

# Effects of Cell Shapes on the Routability of Digital Microfluidic Biochips

Leonard Schneider<sup>1</sup> Oliver Keszocze<sup>1,2</sup> Jannis Stoppe<sup>1,2</sup> Rolf Drechsler<sup>1,2</sup>

<sup>1</sup>Institute of Computer Science, University of Bremen, 28359 Bremen, Germany

<sup>2</sup>Cyber Physical Systems, DFKI GmbH, 28359 Bremen, Germany  
{leosc,keszocze,jstoppe,drechsle}@informatik.uni-bremen.de

**Abstract**—Digital Microfluidic Biochips (DMFBs) are an emerging technology promising a high degree of automation in laboratory procedures by means of manipulating small discretized amounts of fluids. A crucial part in conducting experiments on biochips is the routing of discretized droplets. While doing so, droplets must not enter each others' interference region to avoid unintended mixing. This leads to cells in the proximity of the droplet being impassable for others. For different cell shapes, the effect of these temporary blockages varies as the adjacency of cells changes with their shapes. Yet, no evaluation with respect to routability in relation to cell shapes has been conducted so far. This paper analyses and compares various tessellations for the field of cells. Routing benchmarks are mapped to these and the results are compared in order to determine if and how cell shapes affect the performance of DMFBs, showing that certain cell shapes are superior to others.

## I. INTRODUCTION

Beyond improving their computational power, creating alternative applications for integrated circuits has garnered significant attention in research over the past years [2]. One key area of development has been the creation of Digital Microfluidic Biochips (DMFBs). DMFBs are an emerging technology that promises to miniaturize and enhance automation for laboratory procedures in molecular biology and biochemistry. In this regard, they are often referred to as *labs-on-a-chip*. They reduce the lab space, sample volume and time required for many laboratory procedures, including DNA sequencing, protein crystallization, drug discovery, immunoassays and neonatal diagnostics [2].

Relying on a high degree of run-time flexibility, automated design and ease-of-use of such systems necessitates biochip-specific tools for Computer-Aided Design (CAD) [11]. Consequently, the design of such tools will pose a challenge in the development and integration of DMFBs.

DMFBs work by manipulating fluid sample carriers (called *droplets*) containing biological and chemical samples on a micro- to nano-liter scale. In an assay, the droplets are placed on the chip and moved between modules on the biochip in order to perform various microfluidic operations. The procedure determining this movement is referred to as *Routing*. Routing is a crucial stage in the design of a DMFB that determines the transportation of droplets on the chip, taking into account the microfluidic properties and dynamics inherent in the problem (see e.g. [11]). The number of necessary steps to solve such a routing problem is an important factor in regards to production and maintenance cost of such a chip, necessitating *minimality* of a routing. It is to suspect that the ratio of fields in the interference region, e.g. the constraining region around a droplet, and the number of reachable fields for such a droplet have a prominent impact on routing performance.

We utilize an *exact* routing methodology to find minimal solutions to given routing problems on DMFBs for different cell shapes, allowing an assessment of the performance of routing on square grids to alternative shapes.

## II. MOTIVATION & BACKGROUND

### A. Design of Digital Microfluidic Biochips

The design of a biochip is usually carried out over several steps in a process known as *DMFB Synthesis*. In this synthesis, the steps are generally conducted in the order of *Allocation*, *Binding*, *Scheduling*, *Placement*, *Routing*, *Pin Mapping* and *Wire Routing*.

An assay is usually represented as a Directed Acyclic Graph (DAGs) [5] serving as a *sequencing graph* in which the desired set of operations to be executed is described. Operations are specified in a *module library*, which contains the necessary definitions for the realization of an operation, like detection or mixing, as well as their time and space requirements on the chip [6].

During *Allocation* and *Binding*, each operation specified in the sequencing graph is mapped to its respective realization defined in the module library. The *Scheduler* then determines the order of operations and assigns a start and a stop time to each operation in the graph.

During *Placement*, each operation is given a position on the chip; due to the dynamic reconfigurability of DMFBs, many operations, such as *Routing*, *Mixing* or *Splitting*, can be realized on any array of cells.

After the modules have been placed onto the chip, *Routing* determines a path between operations, e.g. from a dispensing port to a mixer, between modules, or from modules to an output reservoir. In this, the routing has to ensure that the droplets reach their destinations on the chip and do not interfere with other droplets being moved or performing an operation on the grid.

Finally, *Pin Mapping* determines the order of activation of the electrodes on the grid and *Wire Routing* is used to compute the necessary wiring.

### B. Routing on DMFBs

The routing step is a crucial step in the design of a biochip. Firstly, the routing has to ensure that each droplet reaches its target destination on the chip while adhering to the fluidic constraints imposed on it. Secondly, it has to take into account the different time requirements necessary for each operation in the assay.

Microfluidics pose a unique problem for routing on fixed grids. Due to the fluidic qualities inherent in the elements to be routed, some constraints have to be observed when determining a route between two points:

- 1) Two droplets must not occupy the same cell at the same time.
- 2) Droplets must also not occupy a cell adjacent to another cell with a droplet on it at the same time.
- 3) Droplets can move to reachable fields, but are movement-constrained by interference regions around other droplets.
- 4) Blocked cells on the grid may not be traversed.

### C. Cell Shapes

The conventional and most commonly used layout in which electrode grids are arranged are as a grid of square cells [9]. On a square electrode, the droplet sits at the center of the electrode; a certain portion of this droplet overlaps the sides of the electrode.

Beyond this design, some alternative layouts have also been suggested. Hexagonal fields have already been proposed as early as in [9], and are now indirectly used in Micro-Electrode Dot-Array (MEDA) biochips. In e.g. [7], [12], the square shaped fields are shifted every second line, effectively allowing for movement and an interference region as with hexagonal fields, while triangular or castellated designs are explored in [4].

There are some promising properties of alternative area division methods that could be exploited for DMFB design, e.g. the “zigzag” path a droplet would for example perform during a mixing operation on a triangular grid [3]. However, there has been no investigation into the possible implications on the routing performance of DMFBs with alternative cell shapes (although [9] notes that a hexagonal design is expected to increase the effectiveness of droplet transportation).

We define the **grid** as a *tessellation*, i.e. a regular, two-dimensional arrangement of these cell shapes in such a way that there is no empty space between adjacent cells and that no cells overlap on the plane.

On two-dimensional grids, only three-, four- and six-sided regular polygons can provide repeatable, uniform area division usable for droplet routing. As such, Triangles, Squares (Quadrilaterals) and Hexagons are the only dissimilar area division methods that, arranged on a plane, offer a close-packed design; geometric shapes with less sides cover no area, while regular pentagons and those with more sides cannot be arranged in such a manner that there is no empty space between them [1]. Consequently, we will focus on three distinct designs for cells on DMFBs and the grids they form: *Hexagons*, *Equilateral Triangles* and *Rectangular Triangles*.

## III. METHODOLOGY

To evaluate whether or not a certain tessellation is affecting the performance of DMFBs, the amount of time (in steps) required for droplets to be moved across the chip is used as a measurement. Existing routing problems for square grids are translated into different tessellations and the optimal routing solution is computed in order to determine how the cell shape affects the runtime of the chip.

### A. Field Shapes & Neighbourhoods

Each cell at the coordinates  $(x, y)$  on the grid has a unique set  $\in N_x$  of neighbouring, adjacent cells. On square grids, most cells have four immediately neighbouring cells. Cells at the edge of the grid have three neighbours, while those at the corners have two. This neighbourhood relation contains all immediately accessible cells from the “home” cell and thus constitutes the movable region of the cell. Fig. 1 illustrates such a direct neighbourhood relation for square cells.

Since the DMFB handles fluidics, it has to be ensured that droplets on adjacent cells on the grid do not mix, which can occur when two droplets come in contact. To avoid this, a second neighbourhood relation is introduced which contains *all* surrounding cells at that position — horizontally, vertically and diagonally. In a stricter sense, this is the set of all points of contact of the boundaries of the cell to other cells, which form the interference region of the cell.

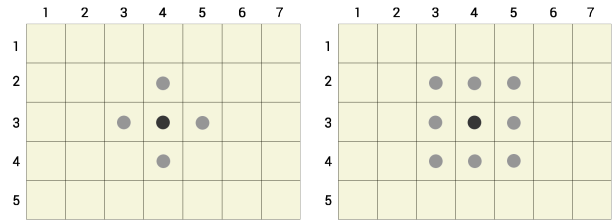


Fig. 1: The movable region (left) / interference region (right) of the square cell at 4,3

Unlike square and triangular grids, which have both direct and indirect neighbours, all neighbours of a hexagonal cell are directly adjacent. Thus, movable and movement-constrained regions can be defined in the same neighbourhood relation  $N_6(x, y)$ . This leads to a ratio of 6 : 6 in regards to the movable and interference regions on the chip.

For rectangular triangles, a method of arranging triangles in an alternating fashion is used, enabling a linear arrangement along the y-axis of the grid. Equilateral triangle tilings were designed to form a linear arrangement on the x-axis of the grid. Since the overlapping droplet area on equilateral triangles is larger than that on other shapes, some researchers argue that the direct *Three-Neighbourhood* would possibly suffice as movement-constraining to prevent unwanted droplet merging [3]. Nevertheless, since a routing leading to unwanted mixing of droplets would result in a faulty assay, a neighbourhood relation that includes direct and indirect neighbours is employed, resulting in a movable-to-interference region ratio of 3 : 12.

### B. Transformation of Benchmarks

The grid configurations against which these alternative area division methods are tested are based on square electrodes. In order to use these models as basis for a comparison of routing performance, a suitable conversion from the base square model to models for triangular and hexagonal grids has to be defined.

To transform a square grid to a hexagonal grid, hexagonal cells are arranged along the dimensions specified by the square grid. This ensures that the number of cells remains the same in both grids. *Spawn* and *target* positions for droplets remain at the same coordinates. Fig. 3 illustrates how the example square configuration from Fig. 2 is translated to a hexagonal setup.

In the sense that triangles can be regarded as “halved” squares, two methods of grid conversion are employed for them. The first method aims for a one-to-one conversion of the grid. This would ensure that the grid has the same dimensions, with droplets and blockages at the same coordinates as in the square grid, allowing for a direct comparison of the routing performance on triangles for similar routing problems as those on squares. Examples of this approach are illustrated in Figure 4. However, such mappings have the problem that they may render benchmarks unsolvable for the resulting triangles: Benchmarks that e.g. rely on straight “lines” of squares to route their droplets along may be transformed into benchmarks with lines of triangles that are only touching on their vertices, thus blocking previously available paths across the chip.

Thus, a second method to *transform* the grid to triangles is introduced. By multiplying the amount of cells along one axis of the grid by two, these potential errors are avoided. In the transformed model, the x-axis is used for this purpose. In this way, a “square” grid may be built to test the routing performance of triangular movement on it. These transformed triangle setups are illustrated in Figure 5.

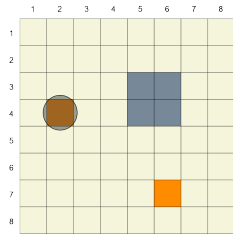


Fig. 2: A sample configuration for a  $8 \times 8$  grid.

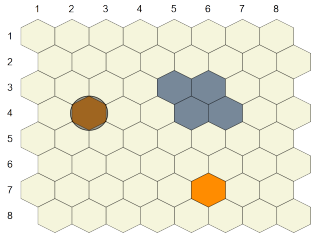


Fig. 3: The same configuration as in the example for squares (see Fig. 2) for a grid of hexagons.

### C. Routing Algorithm

Microfluidics pose a unique problem for routing on fixed grids. Due to the fluidic qualities inherent in the elements to be routed, some constraints have to be observed when determining a route between two points:

- 1) Timing requirements have to be observed.
- 2) Two droplets must not occupy the same cell at the same time.
- 3) Droplets must also not occupy a cell adjacent to another cell with a droplet on it at the same time.
- 4) Droplets may only move via directly neighboring cells, but are movement-constrained by direct and indirect neighbours.
- 5) Blocked cells on the grid may not be traversed.

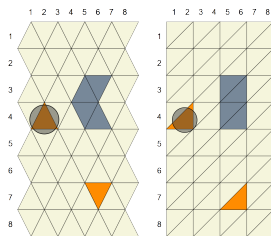


Fig. 4: A grid of equilateral triangles (left) / rectangular triangles (right) in the same configuration as in Fig. 2.

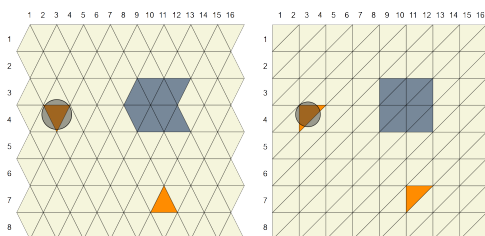


Fig. 5: The transformed grid of equilateral triangles (left) / rectangular triangles (right), based on the example in Fig. 2 with two triangles per original square.

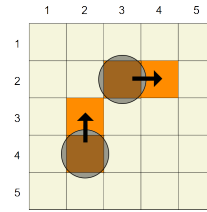


Fig. 6: Unwanted mixing

While 2), 3) and 4) allow the crossing of droplet paths as long as they cross at different time steps and disallow direct merging, they are often not sufficient for preventing all possibilities of unwanted droplet merging. Some cases for constraining fluidic movement have to be observed especially: The arrangement in Fig. 6 presents a valid configuration of droplets on a  $5 \times 5$  grid. Whereas simultaneous movement of the two droplets is usually desired, in this configuration, the droplets can inadvertently merge if a droplet B moves into the neighbourhood of the cell droplet A had *previously* occupied. To prevent these cases, usually a *conservative* routing method is used for preventing all possibilities of droplet merging (i.e. here: enforcing neighbourhood separation before and after a droplet movement), which is also utilized in this work.

In order to ensure the comparability of the results, a methodology for finding exact solutions for routing over given routing problems is utilized as proposed in [6]. It addresses the exponentially hard complexity of determining a minimal solution by utilizing the solving power of expanded SAT solving engines. To encode these routing problems as decision problems for the solving engine, an expansion of SAT known as SAT Modulo Theories (SMT) [8] is utilized.

In order to determine a routing with a minimal number of time steps using SMT, the optimization problem in question has to be stated as a sequence of decision problems for the solver to consider. This is done by encoding each droplet  $i$  on all positions  $(x, y)$  at each time step  $t$  as a symbol  $d_{x,y,i}^t$ . Should a constraint, such as movement or fluidic constraints, prohibit the presence of a droplet at one such time step, the symbol is negated. The conjunction of these symbols forms the resulting formula and is given in Conjunctive Normal Form (CNF) to the solver. While the specific details are left out due to page limitations, understand that this approach retrieves the smallest number of time steps  $t$  with which this is solvable. Moreover, since this approach presents a way to finding exact solutions for routing problems while allowing generic position and movement formulation, this method allows for determining a routing on grids with alternative area divisions.

## IV. EXPERIMENTS

A set of 168 benchmarks were taken from [11] to test the routing performance on these field shapes. These include both benchmarks used for in-vitro-diagnostics as well as benchmarks used for protein crystallization.

To produce the appropriate grid configuration, the base grid was converted as described in Section III. The classification number is the amount of *Total Steps Necessary* to solve the respective benchmarks. Furthermore the *Total Difference*, the *Percentual Change* and the *Average Difference* were calculated.

In comparing the routing performance on these field shapes, 42 benchmarks had to be discarded. For them, no solution could be found for either or both rectangular or triangular field shapes because conversion had led to an unsolvable starting state.







Type of grid						
Total steps necessary	1642	1399	2269	2447	2897	2702
Total Difference (steps)	0	-243	+627	+805	+1255	+1060
Percentual Change	0%	=14.80%	±38.19%	±49.03%	±76.43%	±64.55%
Average Difference (steps)	0	=1.93	±4.98	±6.39	±9.96	±8.41

TABLE I: Overall Benchmark results for routing on squares (I), hexagons (II), rectangular triangles (III), equilateral triangles (IV), and transformed rectangular (V) and equilateral (VI) grids

Hexagons lead to a better overall performance on all benchmarks. No hexagonal benchmark resulted in worse routing times; in each benchmark, they either performed equally well or better than squares. They take significantly less steps (sum of steps in all benchmarks: 1399 vs. 1642) than squares in solving a benchmark.

Contrarily, the results for routing on equilateral and rectangular triangle field shapes proved worse. Interestingly, only one benchmark was unsolvable for equilateral triangles which was solvable for rectangular triangles. Overall, more benchmarks could be solved on equilateral triangles. In every benchmark equilateral triangles required equally as many or more steps to solve compared to square grids.

In contrast, on rectangular triangles, some benchmarks were actually solvable in *less* steps than on squares, for example for benchmark *Protein 24*. This benchmark combines some features that enable the exploitation of the rectangular triangle tiling in a way that leads to improved routing times: Due to the spacious droplet placement, they do not come into their respective interference regions during routing. This allows them to assume a more or less direct path to their target coordinates.

Secondly, in the rectangular grid, the spawn and target locations are placed almost diagonally from each other. This layout enables the simultaneous movement in two directions, without having to traverse three cells (like on squares) to move in a diagonal. Together, on similar grid configurations, this can actually lead to improved routing times.

However, in total, both equilateral and rectangular triangle grids performed worse. In the applicable benchmarks, rectangular triangles took nearly 40% more steps and equilateral triangles took almost 50% more; hence the latter proved to be the least effective method of area division in terms of routing performance.

A similar verdict can be passed for transformed grids. In comparison, a shift of routing performances can be noticed: Whereas in regular, non-transformed grid configurations, equilateral triangles proved worse, transformed equilateral triangle grids perform *better* than their rectangular counterparts. In total though, they performed expectedly significantly worse. Distributed over all 126 benchmarks, they take around 8 to 10 steps more on each benchmark.

Nevertheless, transformed grids do not perform dramatically worse compared to regular grids. Because of having to move over twice the number of cells along the x-axis of the grid and their higher ratio of movable to interference regions, an average increase in necessary steps of at least 50% could be expected (increasing 100% in one of two dimensions and assuming that routing benchmarks do not favour a specific direction, plus routing issues due to the ratio). However, the increased distance of droplets on the board, while at the same time making neighbourhood separation easier due to a higher number of cells, seems to be a deciding factor in the comparatively good performance of transformed grids.

## V. CONCLUSIONS

In this work a methodology for exact routing on grids comprised of alternative field shapes was proposed, i.e. hexagons, rectangular and equilateral triangles as well as the conventional square design. To do so, a routing method that computes exact solutions to these routing problems was illustrated and applied to a large set of benchmarks.

Hexagonal grids always perform better than or just as good as square grids. Triangular grids are, in total, less efficient in matters of routing performance. Under certain circumstances, utilizing triangular grids can result in an improved routing in relation to square grids though, e.g. when diagonal connections in the tessellation could be exploited.

Different area division methods thus lead to significantly differing results concerning routing performance and can thus help improving the overall performance of DMFBs. As such, these results can help decide on the suitability of these field shapes for their use in constructing electrode arrays for DMFBs.

## REFERENCES

- [1] P. L. Bowers and K. Stephenson. A “regular” pentagonal tiling of the plane. *Conformal Geometry and Dynamics of the American Mathematical Society*, 1(5):58–68, 1997.
- [2] K. Chakrabarty. Design and optimization methods for digital microfluidic biochips: A vision for functional diversity and more than moore. In *SOC Conference (SOCC), 2011 IEEE International*, pages 5–5, September 2011.
- [3] P. Datta, A. Dutta, R. Majumder, A. Chakrabarty, D. Dhal, and R. K. Pal. A technology shift towards triangular electrodes from square electrodes in design of digital microfluidic biochip. In *Electrical and Computer Engineering (ICECE), 2014 International Conference on*, pages 1–4. IEEE, 2014.
- [4] M. Frénéa, H. Lhermite, B. Le Pioufle, and H. Fujita. Design of biochip microelectrode arrays for cell arrangement. In *2nd Annual International IEEE-EMB Special Topic Conference on Microtechnologies in Medicine & Biology*, pages 140–143. IEEE, 2002.
- [5] D. Grissom, K. O’Neal, B. Preciado, H. Patel, R. Doherty, N. Liao, and P. Brisk. A digital microfluidic biochip synthesis framework. In *VLSI and System-on-Chip, 2012 (VLSI-SoC), IEEE/IFIP 20th International Conference on*, pages 177–182, Oct 2012.
- [6] O. Keszocze, R. Wille, and R. Drechsler. Exact Routing for Digital Microfluidic Biochips with Temporary Blockages. In *International Conference On Computer Aided Design*, pages 405–410. IEEE, 2014.
- [7] Z. Li, K. Y.-T. Lai, P.-H. Yu, T.-Y. Ho, K. Chakrabarty, and C.-Y. Lee. High-level synthesis for micro-electrode-dot-array digital microfluidic biochips. In *Design Automation Conference*, page 146. ACM, 2016.
- [8] Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. Solving SAT and SAT Modulo Theories. *Journal of the ACM (JACM)*, 53(6):937–977, 2006.
- [9] F. Su and K. Chakrabarty. Yield enhancement of reconfigurable microfluidics-based biochips using interstitial redundancy. *ACM Journal on Emerging Technologies in Computing Systems*, 2(2):104–128, 2006.
- [10] F. Su, K. Chakrabarty, and R. B. Fair. Microfluidics-based biochips: Technology issues, implementation platforms, and design-automation challenges. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(2):211–223, 2006.
- [11] F. Su, W. Hwang, and K. Chakrabarty. Droplet routing in the synthesis of digital microfluidic biochips. In *Design, Automation and Test in Europe*, volume 1, pages 1–6. IEEE, 2006.
- [12] G. Wang, D. Teng, Y.-T. Lai, Y.-W. Lu, Y. Ho, and C.-Y. Lee. Field-programmable lab-on-a-chip based on microelectrode dot array architecture. *IET Nanobiotechnology*, 8(3):163–171, 2014.