Temporal Planning for Droplet Routing on Microfluidic Biochips - Master Thesis Preparation

Fabian Burch
University of Basel
f.burch@unibas.ch

May 11, 2022

1 Introduction

Advances in microfluidic technologies have enabled the production of Digital Microfluidic Biochips (DMFBs), a fully electronically controllable lab on a chip. A DMFB essentially is a 2D-grid of electrodes on which small droplets of biochemical assets can be moved, mixed and measured, as can be seen in Figure 1. Whole biochemical experiments can be performed on such chips without the need for human interaction. This technology provides various advantages over classical laboratory experimentation, such as miniaturization, automation, repeatability, pruning human errors as well as being controllable by software. DMFBs are based on the effect of electrowetting [1]. By applying a voltage to an electrode, the contact angle of a droplet close to the electrode can be affected, effectively drawing the droplet onto the electrode [2]. This allows completely digital control of liquid droplets with the help of a grid of electrodes. A variety of biochemical operations can be performed just by using this phenomenon. They include, but are not limited to, movement, mixing, splitting, storage and detection. But such a powerful platform also comes with many challenges. In the following section we will discuss the issues that come with the design of DMFBs and what has been done so far in trying to solve those issues.

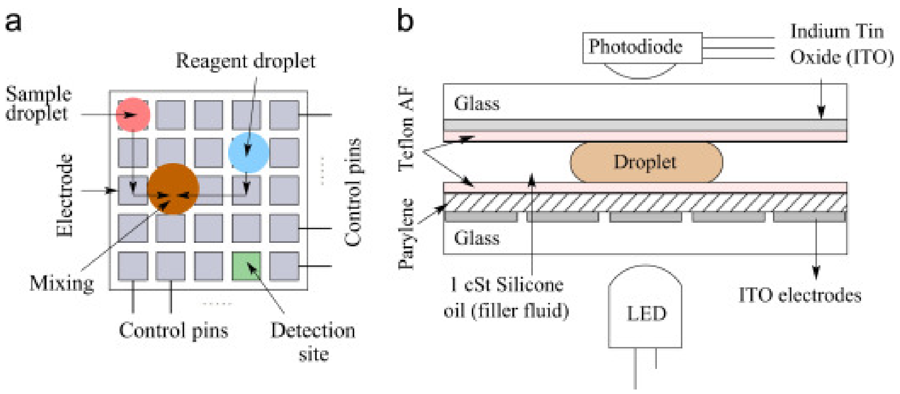
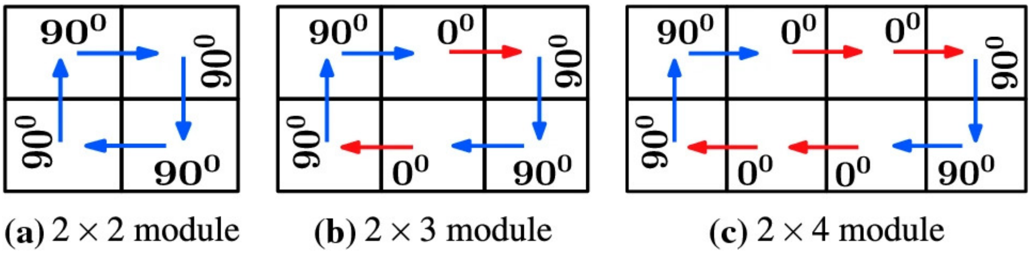


Figure 1: a) Schema of a simple DMFB. A 5-by-5 grid of electrodes is used to move two droplets to the same location, mixing them. They then can be moved to a detection site. b) Side view of a DMFB. Droplets are enclosed between two plates in a filler fluid (usually silicone oil) [3]. Open chip designs are also possible and cheaper to achieve. Picture taken from ”Waste-aware single-target dilution of a biochemical fluid using digital microfluidic biochips” by Sudip Roy et al. [4]

2 Previous Work

When designing a DMFB experiment, one has to take into account various factors. Let's assume we want to mix two droplets. This can for example be achieved by a 2x2 mixing field in which the assets to be mixed are moved in circular fashion [5]. Usually a guard ring is applied around a mixer, increasing the occupied area to a 4x4 square for a 2x2 mixer. Three different sizes of mixing modules can be seen in Figure 2. Mixers, detectors, dilutors and storage units are examples for DMFB modules. Modules of all sizes are stored, alongside any accompanying information such as size and execution time, in a module library [6]. All the biochemical operations of an experiment have to be mapped to the DMFB modules. This is called binding. After finding appropriate modules for all operations, a schedule for the order of the operations has to be made and the modules have to be allocated to electrodes on the chip. Due to the dynamic nature of DMFBs, the modules are not to be seen as static but as dynamic objects that can be allocated to certain electrodes for the duration of the operation. If we return to our mixer example we would allocate a 4x4 area on the chip for the mixing operation. After completing the mixing process, the 4x4 space can be used for a completely different operation with



(a) 2 × 2 module (b) 2 × 3 module (c) 2 × 4 module

Figure 2: Differently sized mixers. Generally speaking, bigger mixers will mix droplets faster than smaller ones. Picture taken from ”An efficient module-less synthesis approach for Digital Microfluidic Biochip” by Sarit Chakraborty & Susanta Chakraborty [5]

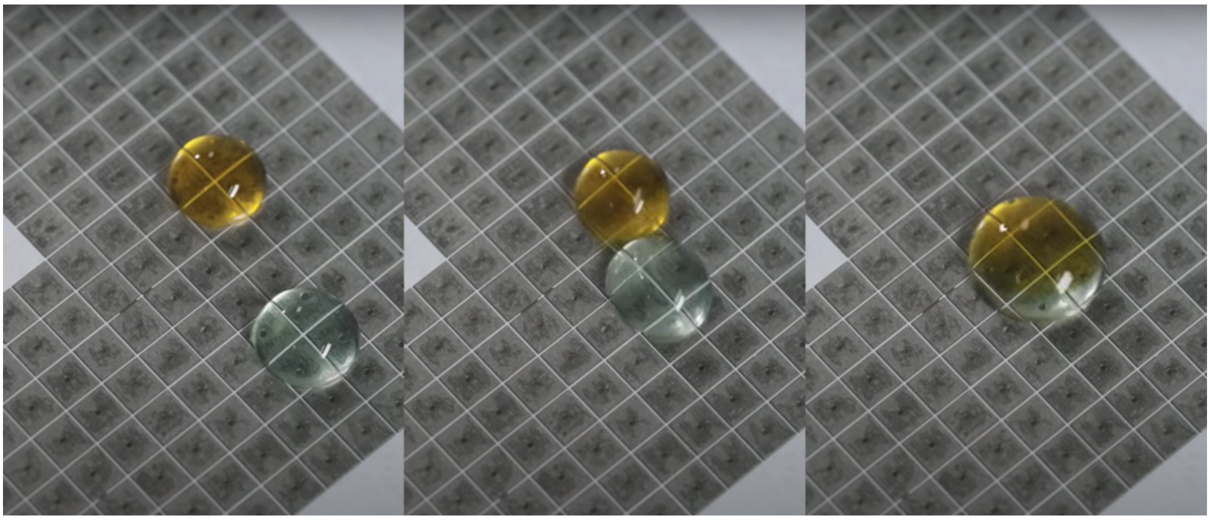


Figure 3: Two droplets that get too close and merge. The improper mixing can be seen as a color gradient on the big droplet. Pictures are screenshots taken from `https://youtu.be/zONBsyhApvU`.

different droplets. Droplets have to be moved to their target destinations between operations. A freshly mixed droplet may need to be moved to a detection site where its color will be measured. Droplets that are not meant to merge have to obey fluidic constraint rules in order to avoid unwanted mixing [7] during moving on the chip, as can be seen in Figure 3. For example if a droplet needs to be moved to an adjacent electrode, but this electrode is also adjacent to another droplet, both droplets will be drawn to the same electrode.

The problem of droplet routing has been covered by several papers with different approaches. Inspired by VLSI[1] design, several papers have used a modified version of the Lee algorithm [8] for finding shortest paths between two points [7, 9, 10, 11]. Keszöcze et al. have described droplet routing as a Boolean Satisfyability (SAT) problem and used the Z3 [12] SAT solver to find minimal paths [13]. A similar approach was used by Böhringer [14]. He describes the routing problem as a graph search problem and uses an A* search to obtain solutions.

Another topic that has been addressed by researchers is electrode control. Each electrode has to be connected to a control pin, that delivers the signals from the control computer. Pin assignment maps each electrode on the chip to a control pin. Multiple electrodes can be assigned to the same pin. Minimizing the number of control pins makes the chip simpler and cheaper. The pin assignment done by Keszöcze in 2015 [15] is based on a symbolic formulation of the problem. This very general approach allows for a variety of solving engines to be used. The authors also use the previously mentioned Z3 engine to test their results.

One possible way to control the electrodes is to connect the DMFB to a PCB (Printed Circuit Board). A PCB that can control every single electrode on the biochip is possible,

[1] Very-Large-Scale Integrated Circuits

but expensive. PCBs can be optimized for specific setups though. By doing a minimal pin assignment, less wires are needed to connect the control pins to the electrodes. In an effort to minimize the number of layers of a PCB and therefore also the production cost, McDaniel et al. proposed another modified Lee algorithm [11]. With a pin assignment as input, their algorithm tries to find a single-layer PCB wire routing. Whenever it cannot find a solution, it increases the number of layers by one until it finds a valid design. The process of finding a suitable wiring for a given pin assignment is called escape routing.

It is apparent that there have been various approaches to try and solve the challenges of cost efficient and convenient DMFBs. From VLSI approaches to SAT solvers, several algorithms have been proven effective. We want to take a new approach that has previously been used in quantum circuit design.

Quantum computers use quantum effects to solve problems conventional computers struggle with. They use quantum bits, short qubits, have a set of interesting properties. Unlike classical bits, qubits are not just either a 0 or a 1, but a superposition of the basis states $|0\rangle$ and $|1\rangle$. A qubit is in a quantum state $|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ where $\alpha, \beta \in \mathbb{C}$ denote probability amplitudes. When a qubit is measured, meaning a measurement device on the chip looks at the state of qubit, its quantum state collapses and either $|0\rangle$ or $|1\rangle$ is returned. The probability that a $|0\rangle$ is returned is $|\alpha|^2$, whereas the probability for $|1\rangle$ to be returned is $|\beta|^2$. States like $|\Psi\rangle$ are called pure or coherent states [16]. However, as soon as we use real-life representations of qubits, we diverge from this idealized model. For example, IBMs 5-qubit quantum chip is based on superconducting qubits [17]. Those superconducting qubits, as well as any other qubit representations that are currently available, are not perfectly isolated from their surroundings and will, over time, exchange information with their surroundings. This will make the qubits no longer pure. This loss of information in qubits is called decoherence [16]. The longer a qubit state is unmeasured, the less likely it is that a measurement returns the $|0\rangle$ or $|1\rangle$ state with their correct probabilities $|\alpha|^2$ or $|\beta|^2$. Therefore it is desirable to minimize the execution time of a quantum circuit.

Gate-model quantum computing chips, like the 5-qubit quantum chip by IBM, are an up-and-coming technology that is scalable and therefore able to run any quantum algorithm [18]. A quantum algorithm is performed by a quantum circuit, which is a sequence of applying quantum gates or measurements on qubits [19]. Quantum gates are the building blocks that quantum circuits are made of. Also called quantum logic gates, they are similar to classical logic gates, with the main difference being that quantum logic gates are reversible. They can be applied to single or multiple qubits, altering their states, just like logic gates. A quantum chip basically is a set of gates that can be applied to the qubits present on the chip. In Figure 4 a simple quantum chip design is shown.

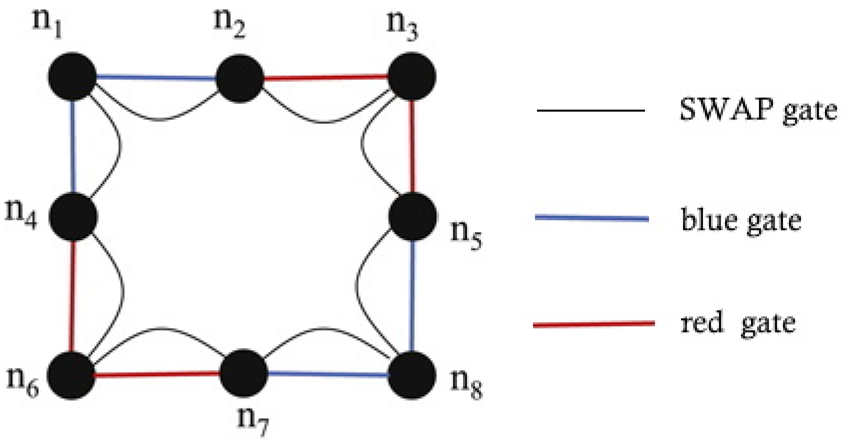


Figure 4: An abstract representation of a quantum chip. The black dots represent positions that qubits can be in (labelled n_1 through n_8). The lines connecting those positions represent gates, that are applied on the qubits in these positions. Picture taken from "Compiling quantum circuits to realistic hardware architectures using temporal planners" by Davide Venturelli et al. [18].

The gates, that are shown as lines connecting two dots, will affect the qubits on the two positions they connect. The most important gate is the SWAP gate, which changes the position of the two qubits it connects. This allows for two specific qubits to be moved to the positions that are connected by another gate that then can be applied to them. Each qubit is connected to two neighbours by SWAP gates and either red or blue gates, which represent some specific two-qubits-gates. With this design it is possible to apply red or blue gates to any pair of qubits by executing a sequence of SWAP gates to move them into position.

Similar to a biochemical experiment that needs to be designed to be performed on a DMFB, a quantum circuit can be designed to perform a certain quantum algorithm on a quantum chip. This makes it interesting for us, as we can try to apply the methods used in quantum circuit design to droplet routing on DMFBs. There are parallels like SWAP gates that correspond to droplet movement and red or blue gates that correspond to modules like mixers or sensors. As mentioned before, quantum computing chips and circuits suffer from non-neglectable decoherence [18]. To counteract this, quantum circuits have to be designed to take minimal time to execute. Venturelli et al. have proposed temporal planning to find circuits with minimal execution time [18]. By describing hardware limitations, possible actions as well as start and goal state in PDDL (Planning Domain Definition Language), they were able to use any off-the-shelf temporal planner compatible with PDDL. For small circuits (less than 22 qubits) they were able to obtain good results in reasonable time. Electrode grids of the size 8x8 are a standard size for DMFBs [5]. Considering that even small mixers already take up a 4x4 square, the problem size is comparable and we should be able to obtain some results in reasonable time.

3 Questions and Problems to Handle in the Thesis

The goal of our thesis is to take the temporal planning approach, used by Venturelli et al. [18], and apply it to the droplet routing problem on DMFBs. Minimizing execution time has not been the focus of research so far. The main focus of our work is to find an implementation of the temporal planning approach for droplet routing on DMFBs, such that we can perform tests similar to the ones done by Su et al. [7] or by Xu and Chakrabarty [9]. If temporal planning proves to be too computationally intensive, we might want to take a classical planning approach, where we the different droplets take turns, to simulate a temporal approach. Alternatively we can use a multi-agent setup and treat the different droplets as independent agents. If we manage to get acceptable results, we can further extend the approach on other DMFB problems. It makes sense to combine scheduling, module placement and droplet routing, as they are the core of a biochemical experiment on a DMFB. Again similar to Su et al. and Xu and Chakrabarty, testing the algorithm on a simple, real-life reaction would lead to meaningful benchmarks. Note that Su et al. do not do scheduling and module placement themselves, but use given schedules and placements and only do the routing themselves. It remains to be seen if multi-agent temporal planning is able to efficiently create schedules for given modules (and possibly a given module placement). When creating actions in PDDL, the execution times can be taken from the module library. Ideally, temporal planning is able to find optimal schedules as well as efficient droplet routing.

Doing pin assignment alongside droplet routing as Keszöcze et al. did in their work [15] would surely be possible, but there is no obvious advantage in using temporal planning for pin assignment. Escape routing was treated with the same algorithms as droplet routing before. Temporal planning though seems not to be a suitable solution for escape routing, as there the goal is not to minimize a timespan, but wire length or PCB layers. A complete pipeline including every design step as attempted by Keszöscze et al. in 2014 [13] seems rather ambitious in the face of the higher computational cost of temporal planning.

If time allows for it, we can try to run previous approaches on the same machine and problems as our approach and compare the results. With that we could effectively rank the temporal approach among the previously applied methods.

3.1 Possible Scope

When we look at the possible scope of our work, we can subdivide the challenges of DMFBs into the following problems (we exclude pin assignment and escape routing, as we assume chips that are fully controllable and every electrode is individually addressable).

- binding (mapping the DMFB modules to their respective biochemical operations)

- scheduling (finding an efficient / optimal order of the operations that need to be executed)
- module placement (finding positions for the modules that allow for short routes between them)
- droplet routing (planning the movement and timing of all droplets on the chip)

What we absolutely want to do is droplet routing, as we see the most potential in temporal planning for this task. Then step-by-step more problems could be included. The next thing would be to take module placement into account. Next, scheduling could be included and then finally binding. All these things are coupled to a certain degree, so it would make sense to integrate them all into out algorithm, if possible.

4 Execution Schedule

WEEK	DATE	TASK
Week 1 (CW20)	16.05.2022	Start
Week 2 (CW21)	21.05.2022	"Introduction" Second Draft, Reading on Temporal Planning
Week 4 (CW23)	06.06.2022	"Related Work" Second Draft, First tests in PDDL completed
Week 6 (CW25)	20.06.2022	Keep "Methods" up-to-date, Test specific routing problem
Week 8/9	5.7. - 12.7.	Holidays
Week 11 (CW30)	25.07.2022	"Methods" Draft, First Results, Finish "Introduction" & "Related Work"
Week 16 (CW35)	29.08.2022	"Results" Draft, Start "Discussion" of results, consider further topics to look into
Week 18 (CW37)	12.09.2022	"Discussion" Draft, Finish "Methods"
Week 22 (CW41)	10.10.2022	No more drafts, final revisions
Week 23 (CW42)	24.10.2022	Writing finished
Week 27 (CW46)	14.11.2022	End

References

[1] Frieder Mugele and Jean-Christophe Baret. “Electrowetting: from basics to applications”. *Journal of physics: condensed matter*, Vol. 17, No. 28 (2005).

[2] Michael G. Pollack, Alexander D. Shenderov, and Richard B. Fair. “Electrowetting-based actuation of droplets for integrated microfluidics”. *Lab on a Chip*, Vol. 2, No. 2 (2002), pp. 96–101.

[3] Richard Fair, Vijay Srinivasan, H. Ren, Philip Paik, Vamsee K. Pamula, and Michael G. Pollack. “Electrowetting-based on-chip sample processing for integrated microfluidics”. In: *IEEE International Electron Devices Meeting 2003.* IEEE. (2003), pp. 32.5.1–32.5.4.

[4] 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*, Vol. 51 (2015), pp. 194–207.

[5] Sarit Chakraborty and Susanta Chakraborty. “An efficient module-less synthesis approach for Digital Microfluidic Biochip”. *SN applied sciences*, Vol. 2, No. 8 (2020).

[6] Wen-Chun Chung, Pei-Yi Cheng, Zipeng Li, and Tsung-Yi Ho. “Module Placement under Completion-Time Uncertainty in Micro-Electrode-Dot-Array Digital Microfluidic Biochips”. *IEEE Transactions on Multi-Scale Computing Systems*, Vol. 4, No. 4 (2018), pp. 811–821.

[7] Fei Su, William Hwang, and Krishnendu Chakrabarty. “Droplet routing in the synthesis of digital microfluidic biochips”. In: *Proceedings of the Design Automation & Test in Europe Conference.* Vol. 1. IEEE. (2006), pp. 323–328.

[8] Sadiq M. Sait and Habib Youssef. *VLSI Physical Design Automation - Theory and Practice.* IEEE, 1995.

[9] Tao Xu and Krishnendu Chakrabarty. “Integrated droplet routing in the synthesis of microfluidic biochips”. In: *Proceedings of the 44th annual Design Automation Conference.* (2007), pp. 948–953.

[10] Qin Wang, Zeyan Li, Haena Cheong, Oh-Sun Kwon, Hailong Yao, Tsung-Yi Ho, Kwanwoo Shin, Bing Li, Ulf Schlichtmann, and Yici Cai. “Control-fluidic codesign for paper-based digital microfluidic biochips”. In: *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD).* IEEE. (2016).

[11] Jeffrey McDaniel, Zachary Zimmerman, Daniel Grissom, and Philip Brisk. “PCB escape routing and layer minimization for digital microfluidic biochips”. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 36, No. 1 (2016), pp. 69–82.

[12] Nikolaj Bjørner, Leonardo de Moura, Lev Nachmanson, and Christoph Wintersteiger. *Z3prover/Z3: The Z3 theorem prover.* `https://github.com/Z3Prover/z3`. Accessed: 12.04.2022.

[13] Oliver Keszöcze, Robert Wille, Tsung-Yi Ho, and Rolf Drechsler. "Exact one-pass synthesis of digital microfluidic biochips". In: *Proceedings of the 51st Annual Design Automation Conference.* (2014).

[14] Karl F. Böhringer. "Towards optimal strategies for moving droplets in digital microfluidic systems". In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004.* Vol. 2. IEEE. (2004), pp. 1468–1474.

[15] Oliver Keszöcze, Robert Wille, Krishnendu Chakrabarty, and Rolf Drechsler. "A general and exact routing methodology for digital microfluidic biochips". In: *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD).* IEEE. (2015), pp. 874–881.

[16] Lu-Ming Duan and Guang-Can Guo. "Preserving coherence in quantum computation by pairing quantum bits". *Physical Review Letters*, Vol. 79, No. 10 (1997), p. 1953.

[17] Jerry Chow and Abhinav Kandala. *IBM's quantum computers: an optimal platform for condensed matter physics research.* `https://research.ibm.com/blog/condensed-matter-physics`. Accessed: 09.05.2022.

[18] Davide Venturelli, Minh Do, Eleanor Rieffel, and Jeremy Frank. "Compiling quantum circuits to realistic hardware architectures using temporal planners". *Quantum Science and Technology*, Vol. 3, No. 2 (2018).

[19] Michele Mosca. "Quantum algorithms". *Computational Complexity: Theory, Techniques, and Applications* (2008).