwise we can consider $C' = C - c$, for an arbitrary $c \in C$, because Condition (MC) is not affected by offsetting C.) Thus $\mathsf{sum}(C) = b\beta$, for some $\beta \in \mathbb{Z}_{\geq 0}$, which gives us that $\mu = \mathsf{sum}(C)/n = b\beta/n$, and, as b and n are co-prime, we can conclude that μ is a multiple of b. This means that $C \cup \{\mu\}$ is b-congruent, proving that Condition (MC) holds for b. ■

Corollary 12 *Let c_{max} be the maximum absolute value of concentrations in C, and suppose that Condition (MC) holds for all $b \in \mathbb{Z}_{>0}$ that are powers of odd prime factors of n and satisfy $b \leq c_{max}$. Then Condition (MC) holds for all odd b.*

To substantiate Corollary 12, note that C satisfies Condition (MC) for all odd $b \in \mathbb{Z}_{>0}$ larger than c_{max}. This holds because for each such b, $c \mod b = c$ for all $c \in C$, so the remainders of the concentrations in C modulo b are different. (Except for the trivial case where all concentrations in C are equal, in which case μ is also equal and thus C satisfies Condition (MC); such C is actually perfectly mixed.)

Algorithm 1 PerfectMixabilityTesting(C)

```
$n \leftarrow |C|$
$\mu \leftarrow \mathsf{ave}(C)$
$c_{max} \leftarrow$ maximum absolute concentration in $C$
$P \leftarrow$ powers of odd prime factors of $n$ that are at most $c_{max}$
for all $p \in P$ do
    if $C$ is $p$-congruent but $C \cup \{\mu\}$ is not then
        return false
    end if
end for
return true
```

Corollary 12 implies that perfect mixability testing can be done in polynomial time (implying the first part of Theorem 5(c)). To justify this, recall that the input size is $s(C) = \sum_{c \in C} \log(|c| + 2) \geq n$. Thus the factoring of n can be computed in time polynomial in the input size; n has at most $\log n$ distinct odd prime factors. Each such prime factor has at most $\log c_{max}$ powers that are no bigger than c_{max}. Therefore, the total number of powers of odd prime factors of n that are no bigger than c_{max} is at most $\log n \log c_{max}$. A pseudocode of the resulting algorithm is given in Algorithm 1.

3.4 Sufficiency of Condition (MC)

In this section we prove that Condition (MC) in Theorem 5(a) is sufficient for perfect mixability. A perfect-mixing graph constructed in our argument has precision at most 1, showing also the first part of Theorem 5(b).

Note that in Theorem 5 we assume that $n = |C| \geq 4$. Regarding smaller values of n, for $n = 2$, trivially, all configurations C with two droplets are perfectly mixable with precision 0. The case $n = 3$ is exceptional, as in this case Theorem 5 is actually false. (For example, consider configuration $C = \{0, 1, 5\}$, for which $\mu = 2$. This configuration is b-incongruent for all odd $b > 1$, so it satisfies condition (MC), but is not perfectly mixable.) Nevertheless, for $n = 3$, perfectly mixable configurations are easy to characterize: Let $C = \{a, b, c\}$, where $a \leq b \leq c$. Then C is perfectly mixable if and only if $b = \frac{1}{2}(a + c)$. Further, if this condition holds, C is perfectly mixable with precision 0. (That this condition is sufficient is obvious. That it is also necessary can be proven by following the argument given in the introduction for the example configuration D right after the definition of PerfectMixability.)

So from now on we assume that $n \geq 4$. Let C be the input configuration and $\mu = \mathsf{ave}(C)$, where $C \cup \{\mu\} \sqsubset \mathbb{Z}$. The outline of our proof is as follows:

- First we prove that C is perfectly mixable with precision 0 when n is a power of 2. This easily extends to configurations C called *near-final*, which are disjoint unions of multisets with the same average and cardinalities being powers of 2. In particular, this proves Theorem 5(a) for $n = 4$.

- Next, we give a proof for $n \geq 7$. The basic idea of the proof is to define an invariant (I) and show that any configuration that satisfies (I) has a mixing operation that either preserves invariant (I) or produces a near-final configuration. Condition (I) is stronger than (MC) (it implies (MC), but not vice versa), but we show that that any configuration that satisfies Condition (MC) can be modified to satisfy (I).

- We then give separate proofs for $n = 5, 6$. The proof for $n = 5$ is similar to the case $n \geq 7$, but it requires a more subtle invariant. The proof for $n = 6$ follows from the proofs for $n \geq 7$ and $n = 5$.

3.4.1 Perfect Mixability of Near-Final Configurations

Let $C \sqsubset \mathbb{Z}$ be a configuration with $|C| = n = \sigma 2^\tau$, for some odd $\sigma \in \mathbb{Z}_{>0}$ and $\tau \in \mathbb{Z}_{\geq 0}$, with $\mathsf{ave}(C) = \mu \in \mathbb{Z}$. We say that C is *near-final* if it can be partitioned into multisets $C_1, C_2, ..., C_k$, such that, for each j, $\mathsf{ave}(C_j) = \mu$ and $|C_j|$ is a power of 2. In this sub-section we show (Lemma 13 below) that near-final configurations are perfectly mixable with precision 0. We also show that configurations with only two different concentrations that satisfy Condition (MC) are near-final, and thus perfectly mixable.

Define $\Psi(C) = \sum_{c \in C} (c - \mu)^2$ as the variance of C. Obviously $\Psi(C) \in \mathbb{Z}_{\geq 0}$, $\Psi(C) = 0$ if and only if C is a perfect mixture, and, by a straightforward calculation, mixing any two different same-parity concentrations in C decreases the value of $\Psi(C)$ by at least 1.

Lemma 13 *If C is near-final then C is perfectly mixable with precision* 0.

Proof. It is sufficient to prove the lemma for the case when n is a power of 2. (Otherwise, we can apply it separately to each set C_j in the partition of C from the definition of near-final configurations.)

So assume that n is a power of 2. It is sufficient to show that if $\|C\| = m \neq 1$ (that is, C is not yet perfectly mixed) then it contains two different concentrations with the same parity. (Each such mixing strictly decreases $\Psi(C)$, so a finite sequence of such mixing operations will perfectly mix C.) This is trivially true if $m \geq 3$, so it is sufficient to prove it when $m = 2$, that is for $C = \{f_1 : c_1, f_2 : c_2\}$. Without loss of generality, by Observation 6, we can assume that $c_2 = 0$, and then we claim that c_1 is even. We have $\mathsf{ave}(C) = \mu = f_1 c_1/n$. As $\mu \in \mathbb{Z}$, $f_1 < n$ and n is a power of 2, we have that c_1 must be even, as claimed. ■

Lemma 14 *Assume that $\|C\| = 2$, say $C = \{f_1 : c_1, f_2 : c_2\}$, and that C satisfies Condition (MC). Then σ divides f_1 and f_2. Consequently, we have that n is not prime and C is near-final.*

Proof. Without loss of generality, we can assume that $c_2 = 0$. (Otherwise we can consider $C' = C - c_2$ instead. This does not affect Condition (MC) and the property of being near-final.) Let $c_1 = \alpha 2^\gamma$, for some odd $\alpha \in \mathbb{Z}_{>0}$. Then $\mu = f_1 c_1/n = f_1 \alpha 2^\gamma/(\sigma 2^\tau)$. Since α divides c_1, Condition (MC) implies that α must also divide μ. In other words, $\mu/\alpha = f_1 2^\gamma/(\sigma 2^\tau)$ is integer. This implies, in turn, that f_1 is a multiple of σ, as claimed. Since $f_2 = n - f_1$, it also gives us that f_2 is a multiple of σ.

This immediately implies that n cannot be prime, for otherwise we would have that $n = \sigma$, so n would be a divisor of $f_1 < n$.

To prove the last claim, let $f_1 = \sigma f'_1$ and $f_2 = \sigma f'_2$, for some $f'_1, f'_2 \in \mathbb{Z}_{>0}$. Split C into σ sub-multisets of the form $C_j = \{f'_1 : c_1, f'_2 : 0\}$, for $j = 1, 2, ..., \sigma$. The cardinality of each C_j is $f'_1 + f'_2 = n/\sigma = 2^\tau$ and its average is $\mathsf{ave}(C_j) = f'_1 c_1/(f'_1 + f'_2) = (f_1/\sigma)c_1/(n/\sigma) = f_1 c_1/n = \mu$. Therefore C is near-final, as claimed. ■

3.4.2 Proof for Arbitrary $n \geq 7$

In this sub-section we prove that Condition (MC) in Theorem 5(a) is sufficient for perfect mixability when $n \geq 7$. Let C be a configuration that satisfies Condition (MC), where $C \cup \{\mu\} \sqsubset \mathbb{Z}$ and $|C| = n$. Also, let the factorization of n be $n = 2^{\tau_0} p_1^{\tau_1} p_2^{\tau_2} ... p_s^{\tau_s}$, where $\{p_1, p_2, ..., p_s\} = \bar{p}$ is the set of the odd prime factors of n and $\{\tau_1, \tau_2, ..., \tau_s\}$ are their corresponding multiplicities.

If $A \sqsubset \mathbb{Z}$ is a configuration with $|A| = n$ (where n is as above) and $\mathsf{ave}(A) \in \mathbb{Z}$, we will say that A is *$\bar{p}$-incongruent* if A is p_r-incongruent for all r. If A is $\bar{p}$-incongruent then, by Lemma 9(a), it is b-incongruent for all b that are powers of p_r's, which, by Corollary 12, implies that A satisfies Condition (MC). Further, if A is also not near-final then Lemma 14 implies that $\|A\| \geq 3$. We summarize these observations below. (They will be often used in this section without an explicit reference.)

Observation 15 *Assume that a configuration $A \sqsubset \mathbb{Z}$ with $\mathsf{ave}(A) \in \mathbb{Z}$ is $\bar{p}$-incongruent. Then (a) A satisfies Condition (MC), and (b) if A is not near-final then $\|A\| \geq 3$.*

Proof outline. The outline of the proof is as follows (see Figure 3.3): Instead of dealing with C directly, we will consider a $\bar{p}$-incongruent configuration $\check{C} \sqsubset \mathbb{Z}$ with $\check{\mu} = \mathsf{ave}(\check{C}) \in \mathbb{Z}$ that is "equivalent" to C in the sense that C is perfectly mixable with precision at most 1 if and only if $\check{C}$ is perfectly mixable with precision 0.

It is thus sufficient to show that $\check{C}$ is perfectly mixable with precision 0. To this end, we first apply some mixing operations to $\check{C}$, producing only integer concentrations, that convert $\check{C}$ into a configuration E such that:

(I.0) $E \sqsubset \mathbb{Z}$ and $\mathsf{ave}(E) = \check{\mu}$,

(I.1) E has at least 2 distinct non-singletons, and

(I.2) E is $\bar{p}$-incongruent.

We refer to the three conditions above as Invariant (I). Then we show that any configuration E that satisfies Invariant (I) has a pair of different concentrations that are "safe" to mix, in the sense that after they are mixed the new configuration is either near-final or satisfies Invariant (I). We can thus repeatedly mix such safe mixing pairs, preserving Invariant (I), until we produce a near-final configuration, that, by the previous section, can be perfectly mixed with precision 0.

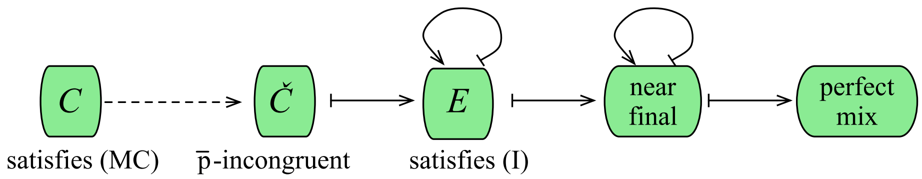

Figure 3.3: Proof outline for $n \geq 7$. The first dashed arrow represents replacing C by $\check{C}$. Solid arrows represent mixing operations.

Replacing C by $\check{C}$. We now explain how to modify C. We will do it in steps. First, let $C' = C - c_1$, for some arbitrarily chosen $c_1 \in C$. Note that $\mu' = \mathsf{ave}(C') = \mu - c_1 \in \mathbb{Z}$, that $0 \in C'$, and that C' satisfies Condition (MC). By Observation 6, C is perfectly mixable if and only if C' is perfectly mixable (with the same precision), so it is sufficient to show that C' is perfectly mixable.

Then, let $\theta \in \mathbb{Z}_{>0}$ be the maximum odd integer that divides all concentrations $c \in C'$ (that is, the greatest common odd divisor of C'). Let $C'' = C'/\theta$. By Observation 8(b) and the paragraph above, C is perfectly mixable if and only C'' is perfectly mixable (with the same precision), so from now on we can replace C by C''.

By Condition (MC) applied to C', θ is a divisor of μ', so $\mu'' = \mathsf{ave}(C'') = \mu'/\theta \in \mathbb{Z}$. Next, we claim that C'' is $\bar{p}$-incongruent. To show this, we argue by contradiction. Suppose that C'' is p_r-congruent for some r. This means that there is $\beta \in \{0, 1, ..., p_r - 1\}$ such that $c \equiv \beta \pmod{p_r}$ for all $c \in C''$. Since $0 \in C''$ (because $0 \in C'$), we must have $\beta = 0$. In other words, all $c \in C''$ are multiples of p_r. That would imply, however, that all $c \in C'$ are multiples of θp_r, which contradicts the choice of θ, completing the proof.

Finally, let $\check{C} = 2{\cdot}C''$ and $\check{\mu} = 2\mu'' = \mathsf{ave}(\check{C})$. All concentrations in $\check{C}$ are even and, since multiplying all concentrations by 2 does not affect $\bar{p}$-incongruence, $\check{C}$ is $\bar{p}$-incongruent. By Observation 7, and the properties of C'' established above, C is perfectly mixable with precision at most 1 if and only if $\check{C}$ is perfectly mixable with precision 0. Therefore, from now on, it is sufficient to show a mixing sequence with all integral concentration values that converts $\check{C}$ into its perfect mixture $\{n : \check{\mu}\}$.

Converting $\check{C}$ into E. Let $\check{C}$ be the configuration constructed above. We now show that with at most two mixing operations, producing only integer values, we can convert $\check{C}$ into a configuration E that satisfies Invariant (I).

Let $A \sqsubset \mathbb{Z}$ be a configuration with $\mathsf{ave}(A) \in \mathbb{Z}$ and $|A| = n$. Assume that A is $\bar{p}$-incongruent. For different concentrations $a, a' \in A$ with the same parity, we say that the pair (a, a') is p_r*-safe* if mixing a and a' converts A into a p_r-incongruent configuration; in other words, there is $a'' \in A - \{a, a'\}$ that satisfies $a'' \not\equiv \frac{1}{2}(a + a') \pmod{p_r}$. (Otherwise, we say that the pair (a, a') is p_r*-unsafe*, or just *unsafe* to generalize.) We will also say that (a, a') is $\bar{p}$*-safe* if it is p_r-safe for all r. For example, let $n = 5$ and $A = \{0, 0, 3, 7, 10\}$, for which $\mathsf{ave}(A) = 4$. Then pair $(0, 10)$ is 5-safe but pair $(3, 7)$ is 5-unsafe.

Lemma 16 *Let A be a $\bar{p}$-incongruent configuration with $\mathsf{ave}(A) \in \mathbb{Z}$ and $|A| = n$. (Recall that $n \geq 7$.) Then*

(a) For each r, there is at most one p_r-unsafe pair in A.

(b) There are at most $n - 5$ droplets involved in same-parity concentration pairs that are unsafe.

(c) If a concentration $a \in A$ is a non-singleton (has multiplicity at least 2) then for any $b \in A$ with $b \neq a$ and the same parity as a, the pair (a, b) is $\bar{p}$-safe.

Proof. (a) Suppose that some pair (a_1, a_2) of concentrations with $a_1 \neq a_2$ and same parity is p_r-unsafe, and let $\beta = (\frac{1}{2}(a_1 + a_2)) \bmod p_r$. The assumption about a_1, a_2 implies that $b \equiv \beta \pmod{p_r}$ for all $b \in A - \{a_1, a_2\}$, and the assumption that A is $\bar{p}$-incongruent implies that $a_i \not\equiv \beta \pmod{p_r}$ for at least one $i \in \{1, 2\}$. We claim that this must in fact hold for *both* $i \in \{1,2\}$. Indeed, say that $a_1 \not\equiv \beta \pmod{p_r}$ but $a_2 \equiv \beta \pmod{p_r}$. This means that $p_r \not| (a_1 - \beta)$ and $p_r | (a_2 - \beta)$, which implies that $p_r \not| (\frac{1}{2}(a_1 + a_2) - \beta)$, contradicting the definition of β. Thus $a_i \not\equiv \beta \pmod{p_r}$ for both $i \in \{1,2\}$, as claimed.

It remains to show that any other pair of concentrations is p_r-safe. Fix three arbitrary concentrations $\{b_1, b_2, b_3\} \subseteq A - \{a_1, a_2\}$, so that we have $b_j \equiv \beta \pmod{p_r}$ for $j \in \{1,2,3\}$. Consider any two different same-parity concentrations $c_1, c_2 \in A$ with $\{c_1, c_2\} \neq \{a_1, a_2\}$, and let A' be obtained from A by mixing droplets c_1 and c_2. Then A' must still contain some droplet b_j and, since $\{c_1, c_2\} \neq \{a_1, a_2\}$, A' will also contain some droplet a_i. As we have $a_i \not\equiv b_j \pmod{p_r}$, A' is p_r-incongruent, and thus (c_1, c_2) is p_r-safe.

(b) By part (a), the number of concentrations involved in same-parity unsafe pairs is at most $2s$, where s is the number of distinct odd prime factors of n, so it remains to show that $2s \leq n - 5$. Indeed, if n equals either 7 or 8 (for which $s = 1$ or 0, respectively), then the inequality holds. For $n \geq 9$, using the fact that $s \leq \log_3 n$, it is sufficient to show that $2 \log_3 n \leq n - 5$. This is true, because for $n = 9$ the equality holds, and for $n \geq 9$ the left-hand side grows slower than the right-hand side.

(c) Fix some factor p_r of n. As A is p_r-incongruent, there is a concentration $c \in A$ with $c \not\equiv a \pmod{p_r}$. We have two cases. If $b \equiv a \pmod{p_r}$ then $b \neq c$, so after mixing the new configuration A' will contain c and $\frac{1}{2}(a+b)$, where $\frac{1}{2}(a+b) \equiv a \pmod{p_r}$, so $c \not\equiv \frac{1}{2}(a+b) \pmod{p_r}$. On the other hand, if $b \not\equiv a \pmod{p_r}$, then A' will contain a and $\frac{1}{2}(a+b)$, and $a \not\equiv \frac{1}{2}(a+b) \pmod{p_r}$. Thus (a,b) is p_r-safe. As this holds for all r, (a,b) is $\bar{p}$-safe. ■

The configuration $\check{C}$ constructed earlier contains only even concentration values, and it already satisfies $\check{C} \cup \{\check{\mu}\} \sqsubset \mathbb{Z}$ and is $\bar{p}$-incongruent (that is, it satisfies conditions (I.0) and (I.2) for E). It remains to show that there are mixing operations involving only droplets already present in $\check{C}$ (and thus of even value, to assure that Condition (I.0) holds) that preserve condition (I.2), and such that the resulting configuration E is either near-final or it satisfies condition (I.1). If $\check{C}$ already has two or more non-singletons, we can take $E = \check{C}$ and we are done, so assume otherwise, namely that there is either one non-singleton in $\check{C}$ or none. We consider three cases.

Case 1: $\check{C}$ has one non-singleton a with multiplicity $f \geq 3$. Mix a with any singleton b and let E be the resulting configuration. In E we have two non-singletons and condition (I.2) will be satisfied, by Lemma 16(c). Thus E satisfies Invariant (I).

Case 2: $\check{C}$ has one non-singleton a with multiplicity 2. By Lemma 16(b), there are at least 5 droplets in $\check{C}$ not involved in any unsafe pair. Thus there are at least 3 singletons, say b, c, d, that are not involved in any unsafe pair. Mixing one of pairs (b, c) or (b, d) produces a concentration other than a. Mix this pair, and let E be the resulting configuration. Then E satisfies Invariant (I).

Case 3: $\check{C}$ has only singletons. By Lemma 16(b), there is a singleton, say $b \in \check{C}$, that is not involved in any unsafe pair (in fact, there are at least five, but we need just one here). Let $c \in \check{C} - \{b\}$ be a singleton nearest to b, that is one that minimizes $|c - b|$. By the choice of b, the pair (b, c) is $\bar{p}$-safe. Let $\check{C}' = \check{C} - \{b, c\} \cup \{a, a\}$, for $a = \frac{1}{2}(b + c)$, be the configuration obtained by mixing this pair. $\check{C}'$ is $\bar{p}$-incongruent and in $\check{C}'$ we have only one non-singleton a and its multiplicity is 2. We can thus apply Case 2 above to $\check{C}'$, converting it to E. (Note that, unlike for the $\check{C}$ in Case 2, our a may be odd. But since we do not mix a in Case 2, the argument is still valid.)

Preserving Invariant (I). We now present the last part of the proof, following the outline given at the beginning of this section. Let $E \sqsubset \mathbb{Z}$ be the configuration, say $E = \{f_1 : e_1, f_2 : e_2, ..., f_m : e_m\}$, with $\mathsf{ave}(E) = \check{\mu}$, obtained from $\check{C}$ by a sequence of mixing operations, as described earlier. If E is near-final then E has a perfect-mixing sequence, by Lemma 13. Otherwise, we show that E has a pair of concentrations whose mixing produces a configuration that is either near-final or satisfies Invariant (I).

Let $e_i, e_j \in E$ be two different concentrations and $e = \frac{1}{2}(e_i + e_j)$. Then the pair (e_i, e_j) is called *a safe mixing pair* if the configuration $E' = E - \{e_i, e_j\} \cup \{e, e\}$ obtained by mixing e_i and e_j is either near-final or satisfies the Invariant (I). We will be always choosing e_i and e_j with the same parity, which is a sufficient and necessary condition for E' to satisfy condition (I.0). Also, for E' to satisfy condition (I.2), the pair $\{e_i, e_j\}$ must be $\bar{p}$-safe.

We next prove that if configuration E satisfies Invariant (I) (and is not near-final) then it must contain a safe mixing pair. This will show that we can repeatedly mix E, maintaining Invariant (I), until we turn E into a near-final configuration, which we can then perfectly mix using Lemma 13.

Lemma 17 *Assume that E contains two different concentrations $e_i, e_j \in E$ with the same parity and $f_i \geq 3$. If E satisfies Invariant (I) then (e_i, e_j) is a safe mixing pair for E.*

Proof. Let $e = \frac{1}{2}(e_i + e_j)$ and let $E' = E - \{e_i, e_j\} \cup \{e, e\}$ be obtained from mixing e_i and e_j. Since $f_i > 1$, Lemma 16(c) implies that condition (I.2) holds for E'. In E' we will still have at least two droplets of concentration e_i and at least two droplets of concentration $e \neq e_i$. So condition (I.1) holds as well. (We remark that we could end up with $\|E'\| = 2$, which can happen if $f_j = 1$ and $\|E\| = 3$ with $e \in E$. If so, since E' satisfies (I.2), it must also satisfy condition (MC), and therefore, by Lemma 14, in this case E' is actually near-final.) ■

Lemma 18 *Assume that E contains three different concentrations $e_i, e_j, e_k \in E$ with the same parity and $f_i, f_j \geq 2$. If E satisfies Invariant (I) then one of (e_i, e_k), (e_j, e_k) is a safe mixing pair for E.*

Proof. Without loss of generality, assume that $|e_i - e_k| \leq |e_j - e_k|$ (otherwise swap i and j). We show that (e_i, e_k) is a safe mixing pair for E. Let $e = \frac{1}{2}(e_i + e_k)$ and let $E' = E - \{e_i, e_k\} \cup \{e, e\}$ be obtained from mixing e_i and e_k. Since $f_i > 1$, Lemma 16(c) implies that condition (I.2) holds for E'. From $|e_i - e_k| \leq |e_j - e_k|$, we have that $e \neq e_j$. So in E' we will have at least two droplets of concentration e_j and at least two droplets of concentration $e \neq e_j$. This means that condition (I.1) holds as well. ■

Lemma 19 *Assume that $\|E\| \geq 4$ and that E contains three different concentrations $e_i, e_j, e_k \in E$ with the same parity such that $f_i \geq 2$ and $f_j = f_k = 1$. If E satisfies Invariant (I), then one of (e_i, e_j), (e_i, e_k) is a safe mixing pair for E.*

Proof. By condition (I.1), there is another concentration $e_l \in E - \{e_i, e_j, e_k\}$ with $f_l \geq 2$. Without loss of generality, we can assume that $e = \frac{1}{2}(e_i + e_j) \neq e_l$ (otherwise we can use e_k instead of e_j). Mixing e_i and e_j produces $E' = E - \{e_i, e_j\} \cup \{e, e\}$. Since $f_i \geq 2$, condition (I.2) is satisfied. In E' there are at least two droplets with concentration e_l and at least two droplets with concentration $e \neq e_l$, so condition (I.1) is satisfied as well. ■

Lemma 20 *Assume that $\|E\| = 3$ and that E is not near-final. If E satisfies Invariant (I) then E has a safe mixing pair.*

Proof. Let $E = \{f_1 : e_1, f_2 : e_2, f_3 : e_3\}$. Reorder E so that $f_1 \geq f_2 \geq f_3$. From $f_1 + f_2 + f_3 = n \geq 7$ we have that $f_1 \geq 3$ and $f_2 \geq 2$. By symmetry, we can also assume that e_1 is even. If either e_2 or e_3 is even, then the existence of a mixing pair follows from Lemma 17.

So we can assume that e_2, e_3 are odd. We claim that (e_2, e_3) is a safe mixing pair. Let $e = \frac{1}{2}(e_2 + e_3)$ and let $E' = E - \{e_2, e_3\} \cup \{e, e\}$ be obtained from mixing e_2 and e_3.

Since $f_2 \geq 2$, Lemma 16(c) implies that condition (I.2) holds for E'. This, and Observation 15(a) imply that if $\|E'\| = 2$, then E' must be near-final, by Lemma 14, and we are done. So for the rest of the proof we can assume that $\|E'\| \geq 3$.

It is now sufficient to prove that E' satisfies (I.1). If $e \neq e_1$, in E' we have at least three droplets with concentration e_1 and at least two with concentration e, so E' satisfies (I.1). Otherwise, $e = e_1$ and we cannot have $f_2 = f_3$, because this would imply that E is near-final (by partitioning E into singletons $\{e_1\}$ and pairs $\{e_2, e_3\}$). So $f_2 > f_3$. Further, the assumption that $\|E'\| \geq 3$ implies that $f_3 \geq 2$. We thus obtain that $f_2 \geq 3$; so in E' there will be at least five droplets with concentration e_1 and at least two droplets with concentration e_2. This proves that (I.1) holds for E'. ■

Lemma 21 *Assume that $\|E\| = 4$ and that E is not near-final. If E satisfies Invariant (I) then E has a safe mixing pair.*

Proof. Let $E = \{f_1 : e_1, f_2 : e_2, f_3 : e_3, f_4 : e_4\}$. By symmetry and reordering, we can assume that e_1 is even and that $f_1 \geq f_2 \geq f_3 \geq f_4$. This, and condition $(I.1)$ imply that $f_1 \geq f_2 \geq 2$. We consider two cases, depending on the value of f_1.

Case 1: $f_1 \geq 3$. If at least one of e_2, e_3, e_4 is even, then the existence of a safe mixing pair follows from Lemma 17.

So assume now that e_2, e_3, e_4 are all odd. If $f_3 \geq 2$, we obtain a safe mixing pair from Lemma 18. Otherwise, $f_3 = f_4 = 1$, and we obtain a safe mixing pair from Lemma 19.

Case 2: $f_1 = 2$. Then $n \geq 7$ implies that $f_2 = f_3 = 2$ as well. If two concentrations among e_2, e_3, e_4 are even, or if e_2, e_3, e_4 are all odd, the existence of a safe mixing pair follows from Lemma 18.

Otherwise, one of e_2, e_3, e_4 is even and two are odd. We then want to mix e_4 with the one of e_1, e_2, e_3 that has the same parity as e_4. For concreteness, assume that e_2 is even and e_3, e_4 are odd. (The argument in all other cases is the same.) We claim that (e_3, e_4) is a safe mixing pair. Indeed, let $E' = E - \{e_3, e_4\} \cup \{e, e\}$, for $e = \frac{1}{2}(e_3 + e_4)$. Since $f_3 > 1$, Lemma 16(c) implies that condition (I.2) holds. In E' we have at least two droplets with concentration e. If $e \neq e_1$, then E' has two droplets with concentration $e_1 \neq e$; otherwise, if $e = e_1$, then E' has two droplets with concentration $e_2 \neq e$. Thus condition (I.1) holds for E'. ■

Lemma 22 *Assume that $\|E\| \geq 5$ and that E is not near-final. If E satisfies Invariant (I) then E has a safe mixing pair.*

Proof. Let $E = \{f_1 : e_1, f_2 : e_2, ..., f_m : e_m\}$, for $m \geq 5$. By symmetry and reordering, we can assume that e_1 is even and that $f_i \geq f_{i+1}$, for $i = 1, 2, ..., m - 1$. We have several cases.

Case 1: $f_1 \geq 3$. The same argument as is in Lemma 21 Case 1 holds.

Case 2: $f_1 = 2$. Then $f_2 = 2$ as well, by condition (I1). We have some sub-cases.

Case 2.1: $f_3 = 2$. In this case, choose three concentrations among e_1, e_2, e_3, e_4, e_5 with the same parity. This will give us three concentrations e_i, e_j, e_k with the same parity and with $f_i = 2$. If $f_j = 2$ or $f_k = 2$, we obtain a safe mixing pair from Lemma 18, otherwise we obtain a safe mixing pair from Lemma 19.

Case 2.2: $f_3 = ... = f_m = 1$ and e_2 is odd. Among e_3, e_4, e_5 there are either two even or two odd concentrations. By symmetry, we can assume e_3, e_4 are even. This gives us three concentrations e_i, e_j, e_k that satisfy the assumptions of Lemma 19, so we obtain a safe mixing pair by applying this lemma.

Case 2.3: $f_3 = ... = f_m = 1$ and e_2 is even. If any concentration among $e_3, e_4, ..., e_m$ is even, then we obtain a safe mixing pair from Lemma 18. Otherwise, $e_3, e_4, \ldots, e_m$ are all odd. This set has $m - 2 = n - 4$ droplets. Thus, by Lemma 16(b), there is at least one concentration e_i, for $i \in \{3, 4, ..., m\}$, such that any pair (e_i, e_j), for $j \in \{3, 4, ..., m\} - \{i\}$, is $\bar{p}$-safe. Let $E' = E - \{e_i, e_j\} \cup \{e, e\}$ be obtained from mixing e_i and e_j, for $e = \frac{1}{2}(e_i + e_j)$. By the choice of e_i, E' satisfies (I.2). E' satisfies (I.1) because it has at least two droplets with concentration e_1 and at least two droplets with concentration e_2.

■

Completing the proof. We are now ready to complete the proof that Condition (MC) in Theorem 5(a) is sufficient for perfect mixability when $n \geq 7$. The argument follows the outline given at the beginning of this section and depicted in Figure 3.3.

Assume that C satisfies Condition (MC). As described earlier in the proof, we first replace C by configuration $\check{C} \sqsubset \mathbb{Z}$ such that (i) $\check{\mu} = \mathsf{ave}(\check{C}) \in \mathbb{Z}$, all values in $\check{C}$ are even, and $\check{C}$ is $\bar{p}$-incongruent, and (ii) C is perfectly mixable with precision at most 1 if and only if $\check{C}$ is perfectly mixable with precision 0.

Then we show that $\check{C}$ has a perfect-mixing sequence (with precision 0), converting $\check{C}$ into its perfect mixture $\{n : \check{\mu}\}$. To this end, we first perform some mixing operations (at most two) that convert $\check{C}$ into a configuration E that either satisfies Invariant (I) or is near-final. If this E is near-final, we can complete the mixing sequence using Lemma 13. If this E is not near-final, then condition (I.2) implies that E satisfies Condition (MC) which, in turn, by Lemma 14, implies that $\|E\| \geq 3$. Therefore, depending on the value of $\|E\|$, we can apply one of Lemmas 20, 21, or 22, to show that E has a safe mixing pair, namely a pair of different concentrations whose mixing either preserves Invariant (I) or produces a near-final configuration. We can thus apply the above argument repeatedly to E. As in Section 3.4.1, each mixing decreases the value of $\Psi(E) = \sum_{e \in E}(e - \check{\mu})^2$. Thus after a finite number of steps we eventually convert E into a near-final configuration, that has a mixing sequence by Lemma 13.

3.4.3 Proof for $n = 5$

In this sub-section we prove that Condition (MC) in Theorem 5(a) is sufficient for perfect mixability when $n = 5$. The overall argument is similar to the case $n \geq 7$ we considered in Section 3.4.2 (and depicted on Figure 3.3), although this time we need a slightly different invariant. This is because in the case when $n = 5$ there are configurations that satisfy Invariant (I) but do not contain any pair of concentrations whose mixing preserves Invariant (I).

For example, $E = \{0, 0, 4, 4, 7\}$ with $\mathsf{ave}(E) = 3$ satisfies Invariant (I). The only pair of different concentrations with the same parity is $(0, 4)$; however, after mixing these concentrations, the new configuration will violate condition (I.1).

Let $A \sqsubset \mathbb{Z}$ be a configuration with $n = |A| = 5$ and $\mathsf{ave}(A) \in \mathbb{Z}$. We say that A is *blocking* if $A = \{n-2 : a_1, a_2, a_3\}$ where $a_1 \neq \frac{1}{2}(a_2 + a_3)$ and a_1 has parity different than a_2, a_3. Otherwise we say that A is *non-blocking.* For example, $A = \{0,0,0,3,7\}$, with $\mathsf{ave}(A) = 2$, is blocking. The intuition is that this A has only one pair of same-parity different concentrations, namely $(3,7)$, but this pair is not $\bar{p}$-safe – mixing 3 and 7 produces configuration $A' = \{0,0,0,5,5\}$ that is 5-congruent (in fact, it also violates Condition (MC)).

We say that A satisfies Invariant (I') if A meets the following three conditions:

(I.0) $A \sqsubset \mathbb{Z}$ and $\mathsf{ave}(A) = \check{\mu}$,

(I.1') A is non-blocking, and

(I.2) A is $\bar{p}$-incongruent.

At this point, we observe that, although we considered the case $n \geq 7$ in the previous section, the claims in Lemma 16(a) and (c) hold also for $n = 5$, and we will be using them in the proof. (Lemma 16(b) does not hold for $n = 5$, however.) We will also frequently use Observation 15 that follows directly from Lemma 14.

So, assume that we are given a configuration C with $C \cup \{\mu\} \sqsubset \mathbb{Z}$ satisfying Condition (MC). As in Section 3.4.2, we start by converting C into a configuration $\check{C} \sqsubset \mathbb{Z}$, with $\check{\mu} = \mathsf{ave}(\check{C}) \in \mathbb{Z}$, such that (i) all concentrations in $\check{C}$ are even, (ii) $\check{C}$ is $\bar{p}$-incongruent, and (iii) C is perfectly mixable with precision at most 1 if and only if $\check{C}$ is perfectly mixable with precision 0.

We then simply take $E = \check{C}$ (unlike in Section 3.4.2, we don't need to modify $\check{C}$). It thus remains to show that E is perfectly mixable with precision 0. By the properties of $\check{C}$, E satisfies conditions (I.0) and (I.2) and, since all concentrations in E are even, it also satisfies (I.1'). Thus E satisfies Invariant (I') and the rest of the proof is devoted to constructing a sequence of mixing operations that preserve Invariant (I') until E becomes near-final, which can be mixed perfectly with Lemma 13.

Preserving Invariant (I'). Assume that E is not near-final and that it satisfies Invariant (I'). We modify the definition of a safe mixing pair: in this proof we will say that a pair of concentrations $\{e, e'\} \subset E$ is *safe*, if mixing e and e' produces a configuration that either satisfies Invariant (I') or is near-final. We now show that each configuration E that satisfies Invariant (I') has a safe mixing pair. As we will always choose a pair of concentrations with the same parity for mixing, condition (I.0) will be trivially preserved, so in the proofs below we focus on explaining why the other two conditions are preserved.

Lemma 23 *Assume that $\|E\| = 3$ and that E is not near-final. If E satisfies Invariant (I') then E has a safe mixing pair.*

Proof. Let $E = \{f_1 : e_1, f_2 : e_2, f_3 : e_3\}$. By symmetry and reordering, respectively, we can assume that e_1 is even and that $f_1 \geq f_2 \geq f_3$. We analyze two cases based on f_1's value:

Case 1: $f_1 = 2$. Thus, $f_2 = 2$ and $f_3 = 1$. We consider two sub-cases.

Case 1.1: e_2 is even. We will mix e_1 and e_2, producing $E' = \{e_1, e_2, e_3, 2 : e\}$, for $e = \frac{1}{2}(e_1 + e_2)$. Since E is not near-final, $e \neq e_3$ (and obviously $e \notin \{e_1, e_2\}$). Since $f_1 = 2$, by Lemma 16(c), E' satisfies condition (I.2). Further, as $\|E'\| = 4$, E' also satisfies condition (I.1'). Therefore, (e_1, e_2) is indeed a safe mixing pair.

Case 1.2: e_2 is odd. Without loss of generality we can assume that e_3 is even (by the odd-even symmetry between e_1 and e_2.) We mix e_1 and e_3, and let E' be the resulting configuration. Since $f_1 = 2$, by Lemma 16(c), pair (e_1, e_3) is $\bar{p}$-safe, so E' satisfies condition (I.2). This, together with Lemma 14 (as $n = 5$ is prime), implies that $\|E'\| > 2$, which means that $\frac{1}{2}(e_1 + e_3) \neq e_2$, implying in turn that E' has two non-singletons. Thereby, E' satisfies (I.1') and thus we can conclude that (e_1, e_3) is a safe mixing pair.

Case 2: $f_1 = 3$. Thus, $f_2 = f_3 = 1$. Since E satisfies (I.1'), and by the assumption that E is not near-final, we have that at least one of e_2 and e_3 is even. By symmetry, we can assume that e_2 is even. We mix e_1 and e_2, and let E' be the new configuration. As $f_1 = 3$, by Lemma 16(c), E' satisfies (I.2). Further, $\frac{1}{2}(e_1 + e_2) \neq e_3$, because otherwise we would have $\|E'\| = 2$, contradicting Lemma 14. So E' contains two non-singletons, and thus it satisfies condition (I.1'). Therefore (e_1, e_2) is a safe mixing pair. ■

Lemma 24 *Assume that $\|E\| = 4$ and that E is not near-final. If E satisfies Invariant (I') then E has a safe mixing pair.*

Proof. Let $E = \{f_1 : e_1, f_2 : e_2, f_3 : e_3, f_4 : e_4\}$. By symmetry and reordering, we can assume that e_1 is even and that $f_1 = 2$ and $f_2 = f_3 = f_4 = 1$. We analyze three cases based on the parities of e_2, e_3, e_4.

Case 1: At least two of e_2, e_3, e_4 are even. Assume without loss of generality that e_2, e_3 are even. Let $e = \frac{1}{2}(e_1 + a_2)$. By symmetry, we can assume that $e \neq e_4$ (for otherwise we can take $e = \frac{1}{2}(e_1 + e_3)$ instead). Let $E' = E - \{e_1, e_2\} \cup \{e, e\}$ be obtained by mixing e_1 and e_2. As $f_1 = 2$, E' satisfies (I.2) by Lemma 16(c). If $e \neq e_3$ then $\|E'\| = 4$, so E' satisfies (I.1'). If $e = e_3$ then $E' = \{e_1, 3 : e_3, e_4\}$ where e_1, e_3 are even, so E' satisfies (I.1') as well. Thus (e_1, e_2) is a safe mixing pair.

Case 2: Exactly one of e_2, e_3, e_4 is even. Assume without loss of generality that e_2 is even. Let E' be obtained by mixing e_1 and e_2, also let $e = \frac{1}{2}(e_1 + e_2)$. E' satisfies (I.2) by Lemma 16(c). If $e \in E'$ then $e \in \{e_3, e_4\}$, where both e_3, e_4 are odd. Else, $e \notin E'$ and $\|E'\| = 4$. Either way E' satisfies (I.1') and (e_1, e_2) is a safe mixing pair.

Case 3: e_2, e_3, e_4 are all odd. By Lemma 16(a) there is at most one 5-unsafe pair. Assume that (e_2, e_3) is 5-safe. (Otherwise use e_4 instead of e_3.) Mixing e_2 and e_3 produces E' that preserves condition (I.2). Moreover, $\frac{1}{2}(e_2 + e_3) \in E$ implies $\|E'\| = 2$ (contradicting Lemma 14), so instead, $\frac{1}{2}(e_2 + e_3) \notin E$ and $\|E'\| = 4$. Thus, E' preserves (I.1') and (e_2, e_3) is a safe mixing pair. ■

Lemma 25 *Assume that $\|E\| = 5$ and that E is not near-final. If E satisfies Invariant (I') then E has a safe mixing pair.*

Proof. Let $E = \{e_1, e_2, e_3, e_4, e_5\}$. By symmetry and reordering, we can assume that e_1, e_2, e_3 have the same parity, say even. Additionally, by Lemma 16(a), there is at most one 5-unsafe pair in E, so we can assume that it does not involve e_1. In other words, all pairs involving e_1 and any other even concentration are 5-safe.

We now have three cases, given below. In each case we mix e_1 with some other even concentration, so conditions (I.0) and (I.2) will be satisfied, and we only need to ensure that (I.1') is satisfied as well.

Case 1: e_4 and e_5 are even. Choose $i \neq 1$ for which $|e_1 - e_i|$ is minimized, and let $e = \frac{1}{2}(e_1 + e_i)$. Then $e \notin E$. Mixing e_1 and e_i gives us configuration $E' = E - \{e_1, e_i\} \cup \{e, e\}$ with $\|E'\| = 4$, so E' satisfies (I.1').

Case 2: e_4 and e_5 have different parity. Say that e_4 is even and e_5 is odd. Let $e = \frac{1}{2}(e_1+e_2)$. We can assume that $e \neq e_5$ (otherwise, swap e_2 with e_3). Mixing e_1 and e_2 gives us configuration $E' = E - \{e_1, e_2\} \cup \{e, e\}$. If $e \notin E$, then $\|E'\| = 4$. Otherwise, $e \in \{e_3, e_4\}$; say that $e = e_3$. Thus $E' = \{3 : e_3, e_4, e_5\}$. Since e_4 and e_5 have different parity, E' is non-blocking. Thus in both subcases E' satisfies (I.1').

Case 3: e_4 and e_5 are odd. By symmetry, we can assume that $|e_1 - e_2| \leq |e_1 - e_3|$. Then $e = \frac{1}{2}(e_1 + e_2) \neq e_3$. If $e \notin E$, then $\|E'\| = 4$. Otherwise, $e \in \{e_4, e_5\}$; say that $e = e_4$, by symmetry. Then $E' = \{e_3, 3 : e_4, e_5\}$, and e_3, e_5 have different parity, so E' is non-blocking. Thus in both subcases E' satisfies (I.1'). ■

Completing the proof. We can now prove that Condition (MC) in Theorem 5(a) is sufficient for perfect mixability when $n = 5$. Assume that C satisfies Condition (MC). As described earlier in this section, we convert C into configuration $\check{C} \sqsubset \mathbb{Z}$ such that C is perfectly mixable with precision at most 1 if and only if $\check{C}$ is perfectly mixable with precision 0. This $\check{C}$ is $\bar{p}$-incongruent, all its concentrations are even, and it satisfies $\check{\mu} = \mathsf{ave}(\check{C}) \in \mathbb{Z}$. Thus, if we take $E = \check{C}$, this E satisfies Invariant (I'). If E is near-final, then we use Lemma 13 to perfectly mix E. If this E is not near-final, then, depending on the value of $\|E\|$, we can apply one of Lemmas 23, 24, or 25, to show that E has a safe mixing pair. As in Section 3.4.2, the value of $\Psi(E)$ decreases at least by 1 after each mixing operation. So after a finite sequence of mixing operations E must become near-final.

3.4.4 Proof for $n = 6$

The proof that Condition (MC) in Theorem 5(a) is sufficient for perfect mixability when $n = 6$ is derived from both Section 3.4.2 and Section 3.4.3.

First of all, as in Section 3.4.3, Invariant (I) cannot always be preserved, so we use Invariant (I') instead. For example, $E = \{0, 0, 6, 6, 5, 7\}$ with $\mathsf{ave}(E) = 4$ satisfies Invariant (I). However, mixing 0 and 6 violates condition (I.1), mixing 5 and 7 violates condition (I.2) (in fact, it also violates Condition (MC)), and any other mixing that involves two distinct concentrations violates condition (I.0).

The outline of the proof for $n = 6$ is the same, given $C \cup \{\mu\} \sqsubset \mathbb{Z}$ satisfying Condition (MC), we construct $\check{C}$ such that C is perfectly mixable with precision at most 1 if and only if $\check{C}$ is perfectly mixable with precision 0. Then, using $\check{C}$, we construct E that satisfies Invariant (I'). After this, we just continuously mix safe mixing pairs until E becomes near-final.

To prove that there is always a safe mixing pair, we can slightly modify Lemmas 23, 24, and 25 such that they hold for $n = 6$. This can be done because both 5 and 6 have exactly one odd prime factor and because Observation 15 also holds for $n = 6$.

3.5 Polynomial Bound for the Number of Mixing Operations

Let $C \sqsubset \mathbb{Z}$ with $\mathsf{ave}(C) \in \mathbb{Z}$ and $|C| = n$ be a configuration that satisfies Condition (MC). The existence of a perfect-mixing graph for C was established in Section 3.4. This graph, however, might be very large – it can be shown that if arbitrary droplets are mixed at each step then it might take an exponential number of steps for the process to converge. In this section we prove Theorem 5(b), namely, that C can be perfectly mixed with precision at most 1 and in a polynomial number of steps. (We already proved that, for $n \leq 3$, C can be perfectly mixed after at most one mixing operation and with precision

0. Hence we will be assuming that $n \geq 4$.) The essence of the proof is to show that in the construction in Section 3.4 it is possible to choose a mixing operation at each step so that the overall number of steps will be polynomial in the input size.

It is sufficient to show that configuration E, constructed from C as in Section 3.4.2, is perfectly mixable with precision at most 0 in a polynomial number of mixing operations. (Constructing such E from C can easily be done in a polynomial number of steps.) For this reason we will assume that E is the initial configuration. Using the same notation as in Section 3.4.2, let $\hat{\mu} = \mathsf{ave}(E)$. Specifically, we will show that $\Psi(E) = \sum_{e \in E}(e - \hat{\mu})^2$ can be decreased down to 0 after a polynomial number of mixing operations, while preserving $E \cup \{\hat{\mu}\} \sqsubset \mathbb{Z}$ after every such mixing.

The general idea is to always mix two concentrations whose difference is big enough such that, after a polynomial number of mixing operations, $\Psi(E)$ decreases by at least a factor of two. It follows that $\Psi(E)$ can be decreased down to 0 after a polynomial number of mixing operations.

Additionally, to guarantee that E is perfectly mixed with precision at most 0, we will only mix pairs of concentrations that are safe, until E becomes near-final; recall that a safe pair is a pair of distinct concentrations whose mixing preserves the corresponding invariant. (If n is a power of two then we only need to mix two same-parity concentrations.) Once E is near-final, it is simple to show a polynomial sequence of mixing operations to perfectly mix E preserving $E \cup \{\hat{\mu}\} \sqsubset \mathbb{Z}$.

3.5.1 Auxiliary Observations

Let $A \sqsubset \mathbb{Z}$ with $\mathsf{ave}(A) = \mu_A \in \mathbb{Z}$ and $|A| = n_A$ be an arbitrary configuration. We define $\min(A)$ and $\max(A)$ as the lowest and highest concentrations in A, respectively. We also define the diameter of A as $\mathsf{diam}(A) = \max(A) - \min(A)$. Recall that the bit size of A is $s(A) = \sum_{a \in A} \log(|a|+2)$. We denote by A_{even} and A_{odd} two disjoint multisets containing all even and odd concentrations in A, respectively; we will often write A_π and $A_{\bar{\pi}}$, with $\pi, \bar{\pi} \in \{even, odd\}$ and $\pi \neq \bar{\pi}$, as a simplification for A_{even} and A_{odd}. We lastly define $\Psi_{A,0}$ as the value of $\Psi(A)$ before any mixing has been performed.

The following observations show that mixing two droplets with concentrations that are sufficiently far apart eventually decreases $\Psi(A)$ by a factor at least two.

Observation 26 *Let $x, y \in A$ such that $|x - y| \geq \mathsf{diam}(A)/\gamma$ for constant $\gamma \in \mathbb{Z}_{>0}$. Mixing x and y decreases $\Psi(A)$ by at least $\Psi(A)/2\gamma^2 n_A$.*

Proof. Assume without loss of generality that $\mu_A = 0$ so that $\Psi(A) = \sum_{a \in A} a^2$. Let A' be obtained after mixing x and y. We have that $\Psi(A)$ decreases by:

$$\Psi(A) - \Psi(A') = x^2 + y^2 - 2\Big(\frac{x+y}{2}\Big)^2 = \frac{(x-y)^2}{2} \geq \frac{\mathsf{diam}(A)^2}{2\gamma^2} \geq \frac{\Psi(A)}{2\gamma^2 n_A}$$

■

Observation 27 *Repeatedly mixing $x, y \in A$ satisfying $|x - y| \geq \mathsf{diam}(A)/\gamma$, for constant $\gamma \in \mathbb{Z}_{>0}$, decreases $\Psi(A)$ by at least $\Psi(A)/2$ after at most $2\gamma^2 n_A$ mixing operations.*

Proof. Assume that $a, b \in A$ satisfy $|a - b| \geq \mathsf{diam}(A)/\gamma$. By Observation 26, mixing a and b produces A' such that $\Psi(A) - \Psi(A') \geq \Psi(A)/2\gamma^2 n_A$; in other words, $\Psi(A') \leq$

$\Psi(A)\big(1 - \frac{1}{2\gamma^2 n_A}\big)$. So, repeatedly mixing $x, y \in A$ satisfying $|x - y| \geq \mathsf{diam}(A)/\gamma$ (for corresponding $\mathsf{diam}(A)$ per mix), for at most $2\gamma^2 n_A$ times produces A'' such that

$$\Psi(A'') \leq \Psi(A)\Big(1 - \frac{1}{2\gamma^2 n_A}\Big)^{2\gamma^2 n_A} \leq \frac{\Psi(A)}{2}$$

which holds because $(1 - 1/x)^x \leq 1/2$ for $x \geq 2$. (In fact, $(1 - 1/x)^x \longrightarrow 1/e$ as x goes to infinity.) ■

The next observations will be used (without any explicit reference) as plain arithmetic simplifications.

Observation 28 *$\log \Psi(A) \leq 2s(A)$.*

Proof. First we show that the inequality holds when $\mu_A = 0$ and then we show that it holds for arbitrary μ_A. Assume that $\mu_A = 0$. Then:

$$\log \Psi(A) = \log \sum_{a \in A} a^2 \leq \log \Big(\sum_{a \in A} \big(|a| + 2\big)\Big)^2 \leq 2 \log \prod_{a \in A} \big(|a| + 2\big) \leq 2 \sum_{a \in A} \log(|a| + 2) \leq 2s(A)$$

For μ_A arbitrary, consider the function $f(A, x) = \sum_{a \in A}(a - x)^2$. Taking the derivative of f with respect to x gives us that the function is minimized when $x = \mu_A$. This, and $f(A, 0) = \sum_{a \in A} a^2$ implies that the argument for $\mu_A = 0$ is sufficient for arbitrary μ_A. ■

Observation 29 *Let $A' \subseteq A$. Then $\Psi(A') \leq \Psi(A)$.*

Proof. Let $\mu_{A'} = \mathsf{ave}(A')$. Consider the function $g(x) = \sum_{a \in A'}(a - x)^2$. This function is minimized when $x = \mu_{A'}$. (This can be shown by computing the derivative of g with respect to x.) From this we get that $g(\mu_A) \geq g(\mu_{A'}) = \Psi(A')$. Further, $g(\mu_A) \leq \Psi(A)$ because $g(\mu_A)$ is a sum of some terms in $\Psi(A)$. Thus, $\Psi(A') = g(\mu_{A'}) \leq g(\mu_A) \leq \Psi(A)$. ■

Observation 30 *Let A' be obtained from A after a mixing operation. Then $\Psi(A') \leq \Psi(A)$.*

Proof. Let $a, b \in A$ be the concentrations of the mixed droplets. Then:

$$\Psi(A) - \Psi(A') = (a - \mu_A)^2 + (b - \mu_A)^2 - 2\Big(\frac{a+b}{2} - \mu_A\Big)^2 = \frac{(a-b)^2}{2}$$

As $(a-b)^2/2$ is non-negative, we have that $\Psi(A') \leq \Psi(A)$. (Note that $\mathsf{ave}(A') = \mu_A$.) ■

3.5.2 Proof for n Power of Two

In this sub-section we prove that E can be perfectly mixed with precision at most 0 in a polynomial number of mixing operations when $|E| = n$ is a power of two. (Note that n power of two implies that there are no odd prime factors, so mixing droplets with the same-parity furthest-apart concentrations is sufficient.)

Let $\pi, \bar{\pi} \in \{even, odd\}$ with $\pi \neq \bar{\pi}$. If any mixing on E_π produces concentrations with parity $\bar{\pi}$, then such concentrations are excluded from E_π and included into $E_{\bar{\pi}}$. Consider Lemma 31 below.

Lemma 31 *After at most $4ns(E)$ furthest-apart mixing operations in E_π, either $E_\pi = \emptyset$ or $\|E_\pi\| = 1$.*

Proof. Recall that $\Psi_{E_\pi,0} = \Psi(E_\pi)$ before any mixing has been performed. Let $|E_\pi| = n_\pi$. By Observation 27, repeatedly mixing $\min(E_\pi)$ with $\max(E_\pi)$ decreases $\Psi(E_\pi)$ by at least $\Psi(E_\pi)/2$ after at most $2n_\pi$ mixing operations; every mixing operation involves two concentrations with difference $\mathsf{diam}(E_\pi)$. Thus, after at most $2n_\pi \log \Psi_{E_\pi,0}$ mixing operations, either $E_\pi = \emptyset$ or $\|E_\pi\| = 1$ hold. Finally, $n_\pi \leq n$ and $\Psi_{E_\pi,0} \leq \Psi_{E,0}$ so $2n_\pi \log \Psi_{E_\pi,0} \leq 2n \log \Psi_{E,0} \leq 4ns(E)$. ■

Even though Lemma 31 is not useful on its own, it helps us analyze a mixing strategy from a more general point, which is presented in Lemma 32 below.

Lemma 32 *After any sequence of at most $8ns(E)$ same-parity furthest-apart mixing operations $\Psi(E)$ decreases by at least $\Psi(E)/18n$.*

Proof. Let $\delta = \mathsf{diam}(E)$ before any mixing operation. If the same-parity furthest-apart pair $x, y \in E$ satisfies $|x - y| \geq \delta/3$, then by Observation 26, mixing x and y decreases $\Psi(E)$ by at least $\Psi(E)/18n$. Otherwise, consider E_{even} and E_{odd} non-empty. Assume without loss of generality that $\max(E_{odd}) > \max(E_{even})$. This, and the difference between every same-parity concentration pair being less than $\delta/3$ implies that $\min(E_{odd}) > \max(E_{odd}) - \delta/3$, $\min(E) = \min(E_{even})$ and $\max(E_{even}) < \min(E_{even}) + \delta/3$. Hence, $|\min(E_{odd}) - \max(E_{even})| \geq \delta/3$.

Now, let $n' = |E_{even}|$ such that $n' < n$. By Lemma 31, it takes at most $4n's(E_{even}) < 4ns(E)$ furthest-apart mixing operations on E_{even} for either $E_{even} = \emptyset$ or $\|E_{even}\| = 1$ to hold (similarly for E_{odd}). Thus, if we repeatedly mix the droplets with the same-parity furthest-apart concentrations in E, eventually, after fewer than $8ns(E)$ such mixing operations, either a mixing in E_{even} produces an odd concentration or a mixing in E_{odd} produces an even concentration. (This is true because, as mentioned in the proof for Lemma 13, n power of two guarantees the existence of two distinct concentrations with same-parity, so we cannot have both $\|E_{even}\| = 1$ and $\|E_{odd}\| = 1$.)

So, if x odd was produced from a mixing in E_{even}, then we mix x with $\max(E_{odd})$. Otherwise, y even was produced from a mixing in E_{odd} and we mix y with $\min(E_{even})$. As both $|x - \max(E_{odd})|$ and $|y - \min(E_{even})|$ are at least $\delta/3$, either mixing decreases $\Psi(E)$ by at least $\Psi(E)/18n$ (see Observation 26), and thus the lemma holds. ■

Lemma 32 leads to a complete proof shown in Theorem 33 below.

Theorem 33 *If $|E| = n$ is a power of two, then E can be perfectly mixed with precision at most* 0 *after no more than $288n^2s^2(E)$ mixing operations.*

Proof. By Lemma 32, $\Psi(E)$ decreases by at least $\Psi(E)/18n$ after a mixing sequence of at most $8ns(E)$ same-parity furthest-apart mixing operations. It follows from Observation 27 that after at most $18n$ such mixing sequences, $\Psi(E)$ decreases by at least a factor of two. Consequently, after at most $18n \log \Psi_{E,0}$ such mixing sequences, E becomes perfectly mixed. Finally, each mixing sequence takes at most $8ns(E)$ mixing operations, thus the total number of mixing operations is at most $288n^2s^2(E)$. ■

Theorem 34 *If E is near-final, then E can be perfectly mixed, with precision at most* 0*, after no more than $144n^3s^2(E)$ mixing operations.*

Proof. Assume that E is near-final. By definition, E can be partition into at most $n/2$ disjoint $E' \sqsubseteq E$ satisfying $|E'| > 1$ power of two with $\mathsf{ave}(E') = \mathsf{ave}(E)$. Each disjoint E' can be perfectly mixed in at most $288n^2s^2(E)$ mixing operations (see Theorem 33 above), thus the theorem holds. ■

3.5.3 An Exponential Bound

In this sub-section we give a simple upper bound on the number of mixing operations to perfectly mix E with precision at most 0 when $|E| = n \geq 5$. This bound is exponential in n — so it's too weak for our purpose — but we will need it to bound the number of mixing steps when n is at most 22 (see Theorem 37).

Let E satisfy Invariant (λ); $\lambda = I'$ for $n = 5, 6$ and $\lambda = I$ for $n \geq 7$. Also, let $A \subseteq E$. We say that a pair of distinct concentrations in A is good if it is safe with respect to E. (Any mixing in A involving a good pair preserves Invariant (λ) on E.) If there is no good pair in A then we say that A is *E-mixed*. Additionally, since E satisfies Invariant (λ), either E is near-final or the existence of a safe pair in E is guaranteed. (This was shown in Sections 3.4.2 and 3.4.3.)

First, in Lemma 35 below we show an upper bound on the number of mixing operations to E-mix $A \subseteq E$. Then, using such results we give an upper bound in Theorem 36 for the number of mixing operations to perfectly mix E with precision at most 0.

Lemma 35 *Assume that E satisfies Invariant (λ). If $A \subseteq E$ with $|A| = k$, then A can be E-mixed with precision at most* 0 *in no more than* $(8k^3 s(A))^k$ *mixing operations.*

Proof. We prove the lemma by induction with respect to k. Let $\phi(A)$ denote the number of furthest-apart good mixing operations to E-mix A.

Base case: If $k \leq 1$, then $\phi(A) = 0$, since A is trivially E-mixed.

Inductive step: Assume that every $A' \subset A$ with $|A'| = k' < k$ can be E-mixed in $\phi(A') \leq (8{k'}^3 s(A'))^{k'}$ mixing operations. We next show that A can be E-mixed in $\phi(A) \leq (8k^3 s(A))^k$ mixing operations.

If A is E-mixed then we are done. Otherwise, let $x, y \in A$ be the furthest-apart good pair. If $|x - y| \geq \mathsf{diam}(A)/k$, then mixing x and y decreases $\Psi(A)$ by at least $\Psi(A)/2k^3$ (see Observation 26). Then, by Observation 27, after at most $2k^3$ such mixing operations

$\Psi(A)$ decreases by a factor at least two. It follows that after at most $2k^3 \log \Psi_{A,0}$ such mixing operations, A becomes E-mixed. Nevertheless, such x and y do not always satisfy $|x - y| \geq \mathsf{diam}(A)/k$. Consequently, we next show a strategy that bounds the number of mixing operations satisfying $|x - y| < \mathsf{diam}(A)/k$.

Let $\delta = \mathsf{diam}(A)$ before any mixing operation. Divide the interval $[\min(A), \max(A)]$ into k equal segments such that for at least one segment with interval $[l, r]$, no concentration in A lies withing the open interval (l, r). Split A into A_1 and A_2 such that $\max(A_1) \leq l$ and $\min(A_2) \geq r$. By our inductive assumption, A_1 and A_2 can be E-mixed in $\phi(A_1), \phi(A_2) < (8k^3 s(A))^{k-1}$ mixing operations, respectively. Therefore, after at most $\phi(A_1) + \phi(A_2)$ mixing operations, either A becomes E-mixed or there is a good pair $x \in A_1$ and $y \in A_2$ that satisfies $|x - y| \geq \delta/k$.

It follows that A can be E-mixed in at most $(8k^3 s(A))^k$ mixing operations:

$$\phi(A) \leq 2k^3 \log \Psi_{A,0} \cdot \big(\phi(A_1) + \phi(A_2) + 1\big) \leq 4k^3 s(A) \cdot 2(8k^3 s(A))^{k-1} \leq (8k^3 s(A))^k$$

■

Theorem 36 *If $|E| = n \geq 5$, then E can be perfectly mixed after at most $(8n^3 s(E))^n + 144n^3 s^2(E)$ mixing operations.*

Proof. Let E satisfy Invariant (λ), for $\lambda \in \{I, I'\}$. We know that if E satisfies Invariant (λ) then E either has a safe pair or it is near-final. Therefore, we first E-mix E using Lemma 35 and then we perfectly mix it using Theorem 34. The total number of mixing operations is thus at most $(8n^3 s(E))^n + 144n^3 s^2(E)$. ■

3.5.4 A Polynomial Bound

In this sub-section we prove in Theorem 37 below that E can be perfectly mixed with precision at most 0 in a polynomial number of mixing operations. (For simplicity, we will be using Observation 26 without explicit reference.)

Theorem 37 *E can be perfectly mixed with precision at most 0 in a polynomial number of mixing operations.*

Proof. If n is a power of two we can simply use Theorem 33 and we are done. Thus, we can assume that $n \geq 5$. If n is a small constant, say $n < 22$, we can perfectly mix E using Theorem 36, where the total number of mixing operations is a polynomial of constant degree. Otherwise, $n \geq 22$ (E satisfies Invariant (I)) and we analyze the following cases.

Assume that E is not perfectly mixed. Let $\gamma \geq 2$ be a small integral constant and $\delta = \mathsf{diam}(E)$ before any mixing operation. Assume by symmetry that $|E_\pi| \geq |E_{\bar{\pi}}|$.

Case 1: $\|E_\pi\| = 1$; thus $|E_{\bar{\pi}}| \geq 1$. Let $a \in E_\pi$. We consider two sub-cases:

Case 1.1: $\min(E_{\bar{\pi}}) < a < \max(E_{\bar{\pi}})$. After at most two safe mixing operations on $E_{\bar{\pi}}$, $\Psi(E)$ decreases by at least $\Psi(E)/32n$ (see Lemma 40).

Case 1.2: $a < \min(E_{\bar{\pi}})$ or $a > \max(E_{\bar{\pi}})$. After at most $128ns(E)+1$ safe mixing operations on $E_{\bar{\pi}}$, $\Psi(E)$ decreases by at least $\Psi(E)/32\gamma^2 n$ (see Lemma 41).

Case 2: $\|E_\pi\| \geq 2$ with $\mathsf{diam}(E_\pi) \geq \delta/\gamma$. After one safe mixing operation on E_π, $\Psi(E)$ decreases by at least $\Psi(E)/8\gamma^2 n$ (see Lemma 42).

Case 3: $\|E_\pi\| \geq 2$ with $\mathsf{diam}(E_\pi) < \delta/\gamma$. Let E' be obtained from E after E-mixing E_π, which takes at most $128ns(E)$ safe mixing operations (see Lemma 43); E'_π is E'-mixed. We consider two sub-cases:

Case 3.1: $|E'_\pi| \geq |E'_{\bar{\pi}}|$. By Observation 38 below and since E'_π is E'-mixed, we have that $\|E'_\pi\| = 1$. Hence, as in Case 1 above, after at most $128ns(E') + 1$ safe mixing operations on E'_π, $\Psi(E')$ decreases by at least $\Psi(E')/32\gamma^2 n$. (In other words, as $\Psi(E) > \Psi(E')$, $\Psi(E)$ decreases by at least $\Psi(E)/32\gamma^2 n$.)

Case 3.2: $|E'_\pi| < |E'_{\bar{\pi}}|$. As $\gamma \geq 2$, $\mathsf{diam}(E'_\pi) < \delta/\gamma$ and because the mixing operations on E_π produced at least one droplet with concentration $\bar{\pi}$, we have that $\mathsf{diam}(E'_{\bar{\pi}}) > \delta/\gamma$. Hence, as in Case 2 above, after one safe mixing operation on $E'_{\bar{\pi}}$, $\Psi(E')$ decreases by at least $\Psi(E')/8\gamma^2 n$. (As $\Psi(E) > \Psi(E')$, $\Psi(E)$ decreases by at least $\Psi(E)/8\gamma^2 n$.)

Applying a sequence of mixing operations specified by the cases above results in a decrease of $\Psi(E)$ by at least $\Psi(E)/32\gamma^2 n$. Thus, by a simple extension to Observation 27, we get that after at most $32\gamma^2 n$ such mixing sequences $\Psi(E)$ decreases by at least $\Psi(E)/2$. It follows that after at most $64\gamma^2 ns(E)$ mixing sequences, E either becomes near-final or perfectly mixed; E near-final can be perfectly mixed in at most $144n^3 s^2(E)$ mixing operations (see Theorem 34). Each mixing sequence takes at most $256ns(E) + 1$ mixing operations, therefore the total number of mixing operations to perfectly mix E is at most $2^{14}\gamma^2 n^2 s^2(E) + 64\gamma^2 ns(E) + 144n^3 s^2(E)$. ■

Observation 38 *Assume that E with $|E| = n \geq 22$ satisfies Invariant (I) and let π such that $|E_\pi| \geq |E_{\bar{\pi}}|$. There is at least one droplet in E_π such that, when paired with any other droplet in E_π, the resulting pair is safe.*

Proof. Lemma 16(a) and the number of distinct odd prime factors of n being less than $\log_3 n$ imply that E has at most $2\lfloor \log_3 n \rfloor$ droplets that are unsafe when paired with other droplets in E. This, and Observation 39 below give that the number of droplets that, when mixed with other droplets in E violate Invariant (I), is at most $2\lfloor \log_3 n \rfloor + 6 < n/2 \leq |E_\pi|$, since $n \geq 22$. Thus, the lemma holds. ■

Observation 39 *Assume that E with $|E| = n \geq 7$ satisfies Invariant (I). The number of droplets involved in mixing operations that decrease the number of non-singletons in E down to one is at most* 6*.*

Proof. If the number of non-singletons in E is more than three, then no mixing decreases the number of non-singletons down to one; similarly when mixing a non-singleton with frequency higher than two. Additionally, mixing two singletons does not decrease the number of non-singletons. Now, E satisfying Invariant (I) implies that there are at least two non-singletons $a, b \in E$. We consider two types of cases where a mixing decreases the number of non-singletons in E down to one:

Case 1: Mixing two non-singletons, say a and b. This can happen when the frequency of both a and b is two each, leading to a total of 4 droplets involved. (There could be another non-singleton $e = \frac{1}{2}(a + b)$ in E, however no mixing involving e decreases the number of non-singletons down to one because either a or b remains non-singleton after the mixing.)

Case 2: Mixing a non-singleton with a singleton. (This can happen when E has exactly two non-singletons, a and b respectively.) Let a and c be the non-singleton and singleton, respectively. If a is a doubleton and $b = \frac{1}{2}(a+c)$, then mixing a and c decreases the number of non-singletons down to one. Similarly, if b is a doubleton and $a = \frac{1}{2}(b + d)$, for some singleton $d \in E$, then mixing b and d decreases the number of non-singletons down to one. Therefore, the total number of droplets (excluding a and b, which were already counted in Case 1 above) is at most 2.

Therefore, the number of droplets involved in mixing operations that decrease the number of non-singletons down two one is at most 6. ■

Lemma 40 *Assume that E with $|E| = n \geq 7$ satisfies Invariant (I). Also, assume that $|E_\pi| \geq |E_{\bar{\pi}}| \geq 1$ and that $\|E_\pi\| = 1$, with $a \in E_\pi$. If $\min(E_{\bar{\pi}}) < a < \max(E_{\bar{\pi}})$ then, after at most two mixing operations, $\Psi(E)$ decreases by at least $\Psi(E)/32n$.*

Proof. Let $\delta = \mathsf{diam}(E)$ before any mixing operation. Assume without loss of generality that $\pi = even$. Since E satisfies Invariant (I), there is at least one non-singleton $b \in E_{odd}$. Either $\min(E_{odd})$ or $\max(E_{odd})$ is furthest from b, so assume without loss of generality that $\min(E_{odd})$ is furthest from b; $|b - \min(E_{odd})| \geq \delta/2$. If $b, \min(E_{odd})$ is a safe pair then we mix them and we are done. Otherwise, E has exactly two non-singletons, $a = \frac{1}{2}(b + \min(E_{odd}))$ and b (which is a doubleton). Consider the following cases:

<u>Case 1</u>: $b = \max(E_{odd})$. Both E satisfying Invariant (I) and $\|E_\pi\| = 1$ imply that there is $c \in E_{odd}$ such that $\min(E_{odd}) < c < b$; mixing b and c produces a non-singleton other than a, so b, c is a safe pair. We analyze the following sub-cases:

<u>Case 1.1</u>: $c < a$. Since $|b - c| \geq \delta/2$, mixing b and c decreases $\Psi(E)$ by at least $\Psi(E)/8n$.

<u>Case 1.2</u>: $c > a$. Let $d = \frac{1}{2}(b + c)$ be the output of the mixing between b and c. If d is odd, then mixing d and $\min(E_{odd})$ decreases $\Psi(E)$ by at least $\Psi(E)/8n$. (Pair $d, \min(E_{odd})$ is a safe pair because their mixing produces a non-singleton other than a.) Otherwise,

d is even and, as $|d-a| \geq \delta/4$, mixing a and d decreases $\Psi(E)$ by at least $\Psi(E)/32n$. (Note that a, d is a safe pair because, after the mixing, a is still a non-singleton.)

Case 2: $b < \max(E_{odd})$. This is similar to Case 1.2 above under the assumption that $b < \max(E_{odd})$ and using $c = \max(E_{odd})$. ■

Lemma 41 *Assume that E with $|E| = n \geq 7$ satisfies Invariant (I). Also, assume that $|E_{\pi}| \geq |E_{\bar{\pi}}| \geq 1$ and that $\|E_{\pi}\| = 1$ with $a \in E_{\pi}$. If either $a < \min(E_{\bar{\pi}})$ or $a > \max(E_{\bar{\pi}})$, then, after at most $128ns(E) + 1$ safe mixing operations, $\Psi(E)$ decreases by at least $\Psi(E)/32\gamma^2 n$, for constant $\gamma \in \mathbb{Z}_{>0}$.*

Proof. Let $\delta = \mathsf{diam}(E)$ before any mixing operation. Assume without loss of generality that $\pi = even$ and that $a < \min(E_{odd})$. We analyze two sub-cases:

Case 1: $\mathsf{diam}(E_{odd}) \geq \delta/2\gamma$. E satisfying Invariant (I) implies that there is a non-singleton $b \in E_{odd}$. Either $\min(E_{odd})$ or $\max(E_{odd})$ is furthest from b, so assume without loss of generality that $\min(E_{odd})$ is furthest from b. As $a < \frac{1}{2}(b + \min(E_{odd}))$, pair $b, \min(E_{odd})$ is a safe pair (mixing b and $\min(E_{odd})$ produces a non-singleton other than a) that satisfies $|b - \min(E_{odd})| \geq \delta/4\gamma$. Therefore, mixing b and $\min(E_{odd})$ decreases $\Psi(E)$ by at least $\Psi(E)/32\gamma^2 n$.

Case 2: $\mathsf{diam}(E_{odd}) < \delta/2\gamma$. This implies that $|\min(E_{odd}) - a| \geq \delta/2\gamma$. E-mix E_{odd} using Lemma 43; E satisfying Invariant (I) implies that an even concentration b is eventually produced. Since a's frequency is at least $\lceil n/2 \rceil \geq 3$, pair a, b is a safe pair; a is still a non-singleton after the mixing. Additionally, since $b > \min(E_{odd})$, $|b - a| > \delta/2\gamma$ and thus mixing a and b decreases $\Psi(E)$ by at least $\Psi(E)/8\gamma^2 n$. Finally, Lemma 43 takes at most $128ns(E)$ mixing operations, so the lemma holds. ■

Lemma 42 *Assume that E with $|E| = n \geq 22$ satisfies Invariant (I). Also, assume that $|E_\pi| \geq |E_{\bar{\pi}}|$ and that $\|E_\pi\| \geq 2$. If $\mathsf{diam}(E_\pi) \geq \mathsf{diam}(E)/\gamma$, for constant $\gamma \in \mathbb{Z}_{>0}$, then after one safe mixing operation $\Psi(E)$ decreases by at least $\Psi(E)/8\gamma^2 n$.*

Proof. Assume without loss of generality that $\pi = even$ and let $a = \min(E_{even})$ and $b = \max(E_{even})$. If a, b is a safe pair then mixing a and b decreases $\Psi(E)$ by at least $\Psi(E)/2\gamma^2 n$. Thus, assume that a, b is not a safe pair.

By Observation 38, and as a, b is not a safe pair, there is $c \in E_{even}$ with $c \notin \{a, b\}$ for which any pair involving c is a safe pair. Now, either a or b is furthest from c, so assume without loss of generality that a is furthest from c. Then, $|c - a| \geq \mathsf{diam}(E)/2\gamma$ and thus mixing a and c decreases $\Psi(E)$ by at least $\Psi(E)/8\gamma^2 n$. ■

Lemma 43 *Assume that E with $|E| = n \geq 7$ satisfies Invariant (I). Then E_π can be E-mixed after at most $128ns(E)$ mixing operations.*

Proof. Assume without loss of generality that $\pi = even$ and let $|E_{even}| = n'$ and $\delta = \mathsf{diam}(E_{even})$ before any mixing operation. If a mixing in E_{even} produces odd concentrations, then such concentrations are excluded from E_{even} and included into E_{odd}; this can happen at most $n'/2$ times, if so, then E_{even} becomes E-mixed.

Assume that the number of safe pairs (with respect to E) in E_{even} is non-zero and let $a = \min(E_{even})$ and $b = \max(E_{even})$. If a safe pair $x, y \in E_{even}$ satisfies $|x - y| \geq \delta/4$, then mixing x and y decreases $\Psi(E_{even})$ by at least $\Psi(E_{even})/32n'$. It follows from from Observation 27 that after at most $32n'$ such mixing operations, $\Psi(E_{even})$ decreases by at least $\Psi(E_{even})/2$. Hence, E_{even} can be E-mixed after at most $32n' \log \Psi(E_{even})$ such mixing operations.

We next show that if E_{even} has not been E-mixed, then after two safe mixing operations (with respect to E) either an odd concentration is produced or $\Psi(E_{even})$ decreases by at least $\Psi(E_{even})/32n'$. Consequently, after at most $128ns(E)$ safe mixing operations, E_{even} becomes E-mixed.

So, if a and b is a safe pair then we are done; $|a-b| = \delta$. Instead, assume that a, b is not a safe pair and consider the following cases:

Case 1: E_{even} has only singletons. This implies that no mixing in E_{even} decreases the number of non-singletons in E. So, let $c, d \in E_{even}$ be the furthest-apart safe pair and $x = \frac{1}{2}(c+d)$ be the output of their mixing. If either $|c-d| \geq \delta/2$ or x odd holds, then we are done. Otherwise, let $y \in \{a, b\}$ be furthest from x. By the choice of y, we have that $|x-y| \geq \delta/2$, and as x is a doubleton, x, y is a safe pair and thus we mix them.

Case 2: E_{even} has exactly one non-singleton c. Either a or b is furthest from c, so assume without loss of generality that b is furthest from c; $|b-c| \geq \delta/2$. If b, c is a safe pair then we mix it and we are done. Otherwise, b, c is not a safe pair, therefore there are exactly two non-singletons in E, $c \in E_{even}$ (which is a doubleton) and $e = (b+c)/2 \notin E_{even}$. Since there is a safe pair in E_{even}, there is some $d \in E_{even}$ such that $d \notin \{b, c\}$. Then, c, d is a safe pair because mixing c and d produces a non-singleton $x = \frac{1}{2}(c+d)$ other than e. So, if $|c-d| \geq \delta/2$ then we mix them and we are done; if x is odd then we are also done. Otherwise, b, x is a safe pair ($e \neq \frac{1}{2}(b+x)$) and $|x-b| \geq \delta/4$, so we mix it.

Case 3: E_{even} has at least two non-singletons $c < d$. We analyze two sub-cases:

Case 3.1: $a = c$. Since a, b is not a safe pair, then mixing b and c decreases the number of non-singletons down to one. This gives us two sub-cases:

Case 3.1.1: $d = b$. E_{even} having a safe pair implies that there is some $x \in E_{even}$ such that $x \notin \{c, d\}$. Either c or d is furthest from x, so assume without loss of generality that d is furthest from x; $|d - x| \geq \delta/2$. Mixing x and d produces a non-singleton other than c, so x, d is a safe pair and we mix it.

Case 3.1.2: $d < b$. This implies that $d = \frac{1}{2}(c + b)$, $b - d = \delta/2$ and b, d is a safe pair (mixing b and d produces a non-singleton other than c), so we mix b and d.

Case 3.2: $a < c$. If $|a - c| \geq \delta/2$ then we mix a and c and we are done; $\frac{1}{2}(a + c) < d$ implies that mixing a and c produces a non-singleton other than d, so a, c is a safe pair. Instead, let $|a-c| < \delta/2$ which implies $|b-c| \geq \delta/2$. Now, if pair b, c is safe then we mix it and we are done, so also assume that b, c is not a safe pair. (We cannot have $b = d$ because that would contradict a, b not being a safe pair; $|a - c| < \delta/2$ implies $c < \frac{1}{2}(a + b)$, so mixing a and b would produce a non-singleton other than c.)

Therefore, as b, c is not a safe pair, we have that $d = \frac{1}{2}(b + c)$ and mixing b and d produces a non-singleton other than c. Hence, b, d is a safe pair satisfying $|b-d| \geq \delta/4$, so we mix it.

■

3.6 Extension to Single-Concentration Producibility

As mentioned in Section 1.3.2, we further extend our perfect-mixing algorithm to construct minimum-waste mixing graphs for MixProducibility, when configurations have considerably-many droplets of equal concentration. Naturally, the resulting algorithm is very convenient for applications that require large volumes of equal-concentration droplets.

For example, in drug manufacturing, where mass production plays an essential role, large amounts of droplets with same concentrations are required in early stages of the production. In such application, waste minimization is a crucial constituent of the entire manufacturing process, thus highlighting the importance of this contribution.

Let T be the target configuration. The construction of a mixing graph for T, with waste at most 1 (which is necessary when the reactant volume in T is non-integral), follows directly from Theorem 44 below.

Theorem 44 *Let $n = |T|$ and $d = \mathsf{prec}(T \cup \{\mu\})$, for $\mu = \mathsf{ave}(T)$. If $n > d2^d > 7$ then T is producible with at most 1 droplet of waste.*

Proof. Let $b = (1-\mu)n$ and $r = n\mu$ be the buffer and reactant volumes in T, respectively. As $n > d2^d$, we have that $b, r > d$. If both b and r are integral, we simply use our perfect-mixing algorithm to produce T without any waste; the initial set is $I = \{b : 0, r : 1\}$, which satisfies Condition (MC). (I satisfies Condition (MC) because, when mapped to integers, each concentration in I is multiplied by 2^d, so every 1 becomes 2^d, and $0 \not\equiv 2^d \mod x$ for all positive odd x.)

Otherwise, let $I' = \{\lceil b \rceil : 0, \lceil r \rceil : 1\}$ with $|I'| = n + 1$. Use Algorithm `Min-Mix` (see Section 1.2.1.1 in Chapter 1) on I' to produce concentration $w = \lceil r \rceil - r$, which represents the waste droplet. Let $I'' = I' - \{w\}$, so that $|I''| = n$ and $\mathsf{ave}(I'') = \mu$. Since Algorithm `Min-Mix` uses at most d droplets of either buffer or reactant to produce w, we have that $0, 1 \in I''$, because $\lceil b \rceil, \lceil r \rceil > d$. This implies (by the same argument as for I above) that I'' satisfies Condition (MC), and, as$|I''| \geq 7$, our perfect-mixing algorithm can perfectly mix I'', producing T. ■

Chapter 4

Other Contributions

In this chapter we present the following contributions. First, an $\mathbb{NP}$-hardness proof for a variant of MixReachability, where a mixing graph of fixed-depth is sought, is given in Section 4.1.1. Then, a necessary condition for reachable sets is presented in Section 4.2. Finally, for special configurations, we give decidability proofs in Section 4.3.

4.1 $\mathbb{NP}$-Hardness of a Restricted Variant

In this section we show that the variant of MixReachability where a mixing graph of fixed-depth is sought is $\mathbb{NP}$-hard. We first give complete proofs for mixing graphs of depth at most one and two, respectively. Then we give a construction of an $\mathbb{NP}$-hardness proof for mixing graphs of fixed depth.

4.1.1 ℕℙ-Hardness for Mixing Graphs of Depth At Most 1

Let the Depth-1-MixReachability problem be the following: Given two configurations I and T, determine whether T is reachable from I via a mixing graph of depth at most one. (The depth is defined as the maximum number of nodes on a path from an input to an output. So a mixing graph of depth 1 does not have any edges between mixers — each mixer is connected to two input nodes and two output nodes.)

Recall that Numerical-3D-Matching is defined as follows: Given three multisets X, Y, Z of non-negative integers such that $|X| = |Y| = |Z| = m$, and a non-negative integer S, determine whether (X, Y, Z) has a 3D-matching consisting of triplets each adding up to S. (A 3D-matching of (X, Y, Z) is defined as a partition M of $X \cup Y \cup Z$ into m triplets of the form $(x, y, z) \in X \times Y \times Z$.) Numerical-3D-Matching is well-known to be ℕℙ-complete.

In Theorem 45 below, we show that the Depth-1-MixReachability problem is ℕℙ-hard by giving a polynomial-time reduction from Numerical-3D-Matching.

Theorem 45 *The* Depth-1-MixReachability *problem is* ℕℙ*-hard.*

Proof. We prove the theorem by giving a polynomial-time reduction from Numerical-3D-Matching. Let $X = \{x_i\}_i$, $Y = \{y_i\}_i$, and $Z = \{z_i\}_i$ be the sets from an instance of Numerical-3D-Matching, as defined above. We construct two configurations I and T as follows. For each $i = 1, 2, \ldots, m$:

- Add one droplet with concentration $a_i = 2x_i + \frac{1}{2}$ and one droplet with concentration $b_i = 2y_i + 1$ to I.
- Add two droplets with concentration $c_i = S - z_i + \frac{3}{4}$ to T.

We claim that there exists a 3D-Matching M of (X, Y, Z) consisting of triplets that add up to S if and only if T is reachable from I via a mixing graph of depth at most one.

($\Longrightarrow$) Assume M is a 3D-Matching of (X, Y, Z) where each triplet $(x_i, y_j, z_k) \in M$ adds up to S. For each $(x_i, y_j, z_k) \in M$, we have

$$\tfrac{1}{2}(a_i + b_j) \;=\; \tfrac{1}{2}[\,(2x_i + \tfrac{1}{2}) + (2y_j + 1)\,] \;=\; x_i + y_j + \tfrac{3}{4} \;=\; S - z_k + \tfrac{3}{4} \;=\; c_k.$$

Create a mixing graph G where for each $(x_i, y_j, z_k) \in M$ we create a mixer node with inputs a_i and b_j and two outputs c_k. Then G converts I into T.

($\Longleftarrow$) Assume that there is a mixing graph G of depth at most 1 that converts I into T. All numbers in I are either half-integral a_i's or integral b_j's, and all numbers c_k in T have fractional part $\frac{3}{4}$. So G must consist of m mixer nodes, where each node has incoming edges from some $a_i \in I$ and some $b_j \in I$ and outgoing edges to two c_k's in T, where $c_k = \frac{1}{2}(a_i + b_j)$. Create a 3D-Matching M as follows: for each such node include the corresponding triplet (x_i, y_j, z_k) in M. By simple calculation (reversing the calculation in implication ($\Longrightarrow$)), we get that $x_i + y_j + z_k = S$. Thus M is indeed a correct solution to the instance of Numerical-3D-Matching. ■

4.1.2 $\mathbb{NP}$-Hardness for Mixing Graphs of Depth At Most 2

Let the Depth-2-MixReachability problem be the following: Given two configurations I and T, determine whether T is reachable from I via a mixing graph of depth at most two.

In Theorem 46 below, we show that the Depth-2-MixReachability problem is $\mathbb{NP}$-hard by giving a polynomial-time reduction from Numerical-3D-Matching.

Theorem 46 *The* Depth-2-MixReachability *problem is* $\mathbb{NP}$*-hard.*

Proof. We prove the theorem by giving a polynomial-time reduction from Numerical-3D-Matching. Let $X = \{x_i\}_i$, $Y = \{y_i\}_i$, and $Z = \{z_i\}_i$ be the sets from an instance of Numerical-3D-Matching. We construct two configurations I and T as follows: Add $2m$ droplets with concentration 0 to I, where $m = |X|$. Then, for each $i = 1, 2, \ldots, m$:

- Add one droplet with concentration $a_i = 4x_i + \frac{1}{4}$ and one droplet with concentration $b_i = 4y_i + 1$ to I.
- Add four droplets with concentration $c_i = S - z_i + \frac{5}{16}$ to T.

We claim that there exists a 3D-Matching M of (X, Y, Z) consisting of triplets that add up to S if and only if T is reachable from I via a mixing graph of depth at most two.

($\Longrightarrow$) Assume M is a 3D-Matching of (X, Y, Z) where each triplet $(x_i, y_j, z_k) \in M$ adds up to S. For each $(x_i, y_j, z_k) \in M$, we have

$$\tfrac{1}{4}(a_i + b_j) \;=\; \tfrac{1}{4}[\,(4x_i + \tfrac{1}{4}) + (4y_j + 1)\,] \;=\; x_i + y_j + \tfrac{5}{16} \;=\; S - z_k + \tfrac{5}{16} \;=\; c_k.$$

Create a mixing graph G of depth two as follows: Let Q_1 and Q_2 represent the set of mixers in G at depth one and two, respectively. For each triplet $(x_i, y_j, z_k) \in M$, add a mixer to Q_1 with inputs (a_i, b_j) so that its outputs equal $\frac{1}{2}(a_i + b_j)$. Also, add two mixers to Q_2 with inputs $(\frac{a_i+b_j}{2}, 0)$ each, so that the four outputs equal $\frac{a_i+b_j}{4}$. Then G converts I into T.

$(\Longleftarrow)$ Assume that there is a mixing graph G of depth at most 2 that converts I into T. Note that all a_i's in I have fractional part $\frac{1}{4}$, while all other numbers are integral b_j's and 0's. Further, all numbers c_k in T have fractional part $\frac{5}{16}$. With this in mind, consider the fractional part of every possible concentration produced by a mixer node at depth one:

$$\begin{array}{rlcrl}
a_i & \mod 2 = \frac{1}{8} & & b_j & \mod 2 = \frac{1}{2} \\
2a_i & \mod 2 = \frac{1}{4} & & 2b_j & \mod 2 = 0 \\
(a_i + b_j) & \mod 2 = \frac{5}{8} & & 0 & \mod 2 = 0
\end{array}$$

Since no fractional part equals $\frac{5}{16}$, then G must be a graph of depth two. Thus, consider the fractional part of every possible concentration produced by a mixer node at depth two:

$$\begin{array}{rlcrl}
a_i & \mod 4 = \frac{1}{16} & & b_j & \mod 4 = \frac{4}{16} \\
2a_i & \mod 4 = \frac{2}{16} & & 2b_j & \mod 4 = \frac{8}{16} \\
3a_i & \mod 4 = \frac{3}{16} & & 3b_j & \mod 4 = \frac{12}{16} \\
4a_i & \mod 4 = \frac{4}{16} & & 4b_j & \mod 4 = 1 \\
(a_i + b_j) & \mod 4 = \frac{5}{16} & \checkmark & (2a_i + 2b_j) & \mod 4 = \frac{10}{16} \\
(2a_i + b_j) & \mod 4 = \frac{6}{16} & & (a_i + 2b_j) & \mod 4 = \frac{9}{16} \\
(3a_i + b_j) & \mod 4 = \frac{7}{16} & & (a_i + 3b_j) & \mod 4 = \frac{13}{16} \\
0 & \mod 4 = 0 & & &
\end{array}$$

Hence, at depth two, the only possible concentration that has fractional part $\frac{5}{16}$ is $\frac{1}{4}(a_i+b_j)$. Therefore, the output of every mixer node at depth two must equal $\frac{1}{4}(a_i + b_j)$. To this end, and from the analysis above for mixer nodes at depth one, the only possible input pairs for mixer nodes at depth two are either $(\frac{a_i+b_j}{2}, 0)$ or $(\frac{a_i}{2}, \frac{b_j}{2})$. Consequently, the input pairs for mixer nodes at depth one are either $(0, 0)$, $(0, a_i)$, $(0, b_j)$ or (a_i, b_j).

Create a 3D-Matching M as follows. For each mixer m in G at depth two: If m's input pair equals $(\frac{a_i}{2}, \frac{b_j}{2})$, then add the corresponding triplet (x_i, y_j, z_k) to M. Otherwise, m's input pair equals $(\frac{a_i+b_j}{2}, 0)$. Then, a_i and b_j can be obtained by identifying the corresponding mixer, whose output pair equals $\frac{a_i+b_j}{2}$, at depth one; the input pair for this mixer node is (a_i, b_j). In either case, a_i and b_j are chosen so that $c_k = \frac{1}{4}(a_i + b_j)$. Finally, add the corresponding triplet (x_i, y_j, z_k) to M.

Every triplet $(x_i, y_j, z_k) \in M$ adds up to S:

$$x_i + y_j + z_k \;=\; \tfrac{1}{16}(4a_i - 1) + \tfrac{1}{16}(4b_j - 4) + (S - c_k + \tfrac{5}{16}) \;=\; \tfrac{1}{4}(a_i + b_j) + S - c_k \;=\; S$$

Thus M is indeed a correct solution to the instance of NUMERICAL-3D-MATCHING. ■

4.1.3 $\mathbb{NP}$-Hardness for Mixing Graphs of Fixed-Depth

Let the FIXED-DEPTH-MIXREACHABILITY problem be the following: given two configurations I and T, and a non-negative integer l, determine whether T is reachable from I via a mixing graph of depth l.

In this section, we show that the FIXED-DEPTH-MIXREACHABILITY problem is $\mathbb{NP}$-hard (see Corollary 47 below) by giving an extension to the proofs in Theorems 45 and 46.

Let $X = \{x_i\}_i$, $Y = \{y_i\}_i$, and $Z = \{z_i\}_i$ be the sets from an instance of NUMERICAL-3D-MATCHING. We construct two configurations I and T as follows: Add $m\sum_{i=1}^{l-1} 2^i$ droplets with concentration 0 to I, where $m = |X|$. Then, for each $i = 1, 2, \ldots, m$:

- Add one droplet with concentration $a_i = 2^l \cdot x_i + \frac{1}{2^l}$ and one droplet with concentration $b_i = 2^l \cdot y_i + 1$ to I.
- Add 2^l droplets with concentration $c_i = S - z_k + \frac{2^l+1}{2^{2l}}$ to T.

Corollary 47 *The* FIXED-DEPTH-MIXREACHABILITY *problem is* $\mathbb{NP}$*-hard.*

4.2 A Necessary Condition for Mixing Reachability

In this section, we present a necessary condition for reachable sets. We refer to this condition as *dominance relation.* Additionally, we show that, if we are allowed to mix arbitrary volumes of fluids, this dominance relation is also a sufficient condition.

Let C be an arbitrary configuration with $|C| = n$. We define function $\varrho()$ as follows: $\varrho(C)$ denotes an enumeration of C for which the i^{th} entry $\varrho(C)_i$ refers to the i^{th} smaller concentration in C. $\varrho(C,k) = \sum_{i=1}^{k} \varrho(C)_i$ denotes the sum of the k smaller concentrations in C. For example, for $C = \{(2 : \frac{6}{8}), (2 : \frac{1}{8}), \frac{3}{8}\}$, $\varrho(C)_3 = \frac{3}{8}$ and $\varrho(C,3) = \frac{5}{8}$.

Given two configurations I and T with $|I| = |T|$, we say that T *dominates* I if $\varrho(T,k) \geq \varrho(I,k)$ for all k. We show that dominance relation is a necessary condition for MIXREACHABILITY. We also show that if we are allowed to mix two droplets in any volume proportion, then this dominance relation becomes also a sufficient condition.

4.2.1 A Necessary Condition

Let I and T be the initial and target configurations, respectively, with $|I| = |T| = n$. In this section, we show that the dominance relation is a necessary condition for MixReachability. In other words, we show that if T is reachable from I then T dominates I.

To this end, assume that T is reachable from I and let G be a mixing sequence that converts I into T. By Lemma 48 below, after a mixing operation on I, the resulting configuration dominates I. Therefore, since T can be obtained by applying the mixing sequence G to I, and as the dominance relation is transitive, we have that T dominates I.

Lemma 48 *Let A be an arbitrary configuration. Also, let A' be obtained after a mixing operation on A. Then A' dominates A.*

Proof. Without loss of generality, assume that the concentrations of the mixed droplets in A are $\varrho(A)_i$ and $\varrho(A)_j$, with $i < j$. (Otherwise, $i = j$ and thus $A = A'$, so A' trivially dominates A.) Let $\sigma = \frac{1}{2}(\varrho(A)_j - \varrho(A)_i)$. Mixing $\varrho(A)_i$ with $\varrho(A)_j$ is analogous to adding and subtracting σ to $\varrho(A)_i$ and $\varrho(A)_j$, respectively. To this end, we consider the following process:

1. If $\varrho(A)_i = \varrho(A)_{i+1}$, we can trivially increase i by one without affecting the mixing process. Similarly, if $\varrho(A)_j = \varrho(A)_{j-1}$, we can decrease j by one as well.
2. Otherwise, let $\phi = \min(\, \varrho(A)_{i+1} - \varrho(A)_i,\ \varrho(A)_j - \varrho(A)_{j-1},\ \frac{1}{2}(\varrho(A)_j - \varrho(A)_i)\,)$ and $\hat{A} = A$. Increase $\varrho(\hat{A})_i$ and decrease $\varrho(\hat{A})_j$ by ϕ, respectively. Since $\phi > 0$, we have that $\varrho(\hat{A}, k) > \varrho(A, k)$ for $k = i, i+1, \ldots, j-1$. (For all other indices $l \neq k$, $\varrho(\hat{A}, l) = \varrho(A, l)$.) Thus, $\hat{A}$ dominates A.
3. Repeat the process for $\hat{A}$, $\hat{\imath} = i + 1$ and $\hat{\jmath} = j - 1$, until $\varrho(\hat{A})_{\hat{\imath}} = \varrho(\hat{A})_{\hat{\jmath}}$.

After each iteration, the resulting configuration dominates its previous configuration, so the lemma holds. ■

4.2.2 A Sufficient Condition for a Relaxed Variant

Assume that we are allowed to mix two droplets in any volume proportion. Under this condition, in Theorem 49 below, we show that the dominance relation is a sufficient condition for MixReachability.

Let A be an arbitrary configuration. We define a *fractional mixing* between two droplets $x, y \in A$, as two mixing operations where, for some real number a with $0 < a \leq 1$, in the first mixing operation, a volume of droplet x is mixed with $(1-a)$ volume of droplet y, and in the second mixing operation, $(1-a)$ volume of droplet x is mixed with a volume of droplet y. Thus, for some real number a with $0 < a \leq 1$, a fractional mixing between x and y produces two droplets with concentrations $ax + (1-a)y$ and $(1-a)x + ay$, respectively.

Theorem 49 *Let I and T be the initial and target configurations, respectively, with $|I| = |T|$. Then, T is reachable from I by a sequence of fractional mixes if and only if T dominates I.*

Proof. ($\Longrightarrow$) Assume that T can be obtained from I by a sequence of fractional mixes. That T dominates I follows from Lemma 50 below.

($\Longleftarrow$) Assume that T dominates I. Let $d = \mathsf{prec}(I \cup T)$. By Lemma 51 below, we can implement the process of a droplet giving a unit of concentration $\frac{1}{2^d}$ to another droplet by using a fractional mixing, with corresponding a. Also, by Lemma 52 (below Lemma 51), we know that whenever $\varrho(T, i) > \varrho(I, i)$, for the smallest i, there exists $j > i$ such that $\varrho(I)_j > \varrho(I)_i$, and thus $\varrho(I)_j$ can give a unit of concentration $\frac{1}{2^d}$ to $\varrho(I)_i$. We can thus repeat this process until $I = T$.

The above process finishes after at most $\Psi(T) - \Psi(I)$ droplets have given a unit of concentration $\frac{1}{2^d}$ to other droplets, because after each step, $\Psi(I)$ increases by at least one. (Recall that, for some configuration C, $\Psi(C)$ denotes the variance of the concentrations in C.) ■

Lemma 50 *Let A be an arbitrary configuration and A' be obtained from A after a fractional mixing. Then, A' dominates A.*

Proof. Let a (with $0 < a \leq 1$) be the corresponding real number associated to the fractional mixing used to produce A' from A. Also, let $\varrho(A)_i$ and $\varrho(A)_j$ be the concentrations of the droplets used to produce A', such that a volume of $\varrho(A)_i$ is mixed with $(1-a)$ volume of $\varrho(A)_j$ and $(1-a)$ volume of $\varrho(A)_i$ is mixed with a volume of $\varrho(A)_j$.

Assume without loss of generality that $i < j$ such that $\varrho(A)_i < \varrho(A)_j$. (For $i = j$, we have that $A = A'$ and the lemma trivially holds.) Mixing a volume of $\varrho(A)_i$ with $(1-a)$ volume of $\varrho(A)_j$ is analogous to increasing the concentration of the a-volumed fluid by $\alpha = \lambda - \varrho(A)_i$ and decreasing the concentration of the $(1-a)$-volumed fluid by $\beta = \varrho(A)_j - \lambda$, for $\lambda = a \cdot \varrho(A)_i + (1-a) \cdot \varrho(A)_j$. (Similarly for mixing $(1-a)$ volume of $\varrho(A)_i$ with a volume of $\varrho(A)_j$.)

To prove the lemma under the above analogy, it is convenient to think of the droplets in A as ψ-sized mini-droplets, where $\psi = \mathsf{gcd}(a, 1-a)$. To this end, let $\hat{A}$ denote a configuration containing the corresponding $\psi \cdot |A|$ mini-droplets of A. Also, let $\varrho(\hat{A})_{\hat{i}}$ to $\varrho(\hat{A})_{\hat{i}'}$ and $\varrho(\hat{A})_{\hat{j}}$ to $\varrho(\hat{A})_{\hat{j}'}$ be the a-volumed and the $(1-a)$-volumed mini-droplets, respectively, of $\varrho(A)_i$ and $\varrho(A)_j$, correspondingly, involved in the first mixing operation. Consider the following process:

1. If $\varrho(\hat{A})_{\hat{\imath}'} = \varrho(\hat{A})_{\hat{\imath}'+1}$, we can trivially increase $\hat{\imath}$ and $\hat{\imath}'$ by one without affecting the mixing operation. Similarly, if $\varrho(\hat{A})_{\hat{\jmath}} = \varrho(\hat{A})_{\hat{\jmath}-1}$, we can also decrease both $\hat{\jmath}$ and $\hat{\jmath}'$ by one as well.

2. Otherwise, let $\hat{A}' = \hat{A}$. Simultaneously increase $\varrho(\hat{A}')_{\hat{\imath}}$ through $\varrho(\hat{A}')_{\hat{\imath}'}$ and decrease $\varrho(\hat{A}')_{\hat{\jmath}}$ through $\varrho(\hat{A}')_{\hat{\jmath}'}$ by $\sigma, \phi > 0$ respectively, at a rate proportionally to α and β correspondingly, until either $\varrho(\hat{A}')_{\hat{\imath}'} + \sigma = \varrho(\hat{A}')_{\hat{\imath}'+1}$, $\varrho(\hat{A}')_{\hat{\jmath}} - \phi = \varrho(\hat{A}')_{\hat{\jmath}-1}$ or $\varrho(\hat{A}')_{\hat{\imath}'} + \sigma = \varrho(\hat{A}')_{\hat{\jmath}} - \phi$ holds.

   Either way, as $\sigma, \phi > 0$, we have that $\varrho(\hat{A}', k) > \varrho(\hat{A}, k)$ for $k = \hat{\imath}, \hat{\imath}+1, \ldots, \hat{\jmath}-1$. (For all other indices $l \neq k$, $\varrho(\hat{A}', l) = \varrho(\hat{A}, l)$.) Thus, $\hat{A}'$ dominates $\hat{A}$.

3. Repeat the process for $\hat{A}'$, with corresponding $\hat{\imath}$, $\hat{\imath}'$, $\hat{\jmath}$ and $\hat{\jmath}'$, until $\varrho(\hat{A}')_{\hat{\imath}} = \varrho(\hat{A}')_{\hat{\imath}+1} = \varrho(\hat{A}')_{\hat{\jmath}'} = \lambda$.

After each iteration, the resulting configuration $\hat{A}'$ dominates its corresponding previous configuration $\hat{A}$. This also holds for mixing the remaining $(1-a)$-volumed mini-droplets of $\varrho(A)_i$ with the remaining a-volumed mini-droplets of $\varrho(A)_j$. (Note that the volume of fluid manipulated by the fractional mixing adds up to two droplets, exactly: one droplet of volume per mixing operation.) Therefore, A' must dominate A. ■

Lemma 51 *Let $d = \textsf{prec}(I \cup T)$ and $1 \leq i < j \leq n$. Also, let $\varrho(I)_i = \frac{x}{2^d}$ and $\varrho(I)_j = \frac{y}{2^d}$, where x and y are integers satisfying $x < y$. Applying a fractional mixing to $\varrho(I)_i$ and $\varrho(I)_j$, with corresponding $a = \frac{y-x-1}{y-x}$, produces two droplets with concentrations $\varrho(I)_i + \frac{1}{2^d}$ and $\varrho(I)_j - \frac{1}{2^d}$, respectively.*

Proof.

$$\begin{aligned}\frac{y-x-1}{y-x}\cdot\varrho(I)_i+\frac{1}{y-x}\cdot\varrho(I)_j &= \frac{1}{y-x}\Big[\frac{x(y-x-1)}{2^d}+\frac{y}{2^d}\Big]\\ &= \frac{1}{2^d(y-x)}\big[x(y-x)+(y-x)\big]\\ &= \varrho(I)_i+\frac{1}{2^d}\end{aligned}$$

(The proof for $\frac{y-x-1}{y-x}\cdot\varrho(I)_j-\frac{1}{y-x}\cdot\varrho(I)_i=\varrho(I)_j-\frac{1}{2^d}$ follows from above by switching i with j and by subtracting instead of adding.) ■

Lemma 52 *Assume that T dominates I. Also, assume that $\varrho(T,i)>\varrho(I,i)$ for some $1\le i\le n$. Let ι be the smallest i that satisfies $\varrho(T,\iota)>\varrho(I,\iota)$. Then, there exists $j>\iota$ that satisfies $\varrho(I)_j>\varrho(I)_\iota$.*

Proof. We know that $\varrho(T)_\iota>\varrho(I)_\iota$ because ι is the smallest integer that satisfies $\varrho(T,\iota)>\varrho(I,\iota)$. Also, since T dominates I, there exists some j with $\iota<j\le n$ that satisfies $\varrho(I)_j>\varrho(T)_j$. Hence $\varrho(I)_\iota<\varrho(T)_\iota\le\varrho(T)_j<\varrho(I)_j$, and thus the lemma holds. ■

4.3 Decidability of Special Cases

As mentioned in Chapter 1, it is not known whether MixReachability and MixProducibility are decidable. Nonetheless, in this section, for MixProducibility, we give decidability proofs for special types of configurations. In particular, we give decidability proofs for configurations with at most four droplets, configurations with precision 2 and configurations containing only pcvs. We also give a sufficient condition for producing configurations containing pcvs and one non-pcv. (Recall that a pcv denotes a concentration value with exactly one bit 1 in its binary representation.)

In the following sections, for an arbitrary configuration C, we define by $\mathsf{bv}(C)$ and $\mathsf{rv}(C)$ the buffer and reactant volumes in C, respectively.

4.3.1 Configurations with At Most 4 Droplets

In this section, we show that MixProducibility is decidable for configurations with at most four droplets. Let T with $|T| = n \leq 4$ be the target configuration. If the reactant (or buffer) volume in T is not integral, then T is clearly not producible. So, assume that T's reactant volume (and thus the buffer volume) is integral and let $I = \{a : 0, b : 1\}$ be the initial configuration, where $a = \sum_{c \in T}(1 - c)$ and $b = \sum_{c \in T} c$ are integers; either $a = 0$ or $b = 0$ implies that $I = T$, so further assume that $a, b \geq 1$.

The decidability proof for the case $n \leq 2$ is trivial, since at most one mixing operation is required to determine whether T producible from I. The case $n = 3$ is also easy. Mixing 0 and 1 produces $I' = \left\{1 : c, 2 : \frac{1}{2}\right\}$, for $c \in \{0, 1\}$, with precision $\mathsf{prec}(I') = 1 > \mathsf{prec}(I) = 0$, so the precision increases. In particular, after every mixing operation, the resulting configuration has form $\{1 : c_1, 2 : c_2\}$, with $\mathsf{prec}(c_1) < \mathsf{prec}(c_2)$. Therefore, by Observation 53 below, every mixing operation increases the precision. (Note that we never mix two equal concentrations, since the result is redundant.) Hence, T is producible from I only if after $\mathsf{prec}(T)$ mixing operations, I becomes T.

Observation 53 *Let c_1 and c_2 be two distinct concentrations. If $\mathsf{prec}(c_1) \neq \mathsf{prec}(c_2)$, then $\mathsf{prec}(\frac{1}{2}(c_1 + c_2)) = \max(\mathsf{prec}(c_1), \mathsf{prec}(c_2)) + 1$.*

Proof. Assume (by symmetry) that $\mathsf{prec}(c_1) < \mathsf{prec}(c_2)$. If we represent both c_1 and c_2 by the fractions $c_1 = \frac{x}{2^{\mathsf{prec}(c_2)}}$ and $b = \frac{y}{2^{\mathsf{prec}(c_2)}}$, we get that x is even and y is odd. Therefore, mixing c_1 and c_2 produces

$$c_3 = \frac{c_1 + c_2}{2} = \frac{x + y}{2^{\mathsf{prec}(c_2)+1}}$$

and since $x + y$ is odd, $\mathsf{prec}(c_3) = \mathsf{prec}(c_2) + 1$. ■

It remains to show that MixProducibility is decidable for $n = 4$. The idea of the proof is somewhat similar to the case $n = 3$. We show that after every three mixing operations, either the current configuration becomes perfectly mixed (as in PerfectMixability) or its precision increases. Thus, if the precision of the current configuration exceeds that of the target configuration, then we can stop mixing. This implies that we only need to consider mixing sequences of length at most $3 \cdot \mathsf{prec}(T)$, so there are finitely many mixing sequences to check. The proof is presented in Lemma 54 below.

Lemma 54 *If $n = 4$, then it is decidable to determine whether T is producible.*

Proof. As I consists of only 0's and 1's, the first mixing operation trivially increases the precision. Furthermore, every time a mixing operation increases the precision, a configuration I' of form $I' = \{c_1, c_2, c_3, c_3\}$ (c_1 and c_2 might be equal) is obtained, where c_3 denotes the concentration produced by the mixing operation and where $\mathsf{prec}(c_1, c_2) < \mathsf{prec}(c_3)$.

We next show that after at most three mixing operations, the current configuration either becomes perfectly mixed or its precision increases. Without loss of generality, assume that the initial configuration is $I = \{c_1, c_2, c_3, c_3\}$ with $\mathsf{prec}(c_1, c_2) < \mathsf{prec}(c_3)$. (Otherwise, as mentioned above, I consists of 0's and 1's, so the next mixing operation mixes a 0 and a 1, increasing the precision. Thus we get a configuration I' of this form.) By Observation 53, and as $\mathsf{prec}(c_1, c_2) < \mathsf{prec}(c_3)$, mixing c_3 with either c_1 or c_2 increases the precision. So, instead, assume that c_3 is not mixed at the first step.

Thus, the first mixing operation mixes c_1 and c_2, producing configuration $I'' = \{c_3, c_3, c_4, c_4\}$, where $c_4 = \frac{1}{2}(c_1 + c_2)$ and $\mathsf{prec}(c_4) \leq \mathsf{prec}(c_3)$. (Note that $\mathsf{prec}(c_4) \leq \mathsf{prec}(c_3)$ holds because c_4 is obtained by diving $c_1 + c_2$ by 2, so $\mathsf{prec}(c_4)$ can be at most $\max(\mathsf{prec}(c_1), \mathsf{prec}(c_2)) + 1 \leq \mathsf{prec}(c_3)$.) If $\mathsf{prec}(c_4) < \mathsf{prec}(c_3)$, then mixing c_3 and c_4 increases the precision and we are done. Otherwise, $\mathsf{prec}(c_4) = \mathsf{prec}(c_3)$ and, by Observation 55 below, mixing c_3 and c_4 produces a configuration $I'' = \{c_3, c_4, c_5, c_5\}$ with $\mathsf{prec}(c_3) \neq \mathsf{prec}(c_5)$, where $c_5 = \frac{1}{2}(c_3 + c_4)$. Therefore, if the third (and last) mixing operation involves c_3 and c_4, then I'' becomes perfectly mixed. Otherwise, c_5 is mixed with either c_3 or c_4, and thus the precision increases.

Finally, consider all mixing sequences of length at most $3 \cdot \mathsf{prec}(T)$. If any of those mixing sequences produces T, then T is producible. Otherwise, T is not producible. ■

Observation 55 *Let $C = \{2 : c_1, 2 : c_2\}$ with $c_1 \neq c_2$ and $\mathsf{prec}(c_1) = \mathsf{prec}(c_2)$ be a configuration with integral reactant volume. Then $\mathsf{prec}(\frac{1}{2}(c_1 + c_2)) \neq \mathsf{prec}(c_1)$.*

Proof. If we represent both c_1 and c_2 by the fractions $c_1 = \frac{x}{2^{\mathsf{prec}(c_1)}}$ and $c_2 = \frac{y}{2^{\mathsf{prec}(c_1)}}$, we get that both x and y are odd integers. Let $2\sigma = x + y$. Then

$$\frac{c_1 + c_2}{2} = \frac{x + y}{2^{\mathsf{prec}(c_1)+1}} = \frac{\sigma}{2^{\mathsf{prec}(c_1)}}.$$

Assume by contradiction that $\mathsf{prec}(\frac{1}{2}(c_1 + c_2)) = \mathsf{prec}(c_1)$; that is, assume that σ is odd. Since the reactant volume of C is integral, we have that

$$\frac{4\sigma}{2^{\mathsf{prec}(c_1)}} = \frac{\sigma}{2^{\mathsf{prec}(c_1)-2}}$$

is integral, which only holds for $\mathsf{prec}(c_1) = 2$. (Note that $\mathsf{prec}(c_1) \geq 0$.) Now, as $c_1 \neq c_2$, the only possible values for c_1 and c_2 that satisfy the corresponding constraints are $c_1 = \frac{1}{4}$ and $c_2 = \frac{3}{4}$. However, mixing c_1 and c_2 produces $c_3 = \frac{c_1+c_2}{2} = \frac{1}{2}$, with $\mathsf{prec}(c_3) < \mathsf{prec}(c_1)$, and thus a contradiction. ■

4.3.2 Configurations with Precision 2

In this section, we give a characterization for configurations with precision 2. This characterization is presented in Theorem 56 below. Let $T = \{f_1 : \frac{1}{4}, f_2 : \frac{1}{2}, f_3 : \frac{3}{4}\}$ be the target configuration, where f_1, f_2 and f_3 are non-negative integers.

Theorem 56 *T is producible if and only if (4.1) and (4.2) below hold.*

$$f_1 + 2f_2 + 3f_3 \textit{ is divisible by } 4 \tag{4.1}$$

$$T \neq \{\frac{1}{4}, \frac{3}{4}\} \tag{4.2}$$

Proof. ($\Longrightarrow$) Assume that T is producible. Let

$$\mathsf{rv}(T) = f_1 \cdot \frac{1}{4} + f_2 \cdot \frac{1}{2} + f_3 \cdot \frac{3}{4} = \frac{f_1 + 2f_2 + 3f_3}{4}. \tag{4.3}$$

By definition of MixProducibility, $\mathsf{rv}(T)$ is integral. This implies that (4.1) holds by (4.3). Also, at the beginning of any mixing sequence, we always mix a 0 with a 1, thus producing concentration $\frac{1}{2}$. After this, we will always need to mix an extra 0 to produce concentration $\frac{1}{4}$ or an extra 1 to produce concentration $\frac{3}{4}$, implying $|T| \geq 3$ whenever $f_1 + f_3 \geq 1$. Therefore, (4.2) also holds.

($\Longleftarrow$) Assume that (4.1) and (4.2) hold. By (4.1), we know that f_1 and f_3 are either both even or both odd, so we have two cases:

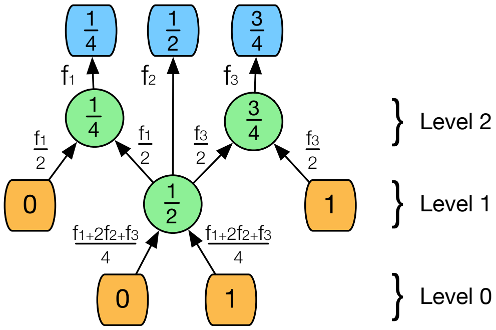

Figure 4.1: Characterization for a configuration with f_1 and f_3 even.

Case 1: Both f_1 and f_3 are even.

The mixing graph in Figure 4.1 shows a characterization for this case. We know that $\frac{f_1}{2}$ and $\frac{f_3}{2}$ are both integral because f_1 and f_3 are even. Also, $\frac{f_1+2f_2+f_3}{4}$ is integral because $\frac{f_1+2f_2+f_3}{4} = \frac{f_1+2f_2+3f_3}{4} - \frac{f_3}{2}$, where both $\frac{f_1+2f_2+3f_3}{4}$ and $\frac{f_3}{2}$ are integral.

Case 2: Both f_1 and f_3 are odd.

We consider the following sub-cases:

Case 2.1: $f_1 = f_3$.

This implies that $f_1 \cdot \frac{1}{4} + f_1 \cdot \frac{3}{4} = f_1$, so, by (4.1), $f_2 \cdot \frac{1}{2}$ must be integral, which implies f_2 is even. Consider the following sub-cases:

Case 2.1.1: $f_1, f_3 = 1$.

The left mixing graph in Figure 4.2 illustrates this characterization. As $f_1, f_3 = 1$, both nodes $\frac{1}{4}$ and $\frac{3}{4}$ at depth 2 have to get rid of one droplet each (since a mixer produces two droplets), and such droplets will eventually become two droplets with concentration $\frac{1}{2}$. This implies $f_2 \geq 2$, so in the mixing graph, $f_2 - 2 \geq 0$ and $\frac{f_2}{2}$ is integral.

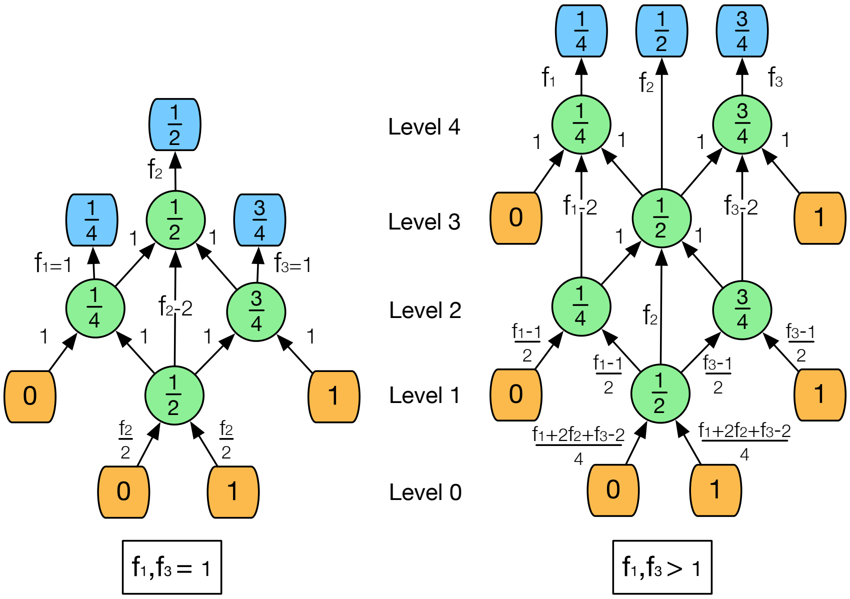

Figure 4.2: Characterization for a configurations with $f1$ and $f3$ odd. The left mixing graph illustrates a characterization for $f1, f3 = 1$. The right mixing graph illustrates a characterization for $f1, f3 > 1$.

Case 2.1.2: $f_1, f_3 > 1$.

The right mixing graph in Figure 4.2 illustrates this characterization. As $f_3 > 1$, then $f_3 - 2 \geq 0$. Additionally, $\frac{f_3-1}{2}$ is integral because f_3 is odd. Finally, $\frac{f_1+2f_2+f_3-2}{4}$ is integral because $\frac{f_1+2f_2+f_3-2}{4} = \frac{f_1+2f_2+3f_3}{4} - \frac{f_3-1}{2}$. The same analysis applies to f_1.

Case 2.2: $f_1 = 1$ and $f_3 > 1$.

As in sub-case 2.1.1 above, node $\frac{1}{4}$ at depth 2 needs to get rid of one droplet. This droplet determines whether f_2 is either even or odd:

Case 2.2.1: f_2 odd.

The left mixing graph in Figure 4.3 illustrates this characterization. In this sub-case, node $\frac{1}{2}$ at depth 3 keeps the output droplet of node $\frac{1}{4}$ at depth 2.

In the mixing graph, $f_3 - 2 \geq 0$ and $\frac{f_3-1}{2}$ is integral because $f_3 > 1$ is odd. Also, as $f_1 = 1$, $\frac{2f_2+f_3-1}{4} = \frac{f_1+2f_2+3f_3}{4} - \frac{f_3+1}{2}$ is integral.

Case 2.2.2: f_2 even.

The right mixing graph in Figure 4.3 illustrates this characterization. In this sub-case, node $\frac{1}{2}$ at depth 3 sends the output droplet of node $\frac{1}{4}$ at depth 2 to node $\frac{3}{4}$ on depth 4. Because of this, and as f_3 is odd, $f_3 \geq 5$. In the mixing graph, $f_3 - 4 \geq 0$ and $\frac{f_3-3}{2}$ is integral because $f_3 \geq 5$ is odd. Finally, $\frac{2f_2+f_3-1}{4}$ is integral as in the left mixing graph.

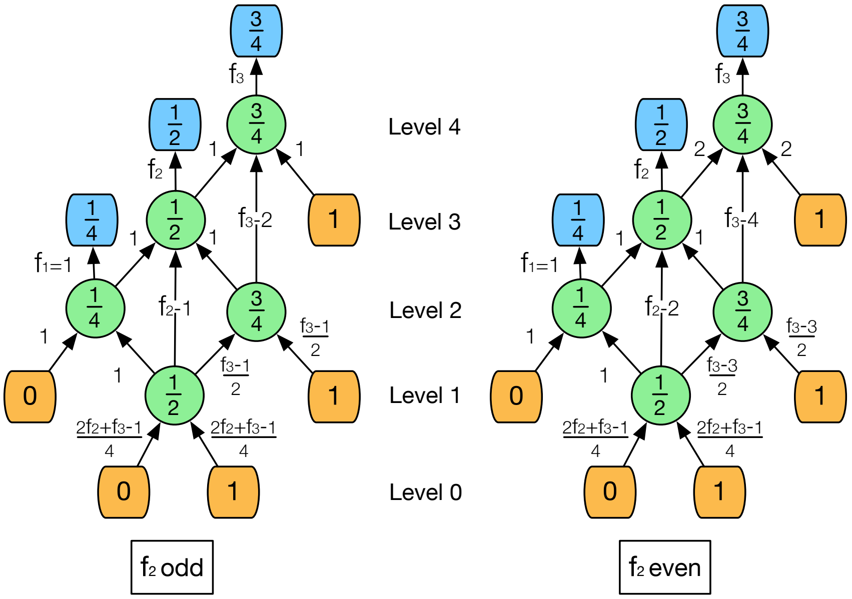

Figure 4.3: Characterization for a configuration with $f_1 = 1$ and $f_3 > 1$ odd. The left mixing graph illustrates the characterization for f_2 odd. The right mixing graph illustrates the characterization for f_2 even.

Case 2.3: $f_1 > 1$ and $f_3 = 1$.

The same analysis as in sub-case 2.2 applies after swapping f_1 with f_3.

■

4.3.3 Configurations with Only PCVs

In this section, we give a characterization for configurations containing only pcvs. Let T be the target configuration with $d = \mathsf{prec}(T)$ and $n = |T|$ non-negative integers. Without loss of generality, assume that for all c in T, $c \leq \frac{1}{2}$. (Otherwise, T might also contain 1's, which basically represent reactant droplets, and thus we can trivially ignore them.)

Theorem 57 *T is producible if and only if $\mathsf{rv}(T)$ is integral.*

Proof. ($\Longrightarrow$) Assume that T is producible. By definition of MixProducibility, the initial set for T contains an integral number 1's, hence $\mathsf{rv}(T)$ is integral.

($\Longleftarrow$) Assume that $\mathsf{rv}(T)$ is integral. This, and n being an integer, imply that $\mathsf{bv}(T)$ is integral. Let $l = 1$ and consider the following mixing process:

1. Let $I = \{(b : 0), (r : \frac{1}{2^{l-1}})\}$ be the current configuration, where $b = \mathsf{bv}(T)$ and $r = \mathsf{rv}(T)$; $b \geq r$ because for all c in T, $c \leq \frac{1}{2^l}$.
2. From I, mix r droplets of concentration 0 with r droplets of concentration $\frac{1}{2^{l-1}}$. Thus, $2r$ droplets of concentration $\frac{1}{2^l}$ are produced.
3. Let x be the number of pcvs with concentration $\frac{1}{2^l}$ in T; $2r \geq x$ because $\mathsf{rv}(I) = \mathsf{rv}(T)$. Output x droplets of concentration $\frac{1}{2^l}$ and let $T' = T - \{x : \frac{1}{2^l}\}$ and $I' = (I \cup \{2r - x : \frac{1}{2^l}\}) - \{r : 0, r : \frac{1}{2^{l-1}}\}$, such that $\mathsf{bv}(I') = \mathsf{bv}(T')$ and $\mathsf{rv}(I') = \mathsf{rv}(T')$.
4. Repeat the process for I' and T' until $T' = \emptyset$.

Given the above process, pcvs in T with concentration $\frac{1}{2^l}$ are produced in iteration l (see step 3). Also, we know that the process stops after d steps, because the smallest pcv in T has concentration $\frac{1}{2^d}$, which is produced at the last iteration d. ■

4.3.4 Configurations with PCVs and One Non-PCV

In this section, in Theorem 58 below, we give a sufficient condition for producible configurations that contain pcvs and one non-pcv. (For simplicity, since 1's are basically reactant droplets, we will assume that all pcvs have concentration at most $\frac{1}{2}$.)

Theorem 58 *Let T be a configuration containing pcvs and one non-pcv, with $d = \mathsf{prec}(T)$ and both $\mathsf{bv}(T)$ and $\mathsf{rv}(T)$ integral. If $\mathsf{rv}(T) \geq 4d$, then T is is producible.*

Proof. Assume that $\mathsf{rv}(T) \geq 4d$ and let $c \in T$ be the non-pcv. By Lemma 59 below, and as $\mathsf{rv}(T)$ is integral, we can always remove from T a subset Q that satisfies $\mathsf{rv}(Q) = 1$ and $c \notin Q$, until there is only $P \subset T$, with $c \in P$ and $\mathsf{rv}(P) = 1$. So, to prove the theorem, we first show how to produce P, and then, using the residual droplets, and some buffer and reactant droplets, we show how to produce $T - P$ (without any waste).

Producing P. We first use Algorithm `Min-Mix` (see Section 1.2.1.1 in Chapter 1) to produce c, which also produces d extra droplets; let C' contain these extra droplets. Then, we produce $P - \{c\}$ (consisting of only pcvs), using exactly one reactant droplet, by using a process similar to the one for configurations with only pcvs (see Section 4.3.3); in this process however we need not to mix all non-0's at each step, but only the necessary ones. This process produces at most d extra droplets (at most one extra droplet per iteration); let P' contain these extra droplets. Thus, P can be produced by also producing P' and C', where $|P'| + |C'| \leq 2d$. (Note that $\mathsf{rv}(C' \cup P')$ is integral because $\mathsf{rv}(\{c\} \cup C')$ and $\mathsf{rv}(P \cup P')$ are integral, and $\mathsf{rv}(\{c\} \cup P) = 1$; in fact, $\mathsf{bv}(C' \cup P')$ is also integral.)

Producing $T - P$. Let $n' = |C' \cup P'| \leq 2d$. If n' is not a power of two, then let $B = \{\beta - n' : 0\}$, where β is the smallest power of two that satisfies $n' < \beta < 2n'$. Otherwise, let $B = \emptyset$.

Also, let $A = \{\mathsf{rv}(B \cup C' \cup P') : 0, \mathsf{bv}(B \cup C' \cup P') : 1\}$. Then, for $I' = \{A \cup B \cup C' \cup P'\}$, we have that $\mathsf{bv}(I') = \mathsf{rv}(I') = \beta$ and thus the average of the concentrations in I' is $\frac{1}{2}$.

Let $T' = \{2\beta : \frac{1}{2}\}$. Then, as $\mathsf{bv}(T') = \mathsf{bv}(I')$ and $\mathsf{rv}(T') = \mathsf{rv}(I')$, produce T' from I' using our perfect-mixing algorithm in Chapter 3. (Note that T' does not need to satisfy Condition (MC), because $|T'|$ power of two implies that T' is near-final.) Clearly, $\mathsf{rv}(T') = \beta < 2n' \leq 4d \leq \mathsf{rv}(T)$. Therefore, as $T - P$ contains only pcvs with concentration at most $\frac{1}{2}$, $T - P$ can be produced from $I = T' \cup \{\mathsf{bv}(T) - \mathsf{bv}(T') : 0, \mathsf{rv}(T) - \mathsf{rv}(T') : 1\}$ by the process described in Section 4.3.3 for configurations with only pcvs. (The first iteration is slightly different though, since we mix $\mathsf{rv}(T) - \mathsf{rv}(T')$ 1's instead of $\mathsf{rv}(I)$.) ■

Lemma 59 *Let A be a configuration with only pcvs and assume that $\mathsf{rv}(A) > 1$. There exists $A' \subset A$ such that $\mathsf{rv}(A') = 1$.*

Proof. Let $A' = \emptyset$ and $n = |A|$. For $i = n, n-1, \ldots, 1$, if $\mathsf{rv}(A') + \varrho(A)_i \leq 1$, add $\varrho(A)_i$ to A'. As $\varrho(A)_{i-1}$ is either equal or half $\varrho(A)_i$, for $1 < i \leq n$, then we have that eventually $\mathsf{rv}(A') = 1$. ■

Bibliography

[1] Ravindra K. Ahuja, James B. Orlin, Giovanni M. Sechi, and Paola Zuddas. Algorithms for the simple equal flow problem. *Management Science*, 45(10):1440–1455, 1999.

[2] Kyle Campbell and Alex Groisman. Generation of complex concentration profiles in microchannels in a logarithmically small number of steps. *Lab on a Chip*, 7(2):264–272, 2007.

[3] Ting W. Chiang, Chia H. Liu, and Juinn D. Huang. Graph-based optimal reactant minimization for sample preparation on digital microfluidic biochips. In *2013 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pages 1–4. IEEE, 2013.

[4] Patrick W. Cooley, David B. Wallace, and Bogdan V. Antohe. Applications of ink-jet printing technology to BioMEMS and microfluidic systems. In *Micromachining and Microfabrication*, pages 177–188. International Society for Optics and Photonics, 2001.

[5] Miguel Coviello Gonzalez and Marek Chrobak. Towards a theory of mixing graphs: a characterization of perfect mixability. In *International Conference on Algorithms and Complexity*, pages 187–198. Springer, 2019.

[6] IBM ILOG CPLEX. V12. 1: User's manual for CPLEX. *International Business Machines Corporation*, 46(53):157, 2009.

[7] Trung A. Dinh, Shinji Yamashita, and Tsung Y. Ho. A network-flow-based optimal sample preparation algorithm for digital microfluidic biochips. In *19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 225–230. IEEE, 2014.

[8] Kathryn A. Dowsland and William B. Dowsland. Packing problems. *European Journal of Operational Research*, 56(1):2–14, 1992.

[9] Shirit Einav, Doron Gerber, Paul D. Bryson, Ella H. Sklan, Menashe Elazar, Sebastian J. Maerkl, Jeffrey S. Glenn, and Stephen R. Quake. Discovery of a hepatitis C target and its pharmacological inhibitors by microfluidic affinity analysis. *Nature Biotechnology*, 26(9):1019–1027, 2008.

[10] Scott A. Elrod, John S. Fitch, and David K. Biegelsen. Low cost piezo printhead based on microfluidics in printed circuit board and screen-printed piezoelectrics, March 25 2008. US Patent 7,347,533.

[11] Mikael Evander, Linda Johansson, Tobias Lilliehorn, Jure Piskur, Magnus Lindvall, Stefan Johansson, Monica Almqvist, Thomas Laurell, and Johan Nilsson. Noninvasive acoustic cell trapping in a microfluidic perfusion system for online bioassays. *Analytical Chemistry*, 79(7):2984–2991, 2007.

[12] Shimon Even, Alon Itai, and Adi Shamir. On the complexity of time table and multi-commodity flow problems. In *16th Annual Symposium on Foundations of Computer Science.*, pages 184–193. IEEE, 1975.

[13] Pietro Ferraro, Lisa Miccio, Simonetta Grilli, Andrea Finizio, Sergio De Nicola, and Veronica Vespini. Manipulating thin liquid films for tunable microlens arrays. *Optics & Photonics News*, 19:34–34, 2008.

[14] Thomas Franke, Adam R. Abate, David A. Weitz, and Achim Wixforth. Surface acoustic wave (SAW) directed droplet flow in microfluidics for PDMS devices. *Lab on a Chip*, 9(18):2625–2627, 2009.

[15] James Friend and Leslie Y. Yeo. Microscale acoustofluidics: Microfluidics driven via acoustics and ultrasonics. *Reviews of Modern Physics*, 83(2):647, 2011.

[16] Harold N. Gabow. An efficient implementation of Edmonds' algorithm for maximum matching on graphs. *Journal of the ACM (JACM)*, 23(2):221–234, 1976.

[17] David Goldberg, Jon Feldman, and Cliff Stein. The integral unit-capacity maximum flow problem with homologous arcs. Technical report, Working Paper, Columbia University, New York, NY, 2006.

[18] S. Grilli, L. Miccio, V. Vespini, A. Finizio, S. De Nicola, and Pietro Ferraro. Liquid micro-lens array activated by selective electrowetting on lithium niobate substrates. *Optics Express*, 16(11):8084–8093, 2008.

[19] Erika Check Hayden. The automated lab. *Nature News*, 516(7529):131, 2014.

[20] Elad Hazan, Shmuel Safra, and Oded Schwartz. On the complexity of approximating k-set packing. *Computational Complexity*, 15(1):20–39, 2006.

[21] Matthew A. Holden, Saurabh Kumar, Ali Beskok, and Paul S. Cremer. Microfluidic diffusion diluter: bulging of PDMS microchannels under pressure-driven flow*. *Journal of Micromechanics and Microengineering*, 13(3):412, 2003.

[22] Matthew A. Holden, Saurabh Kumar, Edward T. Castellana, Ali Beskok, and Paul S. Cremer. Generating fixed concentration arrays in a microfluidic device. *Sensors and Actuators B: Chemical*, 92(1):199–207, 2003.

[23] Frank Hsieh, Hasmik Keshishian, and Craig Muir. Automated high throughput multiple target screening of molecular libraries by microfluidic MALDI-TOF MS. *Journal of Biomolecular Screening*, 3(3):189–198, 1998.

[24] Yi L. Hsieh, Tsung Y. Ho, and Krishnendu Chakrabarty. A reagent-saving mixing algorithm for preparing multiple-target biochemical samples using digital microfluidics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31(11):1656–1669, 2012.

[25] Juinn D. Huang and Chia H. Liu. Sample preparation for droplet-based microfluidics. In *14th International Symposium on Integrated Circuits (ISIC)*, pages 364–367. IEEE, 2014.

[26] Juinn D. Huang, Chia H. Liu, and Ting W. Chiang. Reactant minimization during sample preparation on digital microfluidic biochips using skewed mixing trees. In *Proceedings of the International Conference on Computer-Aided Design*, pages 377–383. ACM, 2012.

[27] Juinn D. Huang, Chia H. Liu, and Huei S. Lin. Reactant and waste minimization in multitarget sample preparation on digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(10):1484–1494, 2013.

[28] Mais J. Jebrail and Aaron R. Wheeler. Digital microfluidic method for protein extraction by precipitation. *Analytical Chemistry*, 81(1):330–335, 2008.

[29] Klavs F. Jensen. Silicon-based microchemical systems: characteristics and applications. *MRS Bulletin*, 31(02):101–107, 2006.

[30] Noo L. Jeon, Stephan K. W. Dertinger, Daniel T. Chiu, Insung S. Choi, Abraham D. Stroock, and George M. Whitesides. Generation of solution and surface gradients using microfluidic systems. *Langmuir*, 16(22):8311–8316, 2000.

[31] Kelly Karns and Amy E. Herr. Human tear protein analysis enabled by an alkaline microfluidic homogeneous immunoassay. *Analytical Chemistry*, 83(21):8115–8122, 2011.

[32] Choong Kim, Kangsun Lee, Jong H. Kim, Kyeong S. Shin, Kyu J. Lee, Tae S. Kim, and Ji Y. Kang. A serial dilution microfluidic device using a ladder network generating logarithmic or linear concentrations. *Lab on a Chip*, 8(3):473–479, 2008.

[33] Kangsun Lee, Choong Kim, Byungwook Ahn, Rajagopal Panchapakesan, Anthony R. Full, Ledum Nordee, Ji Y. Kang, and Kwang W. Oh. Generalized serial dilution module for monotonic and arbitrary microfluidic gradient generators. *Lab on a Chip*, 9(5):709–717, 2009.

[34] Kangsun Lee, Choong Kim, Geunhui Jung, Tae S. Kim, Ji Y. Kang, and Kwang W. Oh. Microfluidic network-based combinatorial dilution device for high throughput screening and optimization. *Microfluidics and Nanofluidics*, 8(5):677–685, 2010.

[35] Kangsun Lee, Choong Kim, Youngeun Kim, Byungwook Ahn, Jaehoon Bang, Jungkwun Kim, Rajagopal Panchapakesan, Yong K. Yoon, Ji Y. Kang, and Kwang W. Oh. Microfluidic concentration-on-demand combinatorial dilutions. *Microfluidics and Nanofluidics*, 11(1):75–86, 2011.

[36] Cheuk W. Li, Rongsheng Chen, and Mengsu Yang. Generation of linear and non-linear concentration gradients along microfluidic channel by microtunnel controlled stepwise addition of sample solution. *Lab on a Chip*, 7(10):1371–1373, 2007.

[37] Peng Li, Zackary S. Stratton, Ming Dao, Jerome Ritz, and Tony J. Huang. Probing circulating tumor cells in microfluidics. *Lab on a Chip*, 13(4):602–609, 2013.

[38] Chia H. Liu, Kuo C. Shen, and Juinn D. Huang. Reactant minimization for sample preparation on microfluidic biochips with various mixing models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(12):1918–1927, 2015.

[39] Chien-Hung Lu, Nicolas C. Pégard, and Jason W. Fleischer. Flow-based structured illumination. *Applied Physics Letters*, 102(16):161115, 2013.

[40] Leanne Marle and Gillian M. Greenway. Microfluidic devices for environmental monitoring. *TrAC Trends in Analytical Chemistry*, 24(9):795–802, 2005.

[41] Carl D. Meinhart and Hongsheng Zhang. The flow structure inside a microfabricated inkjet printhead. *Journal of Microelectromechanical Systems*, 9(1):67–75, 2000.

[42] Carol A. Meyers and Andreas S. Schulz. Integer equal flows. *Operations Research Letters*, 37(4):245–249, 2009.

[43] Debasis Mitra, Sandip Roy, Krishnendu Chakrabarty, and Bhargab B. Bhattacharya. On-chip sample preparation with multiple dilutions using digital microfluidics. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 314–319. IEEE, 2012.

[44] Kwang W. Oh, Kangsun Lee, Byungwook Ahn, and Edward P. Furlani. Design of pressure-driven microfluidic networks using electric circuit analogy. *Lab on a Chip*, 12(3):515–545, 2012.

[45] Phil Paik, Vamsee K. Pamula, Michael G. Pollack, and Richard B. Fair. Electrowetting-based droplet mixers for microfluidic systems. *Lab on a Chip*, 3(1):28–33, 2003.

[46] Amandeep Parmar. *Integer programming approaches for equal-split network flow problems.* PhD thesis, Georgia Institute of Technology, 2007.

[47] Michal Pióro, Aron Szentesi, János Harmatos, Alpár Jüttner, Piotr Gajowniczek, and Stanislaw Kozdrowski. On open shortest path first related network optimization problems. *Performance Evaluation*, 48(4):201–223, 2002.

[48] Michael G. Pollack, Alexander D. Shenderov, and Richard B. Fair. Electrowetting-based actuation of droplets for integrated microfluidics. *Lab on a Chip*, 2(2):96–101, 2002.

[49] Hong Ren, Vijay Srinivasan, and Richard B. Fair. Design and testing of an interpolating mixing architecture for electrowetting-based droplet-on-chip chemical dilution. In *12th International Conference on TRANSDUCERS, Solid-State Sensors, Actuators and Microsystems*, volume 1, pages 619–622. IEEE, 2003.

[50] Sandip Roy, Bhargab B. Bhattacharya, and Krishnendu Chakrabarty. Optimization of dilution and mixing of biochemical samples using digital microfluidic biochips. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(11):1696–1708, 2010.

[51] Sudip Roy, Bhargab B. Bhattacharya, Partha P. Chakrabarti, and Krishnendu Chakrabarty. Layout-aware solution preparation for biochemical analysis on a digital microfluidic biochip. In *24th International Conference on VLSI Design (VLSI Design)*, pages 171–176. IEEE, 2011.

[52] Sudip Roy, Partha P. Chakrabarti, Krishnendu Chakrabarty, and Bhargab B. Bhattacharya. Waste-aware single-target dilution of a biochemical fluid using digital microfluidic biochips. *Integration, the VLSI Journal*, 51:194–207, 2015.

[53] Wajeeh Saadi, Shur J. Wang, Francis Lin, and Noo L. Jeon. A parallel-gradient microfluidic chamber for quantitative analysis of breast cancer cell chemotaxis. *Biomedical Microdevices*, 8(2):109–118, 2006.

[54] Jennifer Sager, Maxwell Young, and Darko Stefanovic. Characterization of transverse channel concentration profiles obtainable with a class of microfluidic networks. *Langmuir*, 22(9):4452–4455, 2006.

[55] V. I. Shevchik. Multicommodity flow problem. *Journal of Mathematical Sciences*, 65(1):1462–1464, 1993.

[56] Jinjie Shi, Xiaole Mao, Daniel Ahmed, Ashley Colletti, and Tony J. Huang. Focusing microparticles in a microfluidic channel with standing surface acoustic waves (SSAW). *Lab on a Chip*, 8(2):221–223, 2008.

[57] Samuel K. Sia and George M. Whitesides. Microfluidic devices fabricated in poly (dimethylsiloxane) for biological studies. *Electrophoresis*, 24(21):3563–3576, 2003.

[58] Todd M. Squires and Stephen R. Quake. Microfluidics: Fluid physics at the nanoliter scale. *Reviews of Modern Physics*, 77(3):977, 2005.

[59] K. Srinathan, Pranava R. Goundan, M. Ashwin Kumar, R. Nandakumar, and C. Pandu Rangan. Theory of equal-flows in networks. In *International Computing and Combinatorics Conference*, pages 514–524. Springer, 2002.

[60] Vijay Srinivasan, Vamsee Pamula, Michael Pollack, and Richard Fair. A digital microfluidic biosensor for multianalyte detection. In *IEEE The Sixteenth Annual International Conference on Micro Electro Mechanical Systems (MEMS)*, pages 327–330. IEEE, 2003.

[61] Vijay Srinivasan, Vamsee K. Pamula, and Richard B. Fair. Droplet-based microfluidic lab-on-a-chip for glucose detection. *Analytica Chimica Acta*, 507(1):145–150, 2004.

[62] Vijay Srinivasan, Vamsee K. Pamula, and Richard B. Fair. An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids. *Lab on a Chip*, 4(4):310–315, 2004.

[63] Howard A. Stone, Abraham D. Stroock, and Armand Ajdari. Engineering flows in small devices: microfluidics toward a lab-on-a-chip. *Annual Review of Fluid Mechanics*, 36:381–411, 2004.

[64] Robert E. Tarjan and Anthony E. Trojanowski. Finding a maximum independent set. *SIAM Journal on Computing*, 6(3):537–546, 1977.

[65] William Thies, John P. Urbanski, Todd Thorsen, and Saman Amarasinghe. Abstraction layers for scalable microfluidic biocomputing. *Natural Computing*, 7(2):255–275, 2008.

[66] Elisabeth Verpoorte. Microfluidic chips for clinical and forensic analysis. *Electrophoresis*, 23(5):677–712, 2002.

[67] Glenn M. Walker, N. Monteiro Riviere, Jillian Rouse, and Adrian T. O'Neill. A linear dilution microfluidic device for cytotoxicity assays. *Lab on a Chip*, 7(2):226–232, 2007.

[68] Yi Wang, Aditya S. Bedekar, S. Krishnamoorthy, Sachin S. Siddhaye, and Shivshankar Sundaram. System-level modeling and simulation of biochemical assays in lab-on-a-chip devices. *Microfluidics and Nanofluidics*, 3(3):307–322, 2007.

[69] George M. Whitesides. The origins and the future of microfluidics. *Nature*, 442(7101):368–373, 2006.

[70] Martin Wiklund, Roy Green, and Mathias Ohlin. Acoustofluidics 14: Applications of acoustic streaming in microfluidic devices. *Lab on a Chip*, 12(14):2438–2451, 2012.

[71] Achim Wixforth. Acoustically driven programmable microfluidics for biological and chemical applications. *Journal of the Association for Laboratory Automation*, 11(6):399–405, 2006.

[72] Tao Xu, Krishnendu Chakrabarty, and Vamsee K. Pamula. Defect-tolerant design and optimization of a digital microfluidic biochip for protein crystallization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(4):552–565, 2010.

[73] Tao Xu, Vamsee K. Pamula, and Krishnendu Chakrabarty. Automated, accurate, and inexpensive solution-preparation on a digital microfluidic biochip. In *Biomedical Circuits and Systems Conference (BioCAS)*, pages 301–304. IEEE, 2008.

[74] Abhimanyu Yadav, Trung A. Dinh, Daiki Kitagawa, and Shigeru Yamashita. ILP-based synthesis for sample preparation applications on digital microfluidic biochips. In *29th International Conference on VLSI Design and 15th International Conference on Embedded Systems (VLSID)*, pages 355–360. IEEE, 2016.

[75] Hayat A. Yusuf, Sara J. Baldock, Robert W. Barber, Peter R. Fielden, Nick J. Goddard, and Bernard J. T. Brown. Novel microsystems for concentration gradient generation through computer optimization with validation using optical instrumentation. *Microelectronic Engineering*, 85(5):1265–1268, 2008.

[76] Hayat A. Yusuf, Sara J. Baldock, Robert W. Barber, Peter R. Fielden, Nick J. Goddard, Stephan Mohr, and Bernard J. T. Brown. Optimisation and analysis of microreactor designs for microfluidic gradient generation using a purpose built optical detection system for entire chip imaging. *Lab on a Chip*, 9(13):1882–1889, 2009.

[77] Hayat A. Yusuf, Sara J. Baldock, Peter R. Fielden, Nick J. Goddard, Stephan Mohr, and Bernard J. T. Brown. Systematic linearisation of a microfluidic gradient network with unequal solution inlet viscosities demonstrated using glycerol. *Microfluidics and Nanofluidics*, 8(5):587–598, 2010.

[78] Weixiong Zhang and Richard E. Korf. A study of complexity transitions on the asymmetric traveling salesman problem. *Artificial Intelligence*, 81(1):223–239, 1996.

Appendix A

Technologies for Fluid-Based Mixing

A.1 Proportional Network

In [36], Li *et al.* proposed a microfluidic device capable of generating linear and non-linear concentration gradients by having the reactant and the buffer fluids mix proportionally along intermediate micro-channels.

They argue that the flow rate across each micro-channel is proportional to their length. So if multiple micro-channels have the same length, the same amount of reactant volume will flow through them.

Figure A.1 shows the structure of a proportional network where all micro-channels have the same length. Such network generates a linear concentration gradient of fluids. The concentration of the fluids is measured at the corresponding cavities along the doted line.

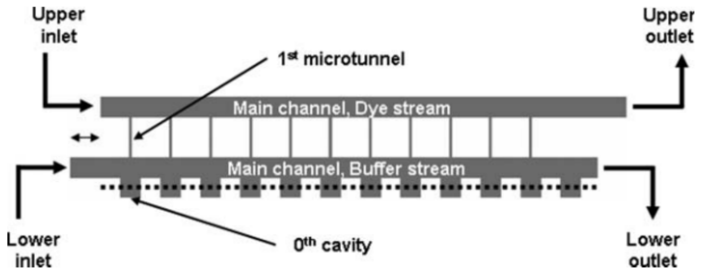

Figure A.1: Proportional network structure. Main channels are the input micro-channels for buffer and reactant (dye) respectively. (Figure taken from [36].)

A.2 T-type Network

A laminar microfluidic convective/diffusive mixing chip is presented in [22] by Holden *et al.*. The mixing chip consists of a T-intersection of two input fluids into a main micro-channel. The input fluids are mixed by a convective/diffusive transport of nanoliter quantities of volume of each fluid along the intersecting micro-channel. Finally, the fluids are divided into a parallel array of independent micro-channels. The target concentrations can be approximated by regulating the flow rate of each input fluid.

Holden *et al.* computations include Navier-Stokes, species transport and continuity equations to approximate the convective/diffusive behavior of the fluids being mixed. These equations involve attributes of the fluids such as velocity, viscosity, density and mass diffusivity.

Also, they extended their approach to polydimethylsiloxane-based mixing chips in [21], which are more flexible than the rigid laminar chips. Figure A.2 illustrates the process of two fluids (Alexa594 and Alexa488) being mixed. The image is only a schematic representation that shows the overall structure of the mixing chip, how the fluids are mixed

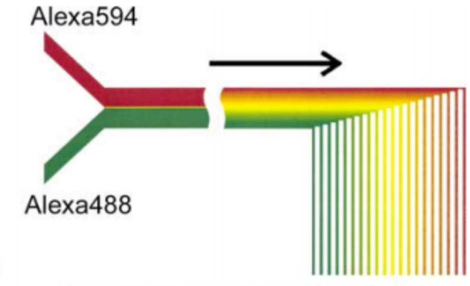

Figure A.2: Example of the T-typed convective/diffusive mixing chip where two fluids are being mixed, Alexa594 and Alexa488 respectively (taken from [22]).

together along the intersecting micro-channel, and how the fluids are routed into the corresponding output micro-channels.

A.3 Serial Network

Kim *et al.* in [32] proposed a serial ladder mixing chip able to generate logarithmic and linear step-wise multiple target concentrations.

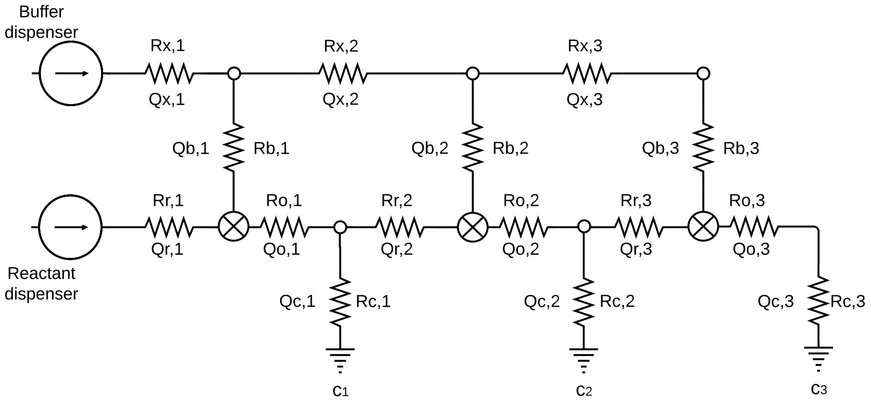

Figure A.3: Serial ladder network structure for three target concentrations.

Their design is analogous to an electric circuit, where the current sources, resistors and grounds represent fluid dispensers, micro-channels' hydrodynamic resistances and fluid output storages, respectively. Figure A.3 illustrates the structure of a serial ladder network for three target concentrations, c_1, c_2, c_3 respectively. $Q_{x,k}$ and $Q_{r,k}$ are the input flow rates at the k^{th} step. $Q_{b,k}$ and $Q_{r,k}$ are the flow rates of the buffer and reactant (already diluted if $k > 1$) for the k^{th} mixing step. $Q_{o,k}$ is the output flow rate for the k^{th} mixing operation and $Q_{c,k}$ is the flow rate for the k^{th} target concentration c_k. Also, each flow rate Q is associated with a resistance R. Micro-mixers are represented with the symbol $\bigotimes$.

Given the target set of concentrations $T = \{c_1, c_2, \ldots, c_n\}$ with their corresponding flow rates $Q_c = \{Q_{c,1}, Q_{c,2}, \ldots, Q_{c,n}\}$, the dilution ratios $\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$ between buffer and reactant at each micro-mixer can be computed using the following equations:

$$c_k = \gamma_k \cdot c_{k-1}$$

$$\gamma_k = \frac{Q_{r,k}}{Q_{b,k} + Q_{r,k}}$$

Only after that, the remaining flow rates can be computed by a series of algebraic equations. The computation goes backwards from the last mixing step, where $Q_{b,n} = (1 - \gamma_n) \cdot Q_{c,n}$ and $Q_{r,n} = \gamma \cdot Q_{c,n}$.

Next, the flow rates in the mixing chip are controlled by adjusting the hydrodynamic resistance of the corresponding micro-channels. At each step there are five resistances $R_{b,k}, R_{r,k}, R_{o,k}, R_{x,k}, R_{c,k}$ respectively, from which $R_{b,k}, R_{r,k}, R_{o,k}$ are fixed as constants to maintain the functionality of the mixing chip while $R_{x,k}, R_{c,k}$ are left as adjustable resistances.

As the flow rate through each resistance is known, the necessary resistances are computed using Kirchhoff's voltage law. Also, the dimensions of the micro-channels are computed by analyzing the fixed resistances of the micro-mixers and their connecting micro-channels' lengths.

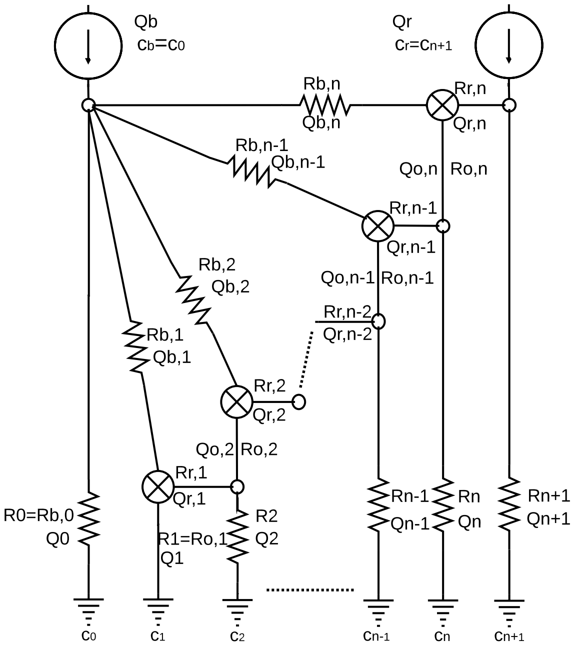

Figure A.4: Serial cascade network structure for n target concentrations.

Kim *et al.* also proposed a serial cascade mixing chip able to generate monotonic and arbitrary multiple target concentrations in [33], which is an extension to the serial ladder network. However, in the serial cascade network, each buffer flow rate is independent from the buffer flow rates of previous steps, which makes it more powerful than the serial ladder network. Figure $A.4$ shows the structure of a serial cascade network for n target concentrations.

A.4 Combinatorial Network

In [34, 35], Lee *et al.* proposed a combinatorial LoC mixing chip that produces seven combinations out of three input reagents and buffer. If the input reagents are A, B and C, then the possible combinations include $\{ABC, AB, BC, AC, A, B, C\}$.

The proposed mixing chip consists of an initial concentration control module followed by a combinatorial dilution module. The concentration control module basically generates different concentrations of each reagent by diluting each reagent with buffer only. This is accomplished by having a common channel share all the input fluids in some order and proportion such that each reagent is only combined with buffer in this phase.

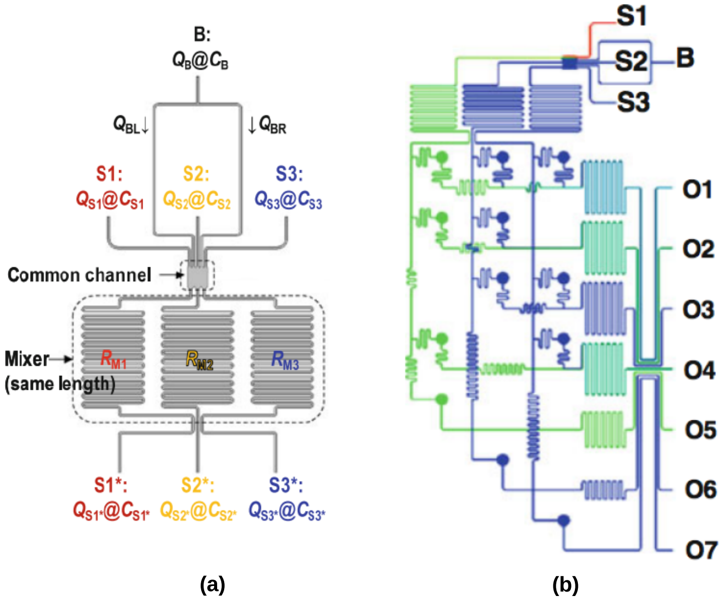

Figure A.5: Combinatorial mixing chip modules. Image (a) illustrates the concentration control module. Image (b) shows the entire combinatorial network of the mixing chip. (Figures taken from [35].)

Figure A.5 (a) shows the structure of the concentration control module, where S_1, S_2, S_3 and B (B_L, B_R) are the input micro-channels for the three reagents and the buffer, respectively, and Q_{S1}, Q_{S2}, Q_{S3}, Q_{BL} and Q_{BR} are their corresponding flow rates. C_{S1}, C_{S2} and C_{S3} are the concentrations of the three reagents, and S_1^*, S_2^* and S_3^* are the output micro-channels for the diluted reagents. The common channel has all input micro-channels as input in the following order: $\{S_1, B_L, S_2, B_L, S_3\}$. As all reagents share this channel (being diverged by buffer), there is a limit on the flow rates that fluids can have such that reagents are not intermixed accidentally. This implies that the concentration gradient that the control module can generate is limited.

The combinatorial dilution module takes the outputs of the concentration control module as intermediate inputs. Then, the intermediate input fluids selectively flow into seven parallel mixing-channels. Given the flow rates and concentrations of the target fluids, all micro-channels' lengths can be computed. Also, using the analogy between microfluidics and electrical circuits, the hydrodynamic resistances at the corresponding micro-channels can be determined by using Ohm's and Kirchhoff's laws. Figure A.5 (b) illustrates the complete network structure of the combinatorial device, where outputs $O1, O2, O3, O4, O5, O6$ and $O7$ represent the reagent combinations ABC, AB, BC, AC, A, B and C, respectively. At the top of the image we have the concentration control module, which is connected to the combinatorial dilution module by the corresponding three output micro-channels.

A.5 Pyramidal Network

In [30], Jeon *et al.* proposed a left-right symmetric pyramidal mixing chip network model for any combination of x input fluids with a gradient output generator of y concentrations, for $x < y$. Such mixing chip consists of $y - x + 1$ levels of vertical serpentine-shaped micro-channels, where the output fluids of each such micro-channel (but the ones at the bottom of the pyramid) are being split into two.

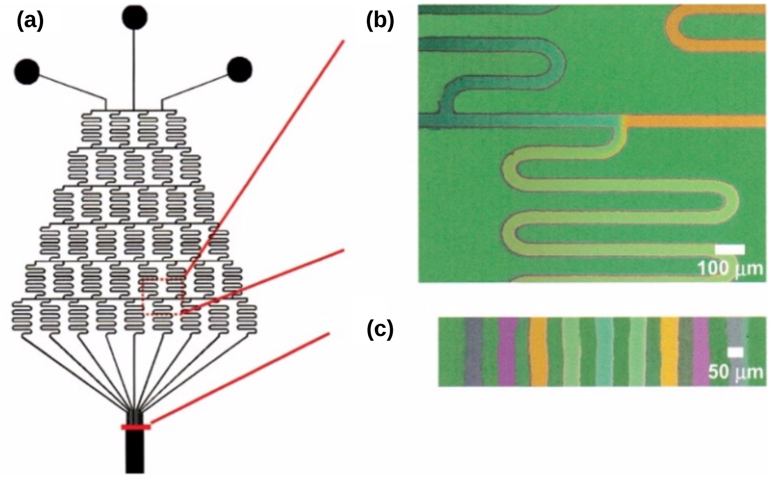

Figure A.6: Pyramidal mixing chip experiment for the dilution of hydrofluoric acid with water. Image (a) shows the network structure of the mixing chip. Image (b) shows two fluids being diffusively mixed. Image (c) shows the output concentration gradient. (Figures taken from [30].)

The mixing chip generates the gradient of concentrations by sequentially splitting and mixing the fluids simultaneously at each level in the network, where fluids travel downwards from the top of the pyramid. Figure $A.6$ (a) illustrates a 4-input 9-output pyramidal mixing chip network design.

The pyramidal mixing chip can be also analyzed with the electrical circuit analogy, as discussed in the previous sections. Vertical micro-channels' lengthiness comes from the fact that fluids are being diffusively mixed along them. Thus, it is important that vertical micro-channels' lengths are sufficiently large to ensure a complete equilibration of the concentrations of the fluids being mixed. The concentration at each micro-channel's output is computed based on the flow rates of the input fluids and the resistances and lengths of the corresponding micro-channels.

Both static and dynamic gradients can be generated by varying the pressure on the fluid dispensers. Also, by increasing the number of levels in the pyramidal network (and thus the number of output micro-channels), a higher gradient resolution of concentrations can be obtained.

Figure $A.6$ illustrates an experiment where hydrofluoric acid is being used as reactant (middle fluid dispenser) and water as buffer (outer fluid dispensers). Image (a) shows the micro-chip network, image (b) shows two fluids being diffusively mixed by a vertical serpentine-shaped micro-channel at the bottom of the pyramid and image (c) shows the output concentration gradient of the mixing chip.

There has been multiple improvements to the model described in [30], some of them include [2, 54, 68, 75–77].