Exact Synthesis of Biomolecular Protocols for Multiple Sample Pathways on Digital Microfluidic Biochips

Oliver Keszocze*†, Mohamed Ibrahim‡, Robert Wille†§, Krishnendu Chakrabarty‡, Rolf Drechsler*†

*Institute of Computer Science, University of Bremen, Bremen, Germany
†Cyber-Physical Systems, DFKI GmbH, Bremen, Germany,
‡Department of Electrical and Computer Engineering, Duke University, Durham, NC
§Institute for Integrated Circuits, Johannes Kepler University, Linz, Austria

{keszocze,drechsler}@informatik.uni-bremen.de {mohamed.s.ibrahim, krish}@duke.edu robert.wille@jku.at

***Abstract*—*Digital-microfluidic biochips* (DMFBs) allow laboratory steps associated with genomic bioassays to be carried out in an automated and highly efficient fashion. However, to support complex (multi-sample) biomolecular protocols for quantitative analysis, conventional design-automation methods are ineffective, thereby motivating the need for more suitable solutions. While first methods to address this problem have recently been introduced, they neglect architectural details and technology-related constraints, and use heuristics to oversimplify the search for a design solution. These simplifications result in designs that may either not be realizable on a DMFB or it is not possible to evaluate how far they are from being optimal. In this work, we propose an exact synthesis method that does not rely on simplistic models and heuristics, but solves the underlying design problem in an exact fashion. The proposed method allows for the determination of exact and realistic results for the realization of quantitative-analysis biomolecular protocols on DMFBs.**

I. Introduction

Digital-microfluidic biochips (DMFBs) have emerged as an enabling lab-on-a-chip technology for contemporary biomolecular research [1]. In contrast to conventional benchtop procedures, DMFBs can precisely control a large number of droplets and biomolecular assay parameters, e.g., protein or messenger RNA expression levels, in a high-throughput manner [2]. Using digital-microfluidics technology, cells are encapsulated in picoliter droplets, electroporated, mixed with reagents such as reverse-transcription components, and heated for biological-signal amplification using a grid of electrodes that are driven by a sequence of actuation voltages from a computer [3]. As a result, genomic bioassays such as nucleic-acid isolation, DNA purification, and DNA amplification can be conducted on a DMFB [4], [5].

In practice, many DMFBs are partitioned into dedicated reaction chambers that are capable of performing different fluid-handling operations. This architecture allows for a modular configuration that facilitates the design of plug-and-play¡ devices, improves fault tolerance, and increases system throughput by scaling up the number of reactions while avoiding reaction interference among different samples [6], [7]. The chambers themselves are then connected by a corresponding topology that is tailored for the target application, whereas the ring-bus architecture is commonly used for infection disease tests or sample preparation applications. For example, both GenMark's recent ePlex system for diagnostics as well as Illumina's NeoPrep solution employ this topology. Fig. 1 shows a commercialized chip by Illumina, where the reaction chambers and the ring are highlighted by red rectangles and a line, respectively.

Today's microbiology applications (e.g., quantitative gene-expression analysis) may include tens of independent sample pathways running through a sequence of procedural decision points (see e.g., [8]). Moreover, multiple reaction pathways, which enable high-resolution reaction-time control, are necessary for producing synthesized organic compounds. Such a laboratory framework, known as flash chemistry [9], [10], requires accelerated reactions and it is extremely valuable for the pharmaceutical industry. Even a small reduction in reaction time obtained using an exact method is significant enabler for flash chemistry Therefore, design automation for integrated biomolecular analysis and support for multiple sample pathways have emerged as a challenging problem. To address this problem, Ibrahim et al. introduced the first design-automation ("synthesis") methodology to support non-trivial biomolecular analysis [11] based on a ring-bus architecture. The proposed synthesis method conducts greedy resource allocation and dynamic mapping of multiple sample pathways to biochip resources.

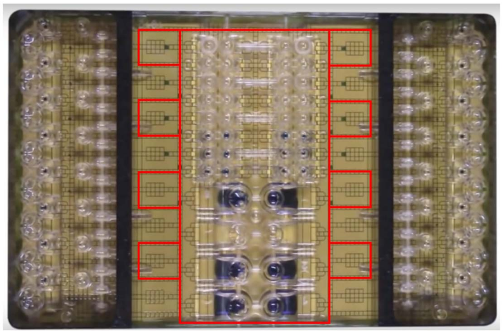


Fig. 1: Illumina's ring-based NeoPrep solution [12].

However, despite the novelty of [11], there are several drawbacks and shortcomings that limit its applicability:

- The method relies on an abstract view of the underlying DMFB, which ignores important architectural details. Because of this, the obtained synthesis results are often infeasible for the given DMFB architecture.
- The method ignores realistic technology-related constraints, such as the biochip's lifetime or the degradation level of the electrodes. As a result. the synthesized designs are likely to lead to faults on the DMFB and, consequently, incorrect outcomes.
- The method relies on heuristics that oversimplify the search for a design solution. Accordingly, it is not possible to evaluate how far the results are from being optimal. Therefore, the heuristic method does not support fine-grained control of pathway reaction time in flash chemistry.

In this work, we propose a synthesis method that overcomes these problems. Instead of relying on greedy heuristics and simplistic abstractions, an *exact* methodology is provided that is capable of determining optimal results while, at the same time, satisfying architectural constraints. Moreover, the pro-

2380-6923/17 $31.00 © 2017 IEEE
DOI 10.1109/VLSID.2018.48

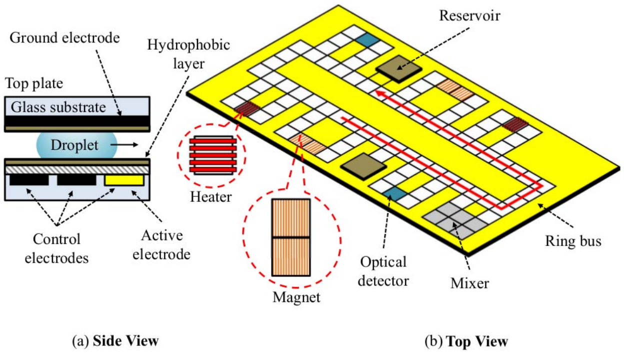


Fig. 2: Schematic view of a DMFB: (a) a side view; (b) a ring-based DMFB with 2D array and on-chip resources.

posed solution can easily be enriched by additional constraints that incorporate realistic technology-related constraints.

Experimental evaluations demonstrate the improvements accomplished with this methodology. For the first time, exact results for the realization of quantitative-analysis biomolecular protocols (with multiple sample pathways) on DMFBs are obtained, which allow to evaluate the impact of abstracting the architectural characteristics as well as the greedy schemes from the previous solution. Our results show that the gap between exact and heuristic solutions is significant and it is magnified for a larger number of pathways in realistic applications. In addition, the proposed approach also allows us to evaluate, for the first time, how ignoring technology-related constraints affects the resulting design.

II. Background and Motivation

This section describes the related background on DMFBs as well as synthesis for biomolecular protocols. It also discusses the shortcomings of the state-of-the-art and motivates the contribution of this work.

DMFBs execute biomolecular protocols by manipulating nanoliter droplets of samples and reagents through a set of elementary fluid-handling operations, such as mixing, incubation, dilution, and heating. As shown in Fig. 2(a), a typical design of an DMFBs consists of two parallel plates coated with a hydrophobic layer. The top plate represents a single conducting layer that is always grounded, whereas the bottom plate comprises an array of control electrodes that can be programmed to move sandwitched droplets using electrowetting [1]. Similar to the field-programmable gate arrays, the array-based design of DMFBs embraces their flexibility, since the majority of fluid-handling operations can be performed anywhere on the array.

Early synthesis methods aimed at leveraging the full flexibility of DMFBs by introducing solutions for scheduling of fluid-handling operations, resource binding, and droplet routing [13], [14], [15], [16], [17], [18]. Such solutions attempt to efficiently synthesize any given protocol in a sample-in-result-out manner, while dynamic re-synthesis is only limited to counteract error occurrences observed in a single sample pathway [19]. Although these methods allowed for the exploitation of the inherent flexibility of DMFBs, they are restricted to simple droplet manipulation only. That is, they are not adequate for realistic biomolecular analysis.

In order to address this problem, a new synthesis methodology has recently been introduced in [11]. It tackles the complexity of microbiology protocols (e.g., gene-expression analysis) by cyber-physical adaptation [20], [21] and abstraction. A framework has been proposed that concurrently controls multiple sample pathways and promptly responds to decisions about the protocol flow at each pathway. To provide architectural support, the concept of virtual topologies from [22] has been borrowed and the DMFB has been assumed to have a unidirectional ring-based topology as shown in Fig. 2(b). Here, a circular array of electrodes forms a ring that is connected to dedicated reaction chambers to perform fluid-handling operations[1]. By this, the actual synthesis problem has been simplified to a resource-allocation problem. However, while the methodology of [11] was able to narrow the gap between DMFBs and biomolecular analysis, it suffers from drawbacks that limit its applicability. More precisely:

A. Abstraction from Architectural Details The synthesis method from [11] relies on an abstract view of the DMFB that ignores architectural details. In particular, the routing between the reaction chambers has been abstracted away. As a result, neither the time needed to move droplets from one reaction chamber to the next one nor collisions of droplets in the ring are considered.

B. Realistic Technology Constraints As DMFBs are getting more involved in point-of-care biomolecular research [24], recent studies have been directed to investigate technology constraints (e.g., lifetime or degradation level of array electrodes [25], [26]) that have to be satisfied to guarantee reliable biochip operations. As a result, design automation researchers are burdened with an extra level of complexity to handle these constraints. In [11], the authors tackled this challenge by presenting a degradation-aware resource allocation, wherein the allocation scheme only restricts the usage of an on-chip resource when its degradation level exceeds the safety margin. That is, the degradation constraint is considered as a soft constraint. This, however, does not guarantee that resources with completely degraded (i.e., broken) electrodes are avoided—leading to possible failures in protocol execution.

C. Greedy Results The solution proposed in [11] relies on heuristics and greedy methods in order to efficiently generate a design. Therefore, the obtained results are often far from being optimal. As an example, consider the protocol to be realized as shown in Fig. 3(a). A ring-based DMFB is available which integrates on-chip resources as shown in Fig. 3(b). The greedy method introduced in [11] scans all pathways and allocates resources to bioassays in a sequential manner such that resources with a lower degradation level are given a higher priority during allocation. Note that, due to the greedy nature of the solution, the algorithm enforces the degradation-aware policy by ensuring that the resources previously selected to execute Bioassay I, such as Detector 1, cannot be reserved to execute an operation in Bioassay II. This yields a solution as shown on the top of Fig. 3(c) with a total completion time of 15 time units[2]. In contrast, an optimal solution only requires 14 time units (as shown on the bottom of Fig. 3(c)). Furthermore, the greedy resource allocation causes the magnet to exceed its safety threshold, potentially leading to a resource breakdown during the execution. The optimal solution, on the other hand, reliably allocates resources so that a safety margin is never exceeded.

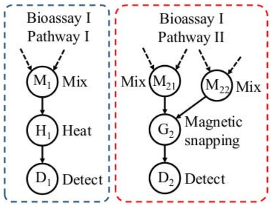
Because of these shortcomings, the method proposed in [11] often leads to inefficient solutions that violate technology-related constraints and are far from being optimal.

III. Problem Formulation and General Idea

The above discussion motivates the following design problem:
Given: A multi-sample biomolecular protocol and an DMFB architecture.

[1] This topology is accordingly applied in commercial solutions e.g. provided by Illumina or GenMark [12], [23].

[2] For the sake of clarity, scheduling time and degradation levels are computed in terms of time units (e.g., seconds).

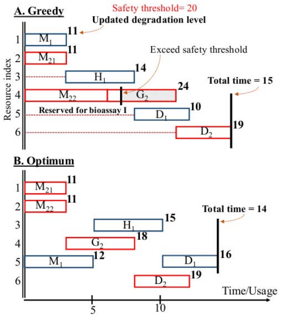


(a) Protocol

#	Resources	N.O.T	M.T	Initial D.L
1	2x3 mixer 1	-	3	8
2	2x3 mixer 2	-	3	8
3	Heater	5	6	10
4	Magnet	5	6	13
5	Detector 1	4	6	6
6	Detector 2	4	6	15

N.O.T: Normal operation time (e.g., heating) **D.L:** Degradation level
M.T: Mixing time when the resource is used for mixing

(b) Chip Resources



(c) Resource Allocation

Fig. 3: An example showing a comparison between the performance of the greedy method in [11] and the optimal behaviour.

Design Goal: A DMFB design realizing the protocol (i.e., a resource allocation and scheduling).
Constraint: Technology constraints (e.g., electrode degradation).
Objective: Minimize the protocol completion time.

The protocol is specified by fluid-handling operations that are arranged in multiple sequencing graphs (associated with different bioassays), as shown in Fig. 3(a). Note that, although two independent sequencing graphs are shown here, they may interact with each other when they both have to be implemented on a resource-limited DMFB, i.e., they compete for on-chip resources. In addition, each sample pathway comprises a tree of sequencing graphs—a sequencing graph is selected for execution at a certain stage based on an intermediate decision point within the protocol flow.

The DMFB architecture is described in terms of a ring-based system, as shown in Fig. 2(b). Using this topology, distances between the resources are known and have to be incorporated when determining a solution. Moreover, droplets can not move past each other on the bus. They can only overtake another droplet that has left the bus to be used in a resource. Finally, technology-related constraints, like e.g., electrode degradation, have to be taken into account. Resources can not be used infinitely but will break after a certain amount of usages. Resource allocation methods need to respect these bounds.

Based on these inputs and objectives, the general idea of the proposed methodology is as follows: First, it is checked whether the desired protocol can be realized on the ring-based DMFB and with the applied constraints at a fix maximal completion time of $T = 1$ (i.e. within a single time step). To this end, a corresponding SMT formulation is created and passed to an SMT solver [27]. If the SMT solver proves that no such realization is possible, T is increased by 1 and the process is repeated. This continues until the SMT solver shows that the desired protocol can be realized under these conditions at a certain time step T. Then, the precise realization is obtained from the solution of the SMT solver. By increasing the value of T in each iteration and by starting with $T = 1$, minimality with respect to the number of time steps is guaranteed. Similar approaches have been successfully employed for, e.g., routing and synthesis [15], [17], [18].

IV. SMT Formulation of the Design Problem

Based on the general idea sketched above, we describe how the design problem is formulated in SMT. More precisely, an SMT formulation is required to represent the following question: Is it possible to realize the given protocol on a ring-based DMFB while, at the same time, satisfying all architectural and technology-related constraints. In this section, details of the corresponding SMT formulation are provided. First, all variables utilized for this purpose are introduced. Next, constraints restricting the set of all possible solutions to only valid ones are described. Passing the resulting formulation to an SMT solver either yields an assignment satisfying all constraints or proves that such an assignment is not possible. In the former case, a realization of the protocol on the DMFB can be obtained. In the latter case, it has been proven that no such realization exists for the currently considered number T of time-steps.

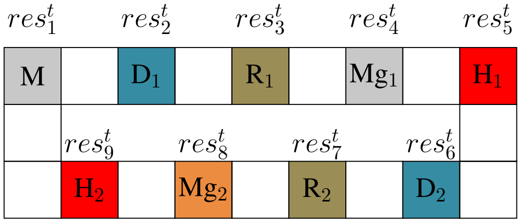


Fig. 4: Variables for the ring-based DMFB from Fig. 2(b).

A. Utilized Variables

In order to formulate the considered design problem as an SMT instance, we introduce an integer variable res_i^t for each resource $i \in R$ (with R representing all possible resources) and for each time step t (with $0 \leq t < T$). Hence, for a protocol composed of n different operations which is to be realized on a ring-based DMFB with k different resources and which should be completed after at most T time steps, we have $0 \leq res_i^t \leq n$ for $1 \leq i \leq n, 0 \leq t < T$. The value of res_i^t describes what operation is executed in resource i at time step t. The value 0 is a special value indicating that no operation is executed in this resource at that time step.

As an example, consider again the ring-based DMFB shown in Fig. 2(b). As can be seen, nine different resources are available. For each of them, a res_i^t variable is introduced as sketched in Fig. 4. Using these variables, the assignment $res_9^2 = 5 \quad res_9^3 = 5 \quad res_9^4 = 5 \quad res_9^5 = 5$ states that operation 5 is executed in time steps 2 to 5 by the heating resource with the identifier 9. Similarly, the variable assignment $res_9^6 = 0$ states that this heating resource is not used in time step 6 (i.e. the previous operation has been completed).

Using these variables, all possible realizations of operations on the DMFB are represented. However, without any further constraints, this formulation permits invalid realizations that, e.g., allow an operation to be executed multiple times (even at the same time step). To prevent this, the assignments of the res_i^t variables have to be restricted.

B. Helper Formulations

In order to properly describe the constraints enforced on the res_i^t variables, we first define a set of *helper formulations*. They describe certain aspects of the overall SMT instance, e.g., whether a certain operation has already been completed. These helpers serve as building blocks for the SMT constraints modeling the overall problem. Besides that, we utilize the notation as summarized in Table I. At the same time, we always provide an intuitive description to ease the readability. Note that, also for sake of readability, corner cases are not explicitly discussed. For example, we are always referring to a predecessor time step by $t - 1$ and do not explicitly discuss the corner case when $t = 0$.

The following helper formulations are used:

- *Executing:* The term $exec(o, t) := \bigvee_{i \in Res(o)} res_i^t = o$, indicates whether the operation o is being executed in time step t.

Table I: Notation used in the SMT formulation.

Symbol	Meaning
i, R	Resources identifier & set of all resources
o, O	Operation identifier and set of all Operations
res_i^t	Variable for usage of resource i at time step t
$Ops(i)$	Operations that can be executed in resource i
$Res(o)$	Resources that can execute operation o
$Prev(o)$	Operations whose result is used by operation o
$Next(o)$	Operations using the results of operation o
$dist(i, j)$	Distances between the resources i and j

- *Starting:* The term $starting(o, t) := \neg exec(o, t-1) \wedge exec(o, t)$ indicates that the operation o is being started in time step t.
- *Completing:* The term
$$completing(o,t) := \neg exec(o,t) \wedge \bigvee_{t'=0}^{t-1} exec(o,t')$$
indicates whether the operation o has completed its operation in time step t.[3]
- *Schedulable:* The term
$$schedulable(o,t) := \bigwedge_{o' \in Prev(o)} completing(o',t),$$
indicates whether the operation o can be scheduled in time step t.
- *Allocation:* We define the term representing an allocation of an operation o to a resource i starting in step t as
$$alloc(o,i,t) := usage(o,i,t) \wedge block(o,i,t),$$
where $usage(o, i, t)$ describes the resource being used and $block(o, i, t)$ ensures that the succeeding nodes of o will not start too early (i.e., respect the time needed to overcome the distances between the resources). Using the abbreviation $t_{end} := t + duration(i) - 1$, $usage$ and $block$ are defined as
$$usage(o,i,t) := \bigwedge_{t'=t}^{t_{end}} res_i^{t'} = o$$
and
$$block(o,i,t) := \bigwedge_{\substack{o' \in Next(o) \\ i' \in Res(o')}} \bigwedge_{t=end}^{end+dist(i,i')} res_{i'}^t \neq o',$$
respectively.

C. Formulation of the Design Problem

Using the variables as well as the helper formulations introduced above, the design problem now can be formulated by means of the following five constraints:

1) Correct Assignment: To ensure that the resources are correctly assigned, e.g., detecting operations are executed in a detecting device, the following constraints are added to the SMT instance
$$\bigwedge_{t=0}^{T} \bigwedge_{i \in R} \left(res_i^t = 0 \vee \bigvee_{o \in Ops(i)} res_i^t = o \right).$$

2) Correct Ordering: To ensure that the operations are scheduled as defined by the sequencing graphs, the following constraints are added
$$\bigwedge_{t=0}^{T-1} \bigwedge_{o \in O} starting(o,t) \Rightarrow schedulable(o,t).$$
This enforces that all predecessors of o have successfully completed before o can start.

3) Correct Allocation: When an operation is starting, a choice has to be made what resource shall be utilized for it. To make sure that exactly one allocation is made (and, at the same time, the chosen resource is blocked for the respective amount of time steps), the constraints
$$\bigwedge_{t=0}^{T-1} \bigwedge_{o \in O} starting(o,t) \Rightarrow \sum_{i \in Res(o)} alloc(o,i,t) = 1$$
are added. The sum ensures that the operation o is allocated in exactly one of the resources i in $Res(o)$.

4) Respect Degradation: To respect the hard technology constraints (here, with respect to degradation), resources can only be used up to a threshold. Each resource i has an associated usage threshold $thres_i$. To enforce this threshold, the constraints
$$\bigwedge_{i \in R} \left(\sum_{t=0}^{T-1} res_i^t \neq 0 \right) \leq thres_i$$
are added. This enforces that the number of time steps in which the resource is occupied by an actual operation (recall that 0 is the special nil operation) stays below the threshold.

5) Enforce Starting: The previous formulations ensure that all solutions adhere to the constraints of the assay. However, up to this point, assigning 0 to all resources for every time step would constitute a valid assignment. Hence, to ensure that operations are actually executed, the constraints
$$\bigwedge_{o \in O} \sum_{t=0}^{T-1} started(o,t) = 1$$
are added. This requires that every operation indeed has to be started exactly once and, by this, enforces the actual execution of the protocol.

V. Applications and Evaluations

The methodology described above significantly improves the previously proposed solution with respect to accuracy as well as by guaranteeing the satisfaction of hard technology constraints. In this section, we demonstrate this by three case studies. We first show how the results obtained by the proposed method (which are optimal and respect architectural characteristics) compare to the results obtained by [11] (which relies on a greedy scheme and ignores architectural characteristics). The next experiment explicitly considers the effect of employing hard technology constraints. Afterwards, the applicability of the proposed method in the context of flash chemistry is evaluated.

Using the proposed methodology, these evaluations can be performed for the first time. To actually conduct the experiments, we make use of the Z3 SMT solver [27].

A. Comparison to Previous Work

In a first series of evaluations, we compared the results obtained by the proposed methodology to previous work [11]. As discussed in Section II, [11] relies on an abstract view of the DMFB that ignores the architectural details and works with greedy schemes. This significantly oversimplifies the search for a design solution. The proposed methodology addresses these shortcomings and, hence, yields more realistic designs.

This has been confirmed on a wide variety of different examples. However, due to page limitations, we summarize explicit results only for a representative scenario, namely, for *Gene Expression Analysis* (GEA), which provides quantitative measures of the transcriptional behavior of a reporter gene under specific epigenetic conditions [8]. This scenario has been considered in [11] as well. We vary the GEA by performing

[3] Note that this formulation currently ignores the duration of the operation. The duration will be enforced by another formulation introduced later.

Table II: Resources used in the experiment

(a) General resources

Resource	Amount
Heater	3
Camera	1
Detector	2
Waste Reservoir	5
Magnet	1

(b) Dispensers

Dispenser	Amount
Wash	6
Sample	2
Buffer_1	2
Buffer_2	2
Beads	1
Elution	1
Primers	1
dNTP	1
CNT	1
NTC	1

Table III: Results for the Gene Expression Analysis.

			Method from [11]		Proposed method	
# Pathways	Assay size	Distance	RRS	URS	Steps	Time (s)
1	1	1	102	102	90	61
1	2	1	128	121	99	101
2	1	1	145	145	108	330
1	3	1	151	137	115	844
3	1	1	175	170	127	1432
2	2	1	183	169	124	1920
1	4	1	177	158	130	6129
1	1	2	n/a		115	208
1	2	2	n/a		125	320
2	1	2	n/a		131	1303
1	3	2	n/a		134	551
3	1	2	n/a		152	5550
2	2	2	n/a		150	6916
1	4	2	n/a		152	4892
1	1	3	n/a		140	548
1	2	3	n/a		151	728
2	1	3	n/a		156	3287
1	3	3	n/a		161	1233
3	1	3	n/a		174	16549
2	2	3	n/a		173	8449
1	4	3	n/a		170	2966
1	1	4	n/a		165	1181
1	2	4	n/a		177	1568
2	1	4	n/a		181	7331
1	3	4	n/a		189	3075

multiple experiments in parallel (# Pathways) and/or running the experiment with a different level of precision (Assay size). As architecture, a ring-based DMFB with resources as listed in Table II has been considered.

Table III summarizes the results obtained by the solution presented in [11] and the methodology proposed in this work. The columns *# Pathways* and *Assay size* denote the number of parallel experiments and the level of precision, respectively. The column *Distance* denotes the distance between the resources on the ring by counting electrodes. The most important values are provided in the columns denoted *Steps* which provide the number of time steps needed by the respectively obtained design in order to complete the assay. These numbers are provided for the designs obtained by the solution proposed in [11] – using two different settings, namely with *Restricted Resource Sharing* (RRS) and with *Unrestricted Resource Sharing* (URS) – as well as for the designs obtained by the proposed solution. In the latter case, we additionally provide the run-time required to generate the design in CPU seconds (see column *Time*)[4].

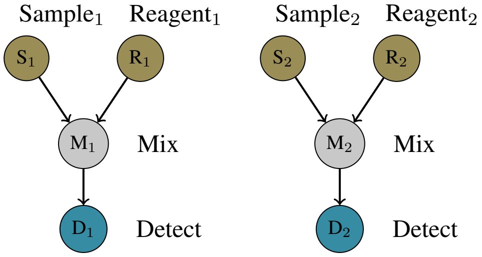


Fig. 5: Protocol used to demonstrate the influence of the degradation threshold.

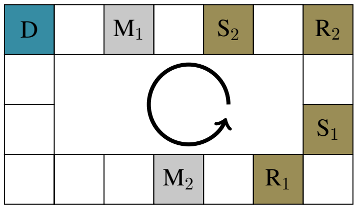


Fig. 6: Architecture used to demonstrate the influence of the degradation threshold.

As the method from [11] does not consider distances between resources, we compare both methods assuming a distance of 1 cell for both. This comes closest to the inherent restriction employed in [11]. Besides that, we additionally provide results obtained with larger distances in the bottom rows of Table III[5]. The results clearly show that the proposed method determines solutions with completion times that are significantly smaller. On average, just 76% and 80% of the steps required by the designs obtained with the previously proposed method in configuration RRS and URS, respectively, are needed. Moreover, the results obtained by the proposed method are guaranteed to be optimal, i.e., no faster realization will be possible under the applied constraints. The applicability of the proposed method shows that there is no need to resort to using heuristic approaches.

B. Effect of Technology Constraints

Using the proposed methodology, technology constraints such as degradation can be enforced during the design of DMFBs for the first time. The advantages of that have been evaluated in a second series of evaluations. Again, a representative is discussed in the following.

More precisely, consider the protocol shown in Fig. 5 to be executed on a ring-based DMFB as sketched in Fig. 6. Furthermore, consider the case where mixers can be applied an arbitrary number of times (no consideration of technology constraints) and where they can be applied only once (i.e. considering a technology constraint). Table IV sketches the obtained results.

When no technology constraint is considered (cf. Table IV(a)), Mixer_1 is simply used twice. This avoids the long run to the second mixer and, hence, yields a significantly faster completion time (namely 16 time steps only). In contrast, when the technology constraint that mixers are to be used only once is enforced (cf. Table IV(b)) droplets may have to take the longer route to Mixer_2. Overall, this leads to a significantly larger completion time of 23 time steps[6].

Overall, studies like that can be conducted using the proposed approach for the first time. In contrast, the state-of-the-art solution from [11] does not consider technology constraints at all, i.e., realizations generated using this approach are more prone to faults or defects. This is avoided with the solution proposed in this work.

[4] These experiments have been conducted on an Linux machine with 3.5 GHz of processor speed and 32 GB of main memory.

[5] Note that distances have not been considered in [11], and, therefore, these rows are marked "n/a". Considering distances yields much more realistic designs as resources are usually not right next to each other in an DMFB.

[6] Recall that both results are optimal under the given constraints.

Table IV: (a) Solution for the protocol using 16 time steps. Mixer$_1$ is used twice. (b) Solution for the protocol with harsh degradation constraints using 23 time steps. Mixer$_1$ can be used only once.

(a) No threshold

Resource	Usage (steps)
Mixer$_1$	5–8; 9–12
Detector	11–13; 14–16
Mixer$_2$	–
Sample$_1$	2–3
Reagent$_1$	3–4
Sample$_2$	1–2
Reagent$_2$	1–2

(b) Use mixer once

Resource	Usage (steps)
Mixer$_1$	11–14
Detector	18–20; 21–23
Mixer$_2$	11–14
Sample$_1$	3–4
Reagent$_1$	6–7
Sample$_2$	1–2
Reagent$_2$	1–2

Table V: Results for the parallel GEA experiments.

Benchmark	Time Steps
short_2_3	208
short_1_3	223
short_1_2	198
medium_2_3	273
medium_1_3	239
medium_1_2	273
long_2_3	343
long_1_3	484
long_1_2	347

C. *Completion-Time Profile for Two-Pathway Reactions*

Next, we aim to show the significance of the exact method in the context of flash chemistry. Recall that biochemical pathways within a flash chemistry environment require fine-grained control and minimization of protocol completion time, even with increasing the pathway length. For this purpose, there is a need to investigate the completion-time profile of reaction pathways when the reaction complexity is increased.

To perform this study, we use the GEA benchmark to formulate a two-pathway reaction. We gradually increase the length of pathways to investigate the growth of protocol time obtained by the exact method. We consider three different cases in terms of pathway length (number of bioassays): (1) a pathway with 6 bioassays; (2) a pathway with 8 bioassays; (3) a pathway with 10 bioassays. Table V lists some of the obtained results.

These results are invaluable to accurately characterize the steps in flash chemistry (while heuristic methods are not useful in this context since they ignore realistic constraints and are they not guaranteed to minimize the reaction time). Our results show that the proposed exact method is feasible for these realistic test cases. The CPU time to generate each of these solutions was in the range of 35 hours; this run time is not a concern because it is a one-time cost and carried out well in advance of the launch of the on-chip experiment. Heuristic methods, which offer reduced CPU time but are associated with longer protocol-completion time, are not necessary because the exact method can be used in practice for these protocols.

VI. Conclusion

In this work, we proposed a methodology supporting complex biomolecular protocols on DMFBs that respects technology constraints while, at the same time, does not make use of an oversimplified model of the DMFB. Experimental results confirm the applicability of the approach. Using the proposed exact methodology, we were able to produce protocol completion times that are at approximately 70% of those of previous work, showing that there still is room for improvement for heuristic approaches or even no need to resort to using heuristics at all. This is noteworthy as in previous work no distances were considered, i.e., one would expected even shorter solutions. We further were able to prove that considering technology constraints is necessary as it can have a significant influence on the protocol completion time.

References

[1] R. B. Fair, "Digital microfluidics: is a true lab-on-a-chip possible?" *Microfluidics and Nanofluidics*, vol. 3, no. 3, pp. 245–281, 2007.

[2] T. Xu, K. Chakrabarty, and V. K. Pamula, "Defect-tolerant design and optimization of a digital microfluidic biochip for protein crystallization," *Trans. on CAD*, vol. 29, no. 4, pp. 552–565, 2010.

[3] M. Pollack, A. Shenderov, and R. Fair, "Electrowetting-based actuation of droplets for integrated microfluidics," *Lab on a Chip*, vol. 2, no. 2, pp. 96–101, 2002.

[4] A. Rival *et al.*, "An EWOD-based microfluidic chip for single-cell isolation, mRNA purification and subsequent multiplex qPCR," *Lab on a Chip*, vol. 14, no. 19, pp. 3739–3749, 2014.

[5] Z. Hua *et al.*, "Multiplexed real-time polymerase chain reaction on a digital microfluidic platform," *Analytical chemistry*, vol. 82, no. 6, pp. 2310–2316, 2010.

[6] L. R. Volpatti and A. K. Yetisen, "Commercialization of microfluidic devices," *Trends in biotechnology*, vol. 32, no. 7, pp. 347–350, 2014.

[7] S. Miller, K. Henthorn, D. Osato, J. Lips, A. Schroeder, R. Humphries, and L. Freeman-cook, "Multiplex detection of respiratory pathogens with genmark's eplex sample-to-answer system," *Clinical Chemistry and Laboratory Medicine*, vol. 54, no. 5, pp. eA5–eA6, 2016.

[8] B. S. Wheeler, B. T. Ruderman, H. F. Willard, and K. C. Scott, "Uncoupling of genomic and epigenetic signals in the maintenance and inheritance of heterochromatin domains in fission yeast," *Genetics*, vol. 190, no. 2, pp. 549–557, 2012. [Online]. Available: http://www.genetics.org/content/190/2/549

[9] J.-i. Yoshida, A. Nagaki, and T. Yamada, "Flash chemistry: fast chemical synthesis by using microreactors," *Chemistry–A European Journal*, vol. 14, no. 25, pp. 7450–7459, 2008.

[10] J.-i. Yoshida, Y. Takahashi, and A. Nagaki, "Flash chemistry: flow chemistry that cannot be done in batch," *Chemical Communications*, vol. 49, no. 85, pp. 9896–9904, 2013.

[11] M. Ibrahim, K. Chakrabarty, and K. Scott, "Integrated and real-time quantitative analysis using cyberphysical digital-microfluidic biochips," in *DATE*, 2016, pp. 630–635.

[12] "Illumina," https://www.illumina.com/systems/neoprep-library-system.html.

[13] F. Su and K. Chakrabarty, "High-level synthesis of digital microfluidic biochips," *JETC*, vol. 3, no. 4, p. 1, 2008.

[14] E. Maftei, P. Pop, and J. Madsen, "Module-based synthesis of digital microfluidic biochips with droplet-aware operation execution," *JETC*, vol. 9, no. 1, p. 2, 2013.

[15] O. Keszocze, R. Wille, and R. Drechsler, "Exact routing for digital microfluidic biochips with temporary blockages," in *ICCAD*, 2014, pp. 405–410.

[16] J.-W. Chang, S.-H. Yeh, T.-W. Huang, and T.-Y. Ho, "Integrated fluidic-chip co-design methodology for digital microfluidic biochips," *Trans. on CAD*, vol. 32, no. 2, pp. 216–227, 2013.

[17] O. Keszocze, R. Wille, T.-Y. Ho, and R. Drechsler, "Exact One-pass Synthesis of Digital Microfluidic Biochips," in *DAC*, ser. DAC, 2014, pp. 142:1–142:6.

[18] O. Keszocze, R. Wille, K. Chakrabarty, and R. Drechsler, "A General and Exact Routing Methodology for Digital Microfluidic Biochips," ser. ICCAD, 2015, pp. 874–881.

[19] Y. Luo, K. Chakrabarty, and T.-Y. Ho, "Error recovery in cyberphysical digital microfluidic biochips," *Trans. on CAD*, vol. 32, no. 1, pp. 59–72, 2013.

[20] K. Hu, M. Ibrahim, L. Chen, Z. Li, K. Chakrabarty, and R. Fair, "Experimental demonstration of error recovery in an integrated cyberphysical digital-microfluidic platform," in *BIOCAS*, 2015, pp. 1–4.

[21] K. Hu, B.-N. Hsu, A. Madison, K. Chakrabarty, and R. Fair, "Fault detection, real-time error recovery, and experimental demonstration for digital microfluidic biochips," in *DATE*, 2013, pp. 559–564.

[22] D. T. Grissom and P. Brisk, "Fast online synthesis of digital microfluidic biochips," *Trans. on CAD*, vol. 33, no. 3, pp. 356–369, 2014.

[23] "Genmark," https://www.genmarkdx.com/solutions/systems/eplex-system/.

[24] E. Samiei, M. Tabrizian, and M. Hoorfar, "A review of digital microfluidics as portable platforms for lab-on a-chip applications," *Lab on a Chip*, vol. 16, no. 13, pp. 2376–2396, 2016.

[25] H. Norian *et al.*, "An integrated CMOS quantitative-polymerase-chain-reaction lab-on-chip for point-of-care diagnostics," *Lab on a Chip*, vol. 14, no. 20, pp. 4076–4084, 2014.

[26] M. Mibus, X. Hu, C. Knospe, M. L. Reed, and G. Zangari, "Failure modes during low voltage electrowetting," *ACS Applied Materials & Interfaces*, vol. 8, no. 24, pp. 15 767–15 777, 2016.

[27] L. De Moura and N. Bjørner, "Z3: An efficient SMT solver." Springer, 2008, pp. 337–340, Z3 is available at http://z3.codeplex.com/.