Droplet Routing for Digital Microfluidic Biochips Based on Microelectrode Dot Array Architecture

A Thesis Submitted to the College of

Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in the Department of Electrical and Computer Engineering

University of Saskatchewan

Saskatoon, Saskatchewan

By

Zhongkai Chen

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Electrical and Computer Engineering

57 Campus Drive

University of Saskatchewan

Saskatoon, Saskatchewan, Canada

S7N 5A9

ACKNOWLEDGMENTS

It is my great honor to have Prof. Hsiang-Yung Daniel Teng as my M.Sc. supervisor during my graduate study. First and foremost I offer my sincerest gratitude to him for his supervision, advice, and guidance from the very early stage of this research as well as giving me extraordinary experiences throughout the work, using his precious time to read this thesis and give critical comments about it. The completion of this thesis would not be possible without Prof. Teng's exceptional supervision and everlasting support. Also I would like to thank my supervisor Prof. Anh Dinh for his constant encouragement and support on my research and graduate studies.

I am heartily thankful to Prof. Dwight Makaroff for his comments and suggestions on this thesis, which greatly improved the quality of the thesis. The advice from Prof. Ramakrishna Gokaraju and Prof. Khan Wahid is really appreciated.

I would like to show my gratitude to Gary Wang's valuable discussion. I gratefully thank Prof. Hsung-Yi Ho in National Cheng Kung University for providing us experiment test cases. The financial support in the form of scholarships provided by the Department and the University is gratefully appreciated.

I wish to thank my parents for their continuous support and encouragement throughout my studies and life.

ABSTRACT

A digital microfluidic biochip (DMFB) is a device that digitizes fluidic samples into tiny droplets and operates chemical processes on a single chip. Movement control of droplets can be realized by using electrowetting-on-dielectric (EWOD) technology. DMFBs have high configurability, high sensitivity, low cost and reduced human error as well as a promising future in the applications of point-of-care medical diagnostic, and DNA sequencing. As the demands of scalability, configurability and portability increase, a new DMFB architecture called Microelectrode Dot Array (MEDA) has been introduced recently to allow configurable electrodes shape and more precise control of droplets.

The objective of this work is to investigate a routing algorithm which can not only handle the routing problem for traditional DMFBs, but also be able to route different sizes of droplets and incorporate diagonal movements for MEDA. The proposed droplet routing algorithm is based on 3D-A* search algorithm. The simulation results show that the proposed algorithm can reduce the maximum latest arrival time, average latest arrival time and total number of used cells. By enabling channel-based routing in MEDA, the equivalent total number of used cells can be significantly reduced. Compared to all existing algorithms, the proposed algorithm can achieve so far the least average latest arrival time.

Table of Contents

PERMISSION TO USE i
ACKNOWLEDGMENTS ii
ABSTRACT iii
Table of Contents iv
List of Tables vii
List of Figures viii
List of Abbreviations xii
Chapter 1 Introduction 1
1.1 Microfluidic Biochips 1
1.2 MEDA Digital Microfluidic Biochip 4
1.3 Motivation of Droplet Routing Algorithm 7
1.4 Objectives of the Thesis Work 9
1.5 Thesis Outline 10
Chapter 2 Microfluidic Droplets Routing 11
2.1 Droplet Routing Problem Formulation 11
2.1.1 Test Cases 11
2.1.2 Time Constraint 12
2.1.3 Fluidic Constraints 12
2.2 Basic Operations 16
2.3 Criteria 18
Chapter 3 Droplet Routing Algorithms 19
3.1 Electrode Addressing Solutions to DMFBs 19
3.2 Different Droplet Routing Algorithms 22

3.2.1 VLSI Global Routing....23
3.2.2 Droplet Routing Algorithms Using Sequential Approach....24
3.2.3 Droplet Routing Algorithm Using Concurrent Approach....27
3.3 Discussion....28

Chapter 4 3D Dynamic-Block-Based Droplet Router....29
4.1 Priority Setting....29
4.2 Routing Algorithm....30
4.2.1 Block Setting Algorithm....33
4.2.2 Net Routing....35
4.2.3 Dynamic Routing....39
4.2.4 Movement Control....40
4.2.5 Channel-Based Routing....41

Chapter 5 Implementation of 3D Dynamic-Block-Based Droplet Router....44
5.1 Data Layer....45
5.2 Logic Layer....51
5.3 Presentation Layer....52
5.4 Adjustable Parameters....55
5.5 Traditional DMFB to MEDA Conversion....56

Chapter 6 Results and Performance Comparison....59
6.1 Conversion of Equivalent Results....59
6.2 Effect of Priority Solvers on Traditional DMFBs....62
6.3 Effect of Straightness-Prone Parameter....66
6.4 Effect of Diagonal Movement....69
6.5 Effect of Used Cells Preference....70
6.6 Effect of Channel-Based Routing for MEDA....71
6.7 Comparision with Previous Published Results....81
6.8 Summary....83

Chapter 7 Conclusion and Future Work....85
7.1 Conclusion....85
7.2 Future Work....87

References 89
Appendix A 95
Appendix B 100
Appendix C 105

List of Tables

Table 3.1 Comparison of different electrode addressing solutions..... 22

Table 4.1 Notation used in the algorithm..... 31

Table 4.2 Temporary Destinations Decision Rules 38

Table 5.1 Notation used in MEDA conversion..... 57

Table 6.1 Notation used in conversion equations. 60

Table 6.2 Information of 4 test cases in benchmark suite III [23]. 63

Table 6.3 Failed sub problems with different priority solvers. 64

Table 6.4 Examples of the effect of straightness-prone parameter. 69

Table 6.5 Examples of the effect of diagonal movement parameter. 70

Table 6.6 Examples of the effect of used cells preference parameter. 71

Table 6.7 Results of Path-based + N.S. and Path-based + D., N.S. on traditional DMFBs. 82

Table 6.8 Optimized results of Path-based + N.S. and Path-based + D., N.S. on traditional DMFBs. 82

Table 6.9 Results of Path-based + N.S. and Path-based + D., N.S. in 3×3 channel-based MEDA. 83

Table 6.10 Results of Path-based + N.S. and Path-based + D., N.S. in non-channel-based 3×3 MEDA. 83

List of Figures

Figure 1.1 A continuous flow microfluidic biochip: 1. Substrate solution; 2. Liquid chromatography effluent; 3. Enzyme solution; 4. Flow towards mass spectrometer [4]. 2

Figure 1.2 The architecture of a DMFB cell [2]. (a) The principle of EWOD. (b) A EWOD-based bi-planar DMFB. 3

Figure 1.3 Highly customizable electrodes in MEDA [2]. 5

Figure 1.4 MEDA allows droplets to move diagonally. 5

Figure 1.5 The contact line (dotted line) in traditional DMFB and MEDA architectures while performing diagonal movement. (a) No contact line to perform diagonal movement in traditional DMFB architecture. (b) Long contact line to exert a large force on the droplet to perform diagonal movement in MEDA. 6

Figure 1.6 Experiment design flow of a MEDA DMFB. 6

Figure 2.1 Structure of a test case file..... 12

Figure 2.2 Parameters of a droplet. 13

Figure 2.3 Droplets of different sizes in a sub problem. 14

Figure 2.4 Potential undesired merging at the next time step. 15

Figure 2.5 Undesirable merging at two consecutive time steps..... 15

Figure 2.6 Moving droplet A by activating electrodes (green shaded grid) on DMFB. 16

Figure 2.7 Merging two droplets by activating the common neighbouring cell (green shaded grid) of two droplets. 17

Figure 2.8 Splitting one droplet to two droplets by activating two neighbouring cells. 18

Figure 3.1 Three electrode addressing solutions. (a) Direct-addressing [11]. (b) Pin-constrained [12]. (c) Cross-reference [13]. 20

Figure 3.2 Categorization of global-routing algorithms [30]..... 23

Figure 3.3 Graph coloring approach was utilized to avoid cross-reference violation. (a) Two vertices are connected with a solid line to activate a specific electrode. (b) The corresponding row and column vertices are connected by dotted line if a certain electrode is not supposed to be activated [13]..... 25

Figure 3.4 Four regions are used to evaluate the bypassibility of a net. 26

Figure 4.1 3D conversion of a routing problem. (a) Five available positions for the droplet at the next time step. (b) A cube defines the fluidic constraints. 32

Figure 4.2 (a) Shaded area is the blocks generated by the movement of a droplet in 2D view. (b) The blocks generated by the movement of a droplet in 3D view in one time step. (c) The blocks generated by a 2×2 droplet in MEDA. 34

Figure 4.3 Problematic 3-pin routing. 36

Figure 4.4 (a) Three available temporary destinations for two droplets. (b) to (d) The merging procedure by the proposed approach. 37

Figure 4.5 The block trimming procedure for 3-pin net, d_A will wait until d_B arrives at its own temporary destination for 1 time step. 39

Figure 4.6 (a) Diagonal movement disabled, Straightness-prone disabled. (b) Diagonal movement disabled, Straightness-prone enabled. (c) Diagonal movement enabled, Straightness-prone disabled. (d) Diagonal movement enabled, Straightness-prone enabled. 41

Figure 4.7 A sub problem is only routable by channel-based routing in MEDA. 42

Figure 4.8 Green shaded grid represents the reserved time steps for the “tail” of a 2×2 droplet. 43

Figure 5.1 Overview of the software architecture. 44

Figure 5.2 (a) A segment of the input description file. (b) visualization of sub problem 2. 46

Figure 5.3 Procedure of interpreting descriptions to objects. 47

Figure 5.4 Classes defined to model DMFB experiment. 49

Figure 5.5 Classes defined for realistic description of DMFB experiment. 50

Figure 5.6 A droplet to be moved from point A to point B. 51

Figure 5.7 Screenshot of the main user interface, the status of droplets and electrode array is shown in PlanarPicBox..... 53

Figure 5.8 Screenshot of simulation platform..... 54

Figure 5.9 Four extra cells are checked in MEDA, which are marked as orange diagonal arrows. 56

Figure 5.10 A sub problem in traditional DMFB and 3×3 MEDA architecture..... 58

Figure 6.1 The equivalent time steps and original total number of used cells (green shaded grid) in a 3 ×3 channel-based MEDA DMFB..... 61

Figure 6.2 The equivalent time steps and original total number of used cells (green shaded grid) in a 3 ×3 non-channel-based MEDA DMFB 62

Figure 6.3 Maximum latest arrival time of different combinations of priority solvers. 65

Figure 6.4 Average latest arrival time of different combinations of priority solvers. 66

Figure 6.5 Total number of used cells of different combinations of priority solvers. 66

Figure 6.6 Maximum latest arrival time of optimized results..... 67

Figure 6.7 Average latest arrival time of optimized results..... 68

Figure 6.8 Total number of used cells of optimized results..... 68

Figure 6.9 Two droplets with different sizes move for two equivalent cells, green shaded grid are counted as original used cells. 72

Figure 6.10 Straightness-prone, No diagonal movement, Used cells preference (MEDA). 73

Figure 6.11 No Straigtness-prone, No Diagonal movement, Used cells preference (MEDA). 74

Figure 6.12 Straightness-prone, Diagonal movement, Used cells preference (MEDA). 75

Figure 6.13 No Straightness-prone, Diagonal movement, Used cells preference (MEDA). 76

Figure 6.14 Straightness-Prone, No Diagonal movement No Used cells preference (MEDA). 77

Figure 6.15 No Straightness-prone, No Diagonal movement, No Used cells preference (MEDA). 78

Figure 6.16 Straightness-prone, Diagonal movement, No Used cells preference (MEDA). 79

Figure 6.17 No Straightness-prone, Diagonal movement, No Used cells preference (MEDA). 80

Figure 7.1 A case which cannot be solved by the proposed algorithm in traditional DMFBs. 87

Figure 7.2 SoKoBan puzzle [38] and its representation. 88

List of Abbreviations

DEP	Dielectrophoresis
DMF	Digital Microfluidics
DMFB	Digital Microfluidic Biochip
DNA	Deoxyribonucleic Acid
EWOD	Electrowetting-On-Dielectric
GDI	Graphics Device Interface
IC	Integrated Circuit
IDE	Integrated Development Environment
ILP	Integer Linear Programming
MEDA	Microelectrode Dot Array
NPC	Non-Player Character
OO	Object-Oriented
UI	User Interface
VLSI	Very Large Scale Integration
VS	Visual Studio

Chapter 1 Introduction

Microfluidics has been defined as “the science and technology of systems that process or manipulate small (10^{-9} to 10^{-18} litres) amounts of fluids, using channels with dimensions of tens to hundreds of micrometers” [1]. Digital microfluidics (DMF) is a new field of technology that manipulates microfluidic droplets on a patterned electrode array. Recently, an innovative DMF platform architecture, called Microelectrode Dot Array (MEDA) using electrowetting-on-dielectric (EWOD) was proposed by Wang, *et al*. [2]. This architecture has a good potential to achieve precise control of multiple droplets concurrently. Development of a droplet router is a necessary procedure for a new DMF architecture. The importance of droplet routing algorithms and the challenges are discussed in this chapter, followed by the motivation and objectives of the thesis.

1.1 Microfluidic Biochips

Microfluidic biochips are important applications of microfluidics; its basic idea is to provide a platform that integrates chemical or biological analyses on a single chip [3]. Early research on microfluidic biochips is based on continuous flow. Fluidic samples and regents are mainly controlled by micrometer-scale valves, actuators, sensors and pumps.

Although some applications such as DNA probing have been successfully implemented on continuous-flow microfluidic biochips, the number of applications is constrained due to its inability to make a complex design and limitation of flexibility. As shown in Figure 1.1, the channels for fluidic samples transportation are fixed after fabrication. Thus commercial products based on this technology (e.g. from Agilent, Fluidigm, Caliper, I-Stat, BioSite, etc.) are all application-specific [3].

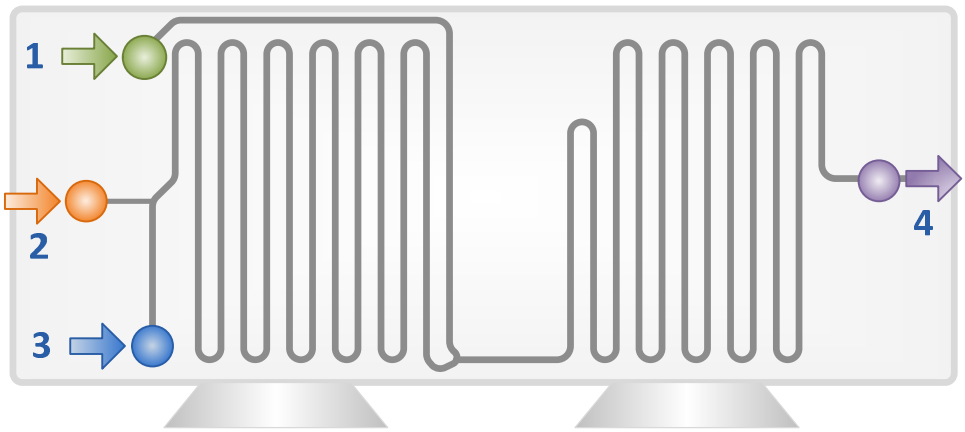


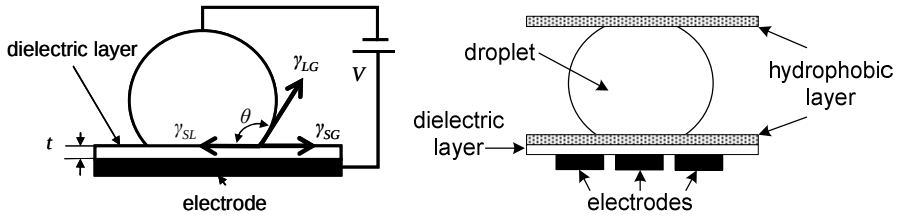
Figure 1.1 A continuous flow microfluidic biochip: 1. Substrate solution; 2. Liquid chromatography effluent; 3. Enzyme solution; 4. Flow towards mass spectrometer [4].

To overcome the drawbacks of traditional microfluidic biochip, new digital microfluidics technologies are developed to digitize fluidic samples into tiny droplets. With the development of digital microfluidics, multiple micro- or nano-litre droplets can be manipulated on an array of electrodes in parallel. Electrowetting-on-dielectric (EWOD) is one of the most commonly used electrical methods for manipulating microfluidic droplets. A hydrodynamic scaling model of droplet actuation was constructed in a systematic manner [5]. Based on the study result, reliable operations of a EWOD actuator is possible as long as the operations are within the limit of the Lippmann-Young equation

Eq. (1.1). Referring to Figure 1.2(a), the change of contact angle by the electric potential, *V*, can be described:

$$\cos\theta(V) - \cos\theta_0 = \frac{\varepsilon_0 \varepsilon}{2\gamma_{LG} t} V^2 \tag{1.1}$$

where $\theta(V)$ is the contact angle when a potential is applied, θ_0 denotes the equilibrium contact angle at V = 0V, ε_0 (8.85×10^{-12} F/m) the permittivity of vacuum , ε the dielectric constant of the dielectric layer, γ_{LG} the liquid-gas interfacial tension, and t its thickness the insulating layer. For droplet motion, a certain contact angle difference is required by applying adequate drive voltage. Note in Lippman-Young equation, the contact angle change is not related to the polarity of the applied potential, *V*. It should also be noted that the three-phase contact line (interface of three media) in contact with activated electrodes affects the force exerting on a droplet.



(a) (b)

Figure 1.2 The architecture of a DMFB cell [2]. (a) The principle of EWOD. (b) A EWOD-based bi-planar DMFB.

As shown in Figure 1.2(b), in a EWOD-based biochip system, droplets are usually sandwiched between two plates where the upper plate serves as ground electrode and the bottom plate is made up of a 2-dimensional electrode array. An activated electrode works like a “magnet” while a droplet is similar to an “iron” in this case. With the actuation of individually controlled electrodes, droplets can be moved from electrode to their adjacent electrodes. The capability of parallel droplet manipulation, together with design

portability and precision, fosters the possibility of concurrent chemical or biological analyses on a single digital microfluidic biochip (DMFB). A DMFB system performs microfluidic operations such as dispensing, moving, splitting, and merging of multiple droplets. Compared to conventional approach, DMFB has higher sensitivity, lower cost and human error as well as a promising future in the applications of point-of-care medical diagnostic and DNA sequencing [3, 6-9].

1.2 MEDA Digital Microfluidic Biochip

Due to the increased complexity of droplet manipulations, the scalability, configurability and portability are becoming important for DMFBs. To meet these requirements, recently a new DMFB architecture called Microelectrode dot array (MEDA) has been introduced by Wang *et al*. [2], which is also based on EWOD technology. Unlike the traditional DMFBs where a droplet is manipulated by one electrode, a droplet are controlled by a cluster of tiny electrodes called microelectrodes whose sizes can be 10 times smaller than traditional electrode (1 mm × 1 mm or larger). Similar to the dot-matrix printer, droplets can be configured to different patterns or shapes and occupy arbitrary number of microelectrodes. Figure 1.3 shows various sizes and shapes of electrodes can be formed on a MEDA DMFB for handling droplets with different sizes and properties.

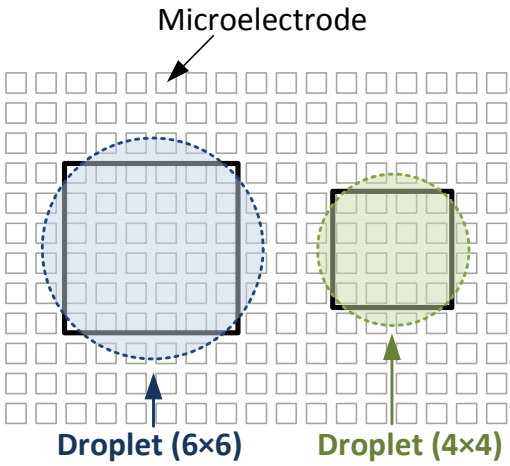
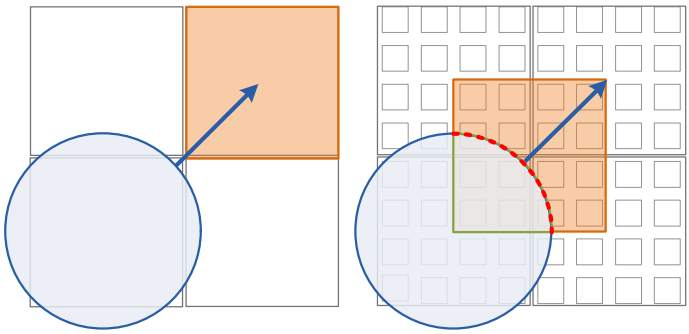


Figure 1.3 Highly customizable electrodes in MEDA [2].

As shown in Figure 1.4, diagonal movement is possible due to highly customizable electrodes in MEDA architecture. As shown in Figure 1.5(a), there might be less or even no contact line in traditional DMFB architecture, so it is unable to move a droplet diagonally. However, as shown in Figure 1.5(b), thanks to the flexibility of MEDA, a long contact line is formed by nearby microelectrodes around a droplet and is able to move the droplet diagonally by activating the microelectrodes gradually. As will be described in Section 4.2.5, it is also possible to generate a narrow virtual channel between source and sink of a droplet for fluidic transportation with MEDA architecture.

Figure 1.4 MEDA allows droplets to move diagonally.



(a) (b)

Figure 1.5 The contact line (dotted line) in traditional DMFB and MEDA architectures while performing diagonal movement. (a) No contact line to perform diagonal movement in traditional DMFB architecture. (b) Long contact line to exert a large force on the droplet to perform diagonal movement in MEDA.

A chemical or biological analyses using MEDA DMFB follows a similar general flow of traditional DMFBs (Figure 1.6):

Figure 1.6 Experiment design flow of a MEDA DMFB.

Experiment Requirements: This stage describes all the required operations of an experiment in a specific order. Take dilution as an example, certain samples needs to be at specific locations in order to mix with reagents. A portion of the mixed samples (as a droplet) under study is moved to a location for further analysis, while the rest is moved to a waste reservoir for disposal.

Case Description and Module Placement: This stage is also referred to synthesis stage. All the required operations are divided into many sub problems. According to the requirements of an experiment, modules such as mixing area, detection area and so on are

placed into each sub problem. Also the start positions and destinations of droplets are designated in this stage. All these information is saved as a case description file.

Routing: The case description file is passed to a droplet router as an input stream. The paths of all the droplets are calculated by droplet router one sub problem at a time.

1.3 Motivation of Droplet Routing Algorithm

The last stage of DMFB design flow, namely droplet routing, which is a stage to find paths between modules, and between modules and I/O ports (e.g., reservoirs), determines the effective usage of DMFBs. Like VLSI routing, droplet routing must also meet certain constraints such as time constraints. However, there are no permanent wire interconnections in droplet routing. A droplet moves along a path by successively activate adjacent electrodes. This temporary path can be shared by various droplets at different times. Several algorithms of droplet routing have been proposed in the past, but they are all designed for traditional DMFB architecture. It should be noticed that the term, "cell" is used interchangeably with the term "electrode" in those algorithms.

The primary goal of this work is to investigate a routing algorithm which can not only handle the routing problem for traditional DMFB architecture, but also be able to route different sizes of droplets and incorporate diagonal movements for MEDA architecture. Compared to traditional DMFB architecture, the complexity of routing for MEDA architecture is greatly increased due to concurrent movements of multiple droplets on a microelectrode array of higher "resolution".

To achieve the goal of droplet routing algorithm, there are four major challenges to be addressed in MEDA DMFB droplet routing:

1. **The routing algorithm can handle both traditional and MEDA DMFBs**

The existing DMFB droplet routing algorithms, which are designed for traditional DMFB architecture, cannot be applied to a MEDA DMFB directly due to the differences between traditional and MEDA architecture. As described previously, MEDA architecture allows different sizes of droplets, diagonal movements, etc. that are not possible in the traditional architecture. An effective routing algorithm that can meet the high complexity of routing and take advantage of the features of MEDA DMFB is required.

2. **Reduce total number of used cells**

Since some electrodes/microelectrodes may not work properly during run time, reducing *total number of used cells* can reduce the risk of malfunction. Moreover, lower cell usage also result in less cross contamination area and improve the accuracy of experiment such as bioassays as a result.

3. **Run time of droplet routing algorithm**

Because of the high configurability of DMFB, experiments such as bioassays operated on DMFB are frequently changed. Routing algorithms should solve the routing problem for a new experiment in an acceptable amount time. Those algorithms taking too much run time are impractical for real applications.

4. **Parallelism and maximum latest arrival time**

As discussed in Section 1.2, an experiment is divided into a number of small sub problems. In each sub problem, the time steps for the last droplet to reach its sink are defined as the latest arrival time of the sub problem. The average value of latest arrival times in an experiment is referred to *parallelism* or *average latest arrival time*. The

largest value of arrival times in an experiment is referred to *maximum latest arrival time*. Increasing parallelism and reducing maximum latest arrival time can save bioassay running time.

1.4 Objectives of the Thesis Work

The goal of DMFB droplet routing is to successfully find paths for all the droplets in a given time while meeting fluidic constraints. Here are the objectives of the thesis work:

1. The routing algorithm not only is able to handle droplet routing for traditional DMFB architecture, but also is able to route multiple droplets of different sizes in parallel and incorporate diagonal movements for concurrent assays on MEDA DMFB.
2. The total number of used cells, parallelism or maximum latest arrival time can be improved compared to all existing works.
3. The trend of paths and the movement directions are customizable in the routing algorithm.
4. Undesired merging of droplets before reaching their designated sink must be avoided in the routing algorithm.

1.5 Thesis Outline

Chapter 2 gives the formulation of droplet routing problem. Chapter 3 presents the previous work in droplet routing area. Chapter 4 and Chapter 5 describe the methodology and implementation of the proposed droplet routing algorithm. Chapter 6 shows the effects of enabling different adjustable parameters and compares the proposed algorithm with recent papers. Finally future work and conclusions are given in Chapter 7.

Chapter 2 Microfluidic Droplets Routing

This chapter introduces the formulation of a droplet routing problem and the structure of input test cases is presented, followed by time constraint and fluidic constraints in a droplet routing problem. Finally, the basic operations of DMFB, and the criteria of a droplet routing algorithm are discussed.

2.1 Droplet Routing Problem Formulation

Let $D_i = \{d_1, d_2, ..., d_n\}$ denote a set of droplets in sub-problem S_i. The goal of DMFB droplet routing is to find a path for each droplet in D_i to its designated destination (sink) while minimizing total number of used cells, maximum latest arrival time and average latest arrival time in a constrained time.

2.1.1 Test Cases

The experiments of DMFBs are described by test cases (See Figure 2.1). A test case consists of a number of sub problems. In each sub problem, there are a number of nets each defining the source and sink of a droplet. A 2-pin net simply indicates the source and sink for one droplet, while a 3-pin net describes two different sources of two droplets

and a common sink. Each sub problem in a test case is passed to a droplet router. The output is the sub problem with the routed paths of all droplets.

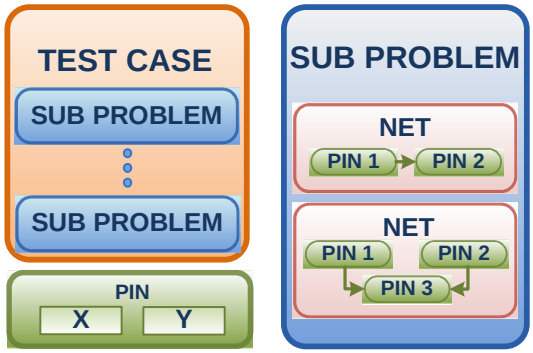


Figure 2.1 Structure of a test case file.

2.1.2 Time Constraint

The arrival time for the latest droplet in a sub problem (i.e. latest arrival time) is not allowed to exceed certain value. This value is defined as time constraint. Time constraint indicates the limited time for any droplet to move in a sub problem. If this constraint is violated, the routing is considered to have failed. In traditional DMFB architecture, it is assumed that a droplet can only move to the adjacent electrode in one time step. For example, a time constraint of 20 time steps is allowed in Benchmark Suite III [10]. The maximum total number of used cells for one droplet is 21 if time constraint is 20. The actual time of each time step is dependent upon the frequency of the actuation voltage signal.

2.1.3 Fluidic Constraints

In addition to time constraint, a droplet routing algorithm also needs to meet fluidic constraints. Since the proposed routing algorithm is for both traditional and MEDA DMFB architectures, where different sizes of droplets are created, moved, split, merged,

and mixed concurrently, the fluidic constraints are different from other droplet routing algorithms. To define droplets with different sizes, the parameters listed below are used (See Figure 2.2):

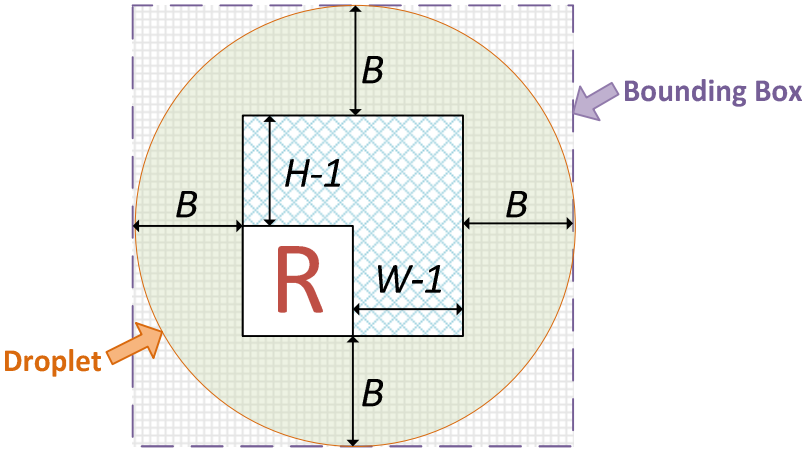


Figure 2.2 Parameters of a droplet.

- *R* is the reference point that indicates the position of a droplet;
- *H* and *W* are height and width of a droplet, respectively;
- *B* (bounding width) is the minimum distance between two droplets.

The outer box wrapping the droplet is called bounding box. To prevent unwanted mixing, a minimum bounding width of *B* must be constantly maintained between two droplets during routing. Note that the minimum distance between two droplets in traditional DMFB architecture is the width of one cell; i.e., $B=1$.

To simplify the presentation of experiments, the bounding boxes of droplets are not presented, i.e., the droplets shown in Figure 2.3 just fill the inner square shaded area in Figure 2.2, but actually the fluidic constraint between droplets is considered in the droplet routing algorithm. The sizes of droplets can be different based on the above definitions as shown in Figure 2.3. For example, d_1 is defined as a droplet whose width (*W*) and height

(*H*) are both equal to 3, while d_2 is defined as a droplet whose width (*W*) and height (*H*) are both equal to 2.

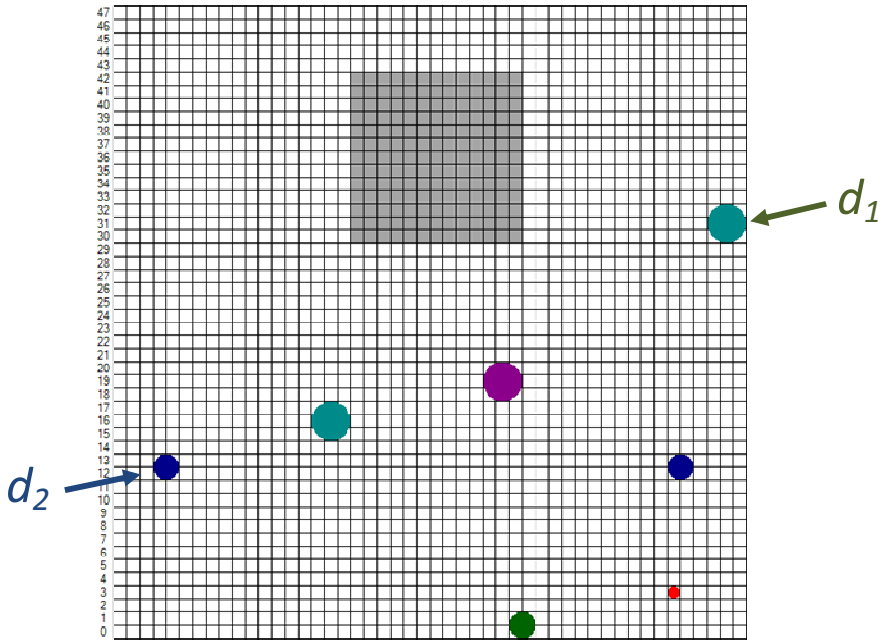


Figure 2.3 Droplets of different sizes in a sub problem.

There are two types of fluidic constraints to prevent unwanted mixing. Let (x_i^t, y_i^t), (x_j^t, y_j^t) denotes the positions of Droplet d_i and Droplet d_j at time step t, these two constraints can be described as follows:

1. Static constraint

Figure 2.4 shows the positions of two droplets at time *t*-1. d_A is at (1, 1) and d_B is at (3, 2). At time *t*, two cells located at (1, 2) and (2, 2) are activated, which causes d_A and d_B merged undesirably. Thus a static constraint defines the minimum distance between droplets at any given time to prevent this situation:

$$| x_i^t - x_j^t | > B \text{ or } | y_i^t - y_j^t | > B \qquad (2.1)$$

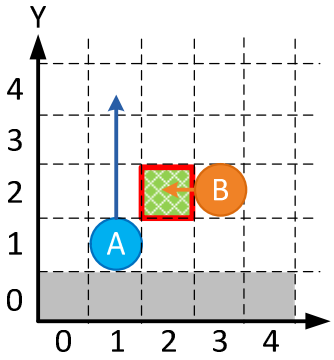


Figure 2.4 Potential undesired merging at the next time step.

2. Dynamic constraint

Figure 2.5 shows the positions of two droplets at time step t; d_A is at (1, 1) and d_B is at (3, 2). At time t+1, two cells located at (0, 1) and (2, 2) are activated. No matter where d_A is at time t+1, once (2, 2) is activated at time t+1, d_A and d_B will be merged undesirably. Thus a dynamic constraint defines the minimum distance between droplets at two consecutive time steps to prevent this situation:

$$|x_i^{t+1} - x_j^t| > B \text{ or } |y_i^{t+1} - y_j^t| > B \tag{2.2}$$

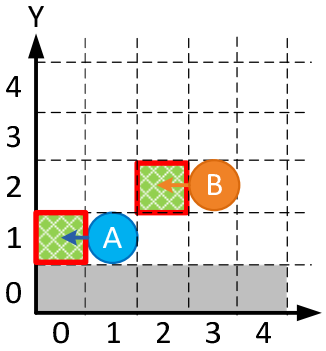


Figure 2.5 Undesirable merging at two consecutive time steps.

2.2 Basic Operations

There are four basic operations in DMFBs: transportation, merging, mixing and splitting. It should be noted that mixing and splitting operations are not used by the given test cases in this thesis work as this thesis work focuses on routing rather than detailed operations of mixing or splitting, but they are important operations for other experiments. Each operation is described as follows:

1. Transportation

Transportation is the most common operation in a DMFB. This operation simply moves a droplet from one place to another place by activating adjacent electrodes in sequence. Figure 2.6 shows moving a droplet A from (1, 1) to (3, 1) by activating electrodes (green shaded grid). In a traditional DMFB, only vertical and horizontal movements are allowed (i.e. up, down, left and right). As described in section 1.2, MEDA allows four extra diagonal movements for routing (i.e. up-left, up-right, down-left and down-right).

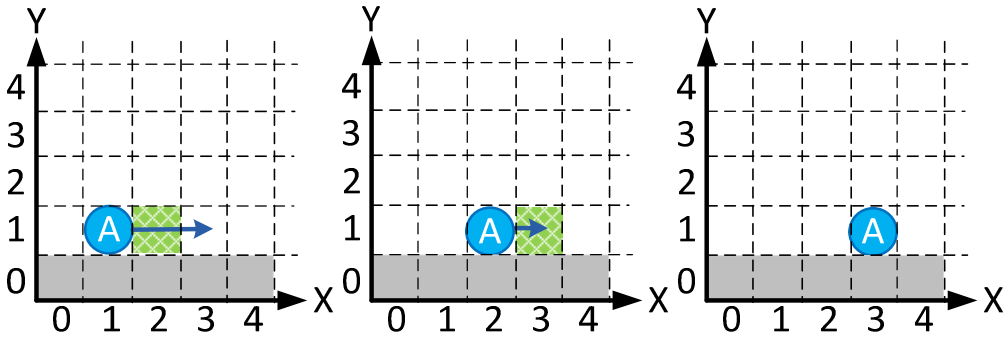


Figure 2.6 Moving droplet A by activating electrodes (green shaded grid) on DMFB.

2. Merging

It is convenient for DMFB to perform merging operation when a droplet needs to be merged with another fluidic sample or reagent. As shown in Figure 2.7, two droplets located at (1, 1) and (3, 1) are merged by activating their common neighbouring cell at (2, 1). It should be noted that Figure 2.7 does not show the actual size of the merged droplet.

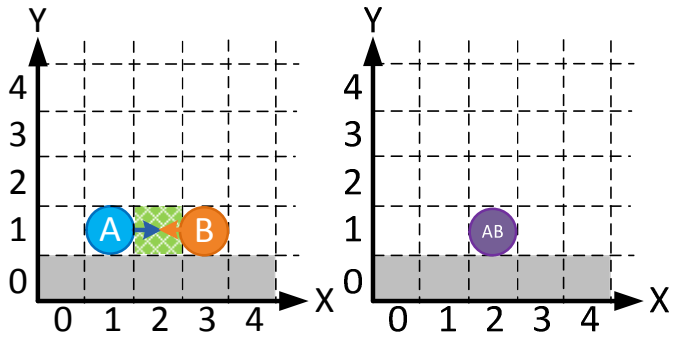


Figure 2.7 Merging two droplets by activating the common neighbouring cell (green shaded grid) of two droplets.

3. Mixing

Two droplets may not be mixed thoroughly after being merged together; a merged droplet needs to be moved forward and backward for several times to ensure the effect of mixing. Usually a mixing module should be placed on a reserved area of a biochip, which is only used for mixing operation.

4. Splitting

Splitting is the reversed operation of merging. This operation is usually used for pre-processing of dilution. With this operation, a sample with high density is split and further mixed with diluents. By activating two neighbouring cells around a droplet, a droplet can be split into two droplets. Figure 2.8 shows the droplet located at (2, 1) is split to two

droplets at (1, 1) and (3, 1). It should be noted that Figure 2.8 does not show the actual size of the split droplet.

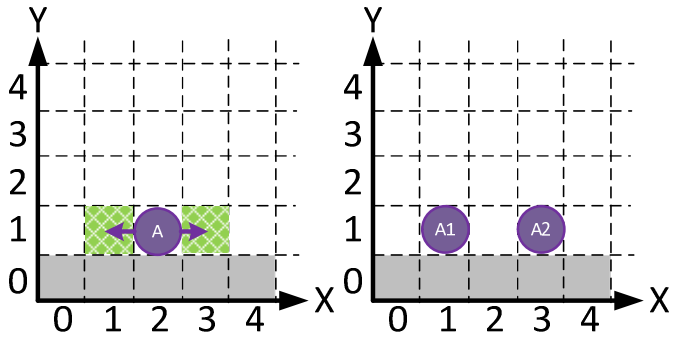


Figure 2.8 Splitting one droplet to two droplets by activating two neighbouring cells.

2.3 Criteria

The droplet routing problem can be solved by many algorithms; several criteria are defined to evaluate the quality of a routing algorithm. The first criterion is the total number of used cells. Effective total number of used cells can reduce contamination area and gain better fault tolerance. The second criterion is average latest arrival time of a test case. This number can represent the parallelism of the routing algorithm which represents total amount of time the experiment takes. The third criterion is maximum latest arrival time, which represents the efficiency of algorithm; especially it shows the capability of handling special sub problems by the routing algorithm. Since the traditional architecture and MEDA architecture are different in terms of “resolution”, a fair comparison of routing algorithms for these two architecture require appropriate conversions of the resulting total number of used cells, parallelism and maximum latest arrival time, as will be discussed in Section 6.1.

Chapter 3 Droplet Routing Algorithms

The objective of droplet routing algorithms is to find paths between modules, and between modules and I/O ports (e.g., reservoirs), which determines the effective usage of DMFBs. Like VLSI routing, droplet routing must also meet certain constraints such as time constraints and fluidic constraints. However, there are no permanent wire interconnections in droplet routing. A droplet moves along a path by successively activate adjacent electrodes. This temporary path can be shared by various droplets at different times. In this chapter, three common electrode addressing solutions are presented. Due to the similarity to droplet routing, VLSI routing is also briefly reviewed. The recent papers in droplet routing area are then discussed, which are further divided into two categories based on the features of droplet routing algorithms: sequential approach and concurrent approach.

3.1 Electrode Addressing Solutions to DMFBs

In DMFB, each electrode can be controlled through a unique address. There are mainly three electrode addressing solutions to DMFBs: direct-addressing, pin-constrained

and cross-reference. The droplet routing algorithms are different depending on different electrode addressing solutions.

In direct-addressing solution, each electrode is connected to individual pin. Figure 3.1(a) shows the architecture of direct-addressing solution. Each electrode is controllable (i.e. activated and deactivated) by a dedicated pin. It has the highest controllability compared to the other two solutions. However, it needs a large number of control pins (i.e., N×M pins) and could increase the size of control module.

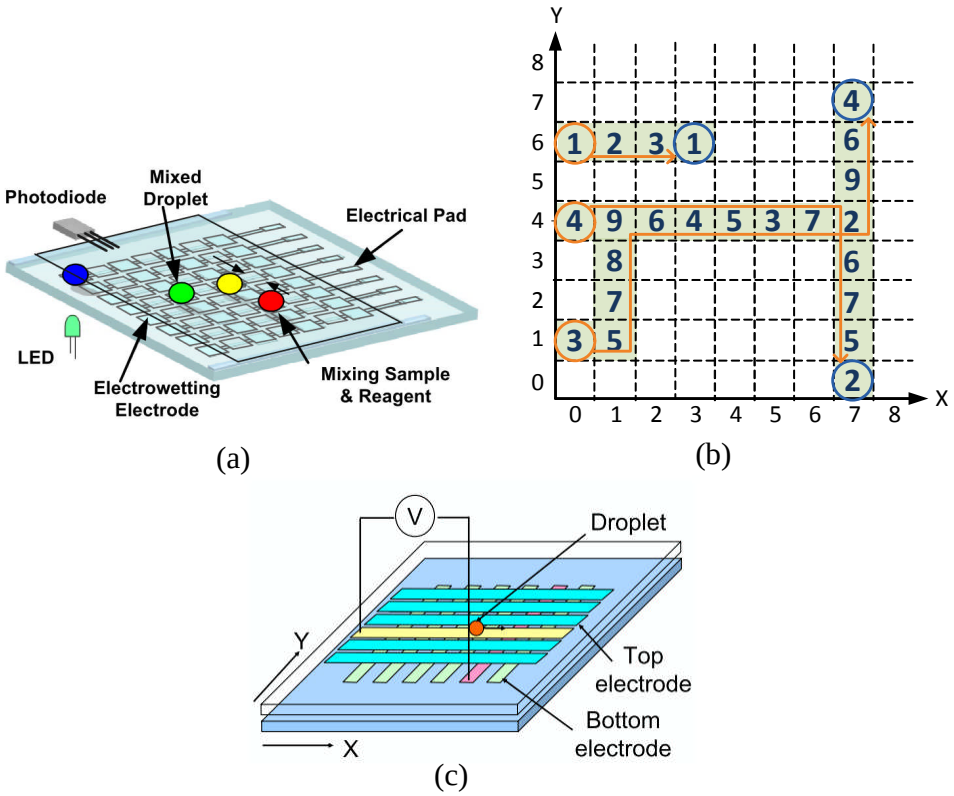


Figure 3.1 Three electrode addressing solutions. (a) Direct-addressing [11]. (b) Pin-constrained [12]. (c) Cross-reference [13].

Pin-constrained solution was proposed by T. Xu *et al.* [10, 14]. The basic idea of this solution is to partition electrodes into several groups. In order to reduce control pins, a group of electrodes is connected to a single control pin. Figure 3.1(b) shows the

architecture of pin-constrained solution, in which a same number indicates the corresponding electrodes are connected to and controlled by the same pin. Although the number of control pins can be reduced, the pin-constrained solution has low flexibility because it is usually specific to a biofluidic application.

Cross-reference solution [15, 16] reduces the control pins from N×M of a direct-addressing solution to N+M. Figure 3.1(c) shows the architecture of cross-reference solution. The electrode bars are placed on both top and bottom plates orthogonally. An electrode is activated when both top and bottom bars are activated. Though the control pins can be reduced, this solution may result in the activation of unintended electrodes.

Table 3.1 summarizes the advantages and disadvantages of these electrode-addressing solutions as well as lists the routing algorithms for specific solution. It can be seen that most of the papers are based on direct-addressing solution, as it has highest flexibility by taking full advantage of the configurability of DMFB. Therefore the proposed droplet routing algorithm is also based on direct-addressing solution. The next section will review existing work in droplet routing area.

Table 3.1 Comparison of different electrode addressing solutions.

Type	Advantage	Disadvantage	Routing algorithms
Direct-addressing	Highest flexibility, electrodes activation is not constrained	Large number of control pins	[11, 17-23]
Pin-constrained	Reduced control pins	Less flexibility, specific to a biofluidic application	[24]
Cross-reference	Reduced from N×M control pins to N+M	Less flexibility, activation of unintended electrodes	[13, 25]

3.2 Different Droplet Routing Algorithms

The goal of VLSI global routing is to connect the pins of functional modules while using less cost for these connections (e.g. wire-length or edge congestion), as well as meeting constraints such as timing constraint and design rules. Due to the similarity between VLSI global routing and DMFB droplet routing, it is valuable to investigate VLSI global routing algorithms as they have been developed for decades of years and become a well established technology in integrated circuit (IC) design. Since the global routing problem has been proved to be NP-hard [26], a heuristic approach is utilized to get approximate results. Like VLSI global routing, droplet routing algorithms can also be divided into sequential approach and concurrent approach. In this section, recent papers in droplet routing area are reviewed based on these two categories.

3.2.1 VLSI Global Routing

In the global routing stage in VLSI design, it is assumed that the circuits are placed in one layer. Pins of chip components are located at the interaction (i.e. vertices) and channels are located on edges in a lattice graph. Like droplet routing, a net is defined as a group of pins which need to be connected together [27, 28].

As shown in Figure 3.2, in the sequential approach, all the nets are sorted based on certain criteria, and then routed in sequence. There are a number of algorithms using this approach such as Lee algorithm [29] and Soukup algorithm [30]. The advantage of this approach is lower complexity and faster running time compared to concurrent approach. The disadvantage is that the appropriate sequence needs to be decided prior to routing; otherwise, the routability or the quality of routing might be affected.

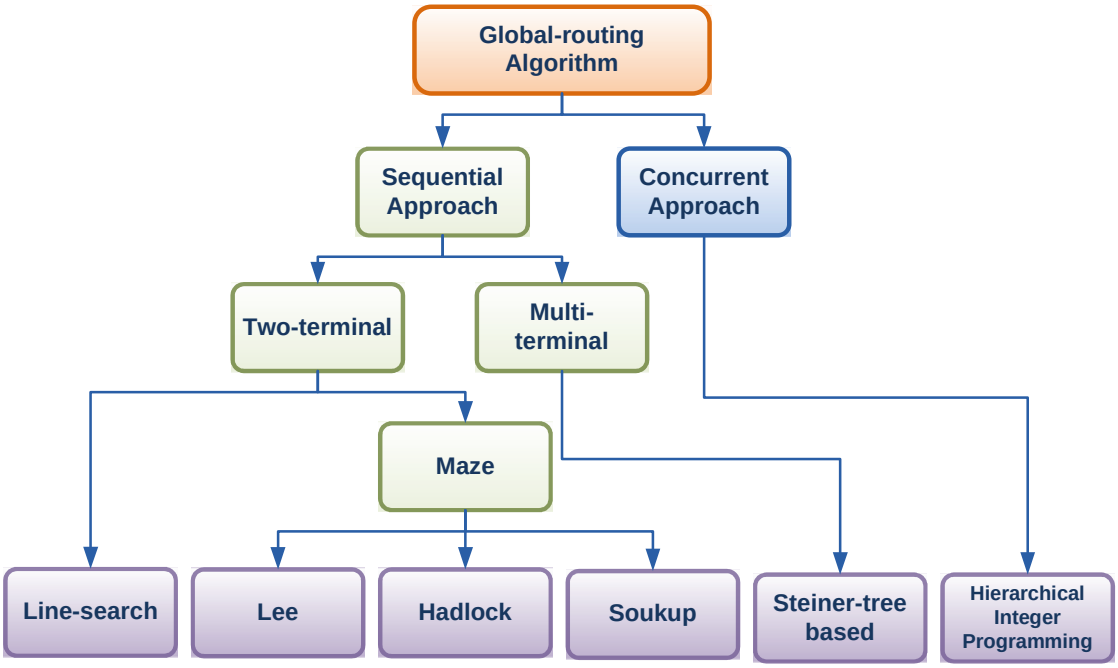


Figure 3.2 Categorization of global-routing algorithms [30].

In the concurrent approach, all the nets are routed in parallel. Usually integer programming such as integer linear programming (ILP) is utilized to find paths for all nets simultaneously. A series of constraint functions and an objective function need to be defined in the modeling stage before calculation. Generally, directly applying ILP is impossible as the VLSI routing is very complex and computation-intensive [27]. To apply ILP, the algorithm must be modified to reduce computation time [31-33].

3.2.2 Droplet Routing Algorithms Using Sequential Approach

Most of the droplet routing algorithms also use sequential approach in a similar way as VLSI global routing algorithms do. A prioritized A* search algorithm proposed by K. Bohringer [17], which is based on two dimensions, does not achieve good routability partly because priorities of droplets are randomly assigned, and droplet stalling for collision prevention is not automatically controlled. Without consideration of the 3-pin net problem and the time constraint, the practical value of this algorithm is compromised to a certain degree.

An algorithm based on Internet routing protocol was proposed by P. Yuh *et al.* [22]. This algorithm requires a routing table to store fixed patterns, which does not fully exploit the dynamic feature of DMFB.

A two-stage algorithm was proposed by F. Su *et al.* [11], where the routing problem is divided into two stages: path finding and rules check. At the first stage, the algorithm applies Lee maze search algorithm[29] to achieve M-shortest paths for each net. If the paths between nets are violated with each other, several rules are applied to modify the paths. As path finding and rules violation may interfere with each other, the separated stages may cause low routability.

A droplet routing algorithm with capability of solving cross-reference problem was proposed by Z. Xiao *et al.* [13]. The priority of nets is determined by the bounding boxes of source and sink pins as well as the Manhattan distance between them. A heuristic path searching approach is applied to each net. A 2-color graph coloring approach is then utilized to solve the cross-reference violation. Figure 3.3 demonstrates graph coloring approach. *R*N represents the activation of N*th* electrode row; similarly, *C*M represents the activation of M*th* electrode column. Two vertices are connected with a solid line to activate a specific electrode. If a certain electrode is not supposed to be activated, as shown in Figure 3.3(b), the corresponding row and column vertices are connected by dotted line.

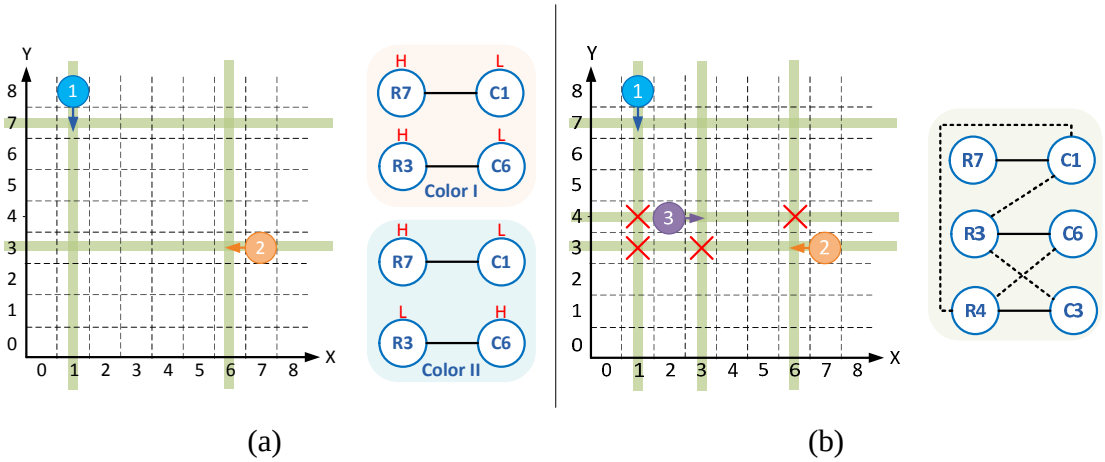


Figure 3.3 Graph coloring approach was utilized to avoid cross-reference violation. (a) Two vertices are connected with a solid line to activate a specific electrode. (b) The corresponding row and column vertices are connected by dotted line if a certain electrode is not supposed to be activated [13].

An algorithm with high routability was proposed by M. Cho *et al.* [21], which is refered to as Cho's algorithm hereafter. To calculate the priority of nets, "bypassibility" of each net is analyzed first. The bypassibility of a droplet is determined by the available routing regions around the droplet. As shown in Figure 3.4, the cells around the sink *T*

are divided into four regions labelled as Horizontal Up and Down, and Vertical Up and Down. A region is said to be available if all the cells in the region are not being used. Higher bypassibility means more available regions. All the nets are routed by maze algorithm based on the bypassibility, and nets are sorted again according to the arrival times of droplets. Finally all the nets are routed by 3D min-cost search algorithm in sequence.

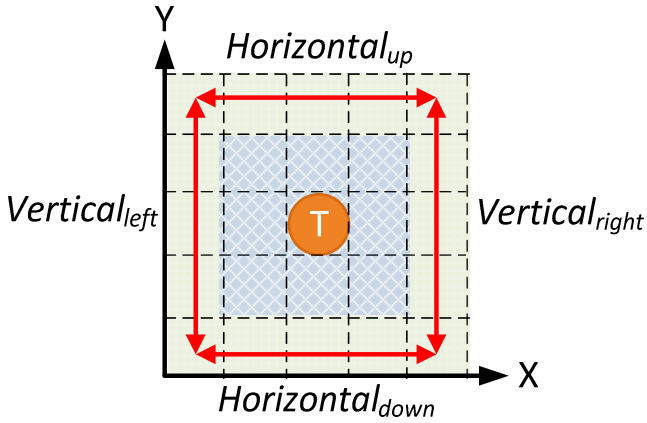


Figure 3.4 Four regions are used to evaluate the bypassibility of a net.

An algorithm with a high routability as well as the lowest total number of used cells, average latest arrival time and maximum latest arrival time (at the time of publication in 2010) was proposed by T. Huang *et al.* [23], which is refered to as Huang's algorithm hereafter. An Entropy-based priority solver sets priority to each net based on the type of pins and area of module blocks in bounding box of a net. A preferred routing tracks approach is then applied to determine paths for each net. Both of Cho's algorithm and Huang's algorithm implemented their own concession solver to improve the routability.

3.2.3 Droplet Routing Algorithm Using Concurrent Approach

ILP is used in several DMFB droplet routers [12, 25]. The objective of droplet routing is to find paths for nets while using as less total number of used cells, latest and average latest arrival time as possible. Thus usually the “objective function” of ILP is to minimize the algebraic sum of these three numbers.

Generally, cross-reference problem is solved after routing [13]. P. Yuh *et al.* emphasizes more on routing and solving cross-reference problem simultaneously [25], which adds a series of constraint functions to avoid cross-reference violation in the model of ILP. The authors proposed a progressive-ILP scheme to reduce the algorithm complexity.

DMFB droplet routing problem is modeled using basic ILP by T. Huang *et al.* [12], which requires a significant amount of time (at least 5 days to solve one benchmark) to solve the routing problem [12]. To reduce computation time, a two-stage technique of global routing was used, followed by incremental ILP-based routing. Reduction of computation time was also achieved by a deterministic ILP.

3.3 Discussion

To take full advantage of both traditional and MEDA, the flexibility is a critical concern to ensure efficient droplets manipulation. Thus direct-addressing solution is utilized in this work. Since the droplet size may vary in MEDA, the complexity of routing problem increases greatly as the size changes. For example, if a droplet occupies 7×7 microelectrodes and all the droplet sizes are assumed to be same, the size of array changes from M×N to M×N×7×7. ILP is not suitable for solving the problem with such a high complexity. Therefore, sequential approach is used in the proposed algorithm. Two recent papers with similar approach [21, 23] are selected to compare with this thesis work.

Chapter 4 3D Dynamic-Block-Based Droplet Router

Since the proposed routing algorithm is based on sequential approach, priority setting and path finding need to be determined respectively. In this chapter, a *path-based priority solver* is proposed, which can also be combined with other priority solvers. In terms of path finding, a *block setting* algorithm is implemented to apply A* search algorithm to the droplet routing problem. To achieve high routability, a *dynamic routing* algorithm is applied during routing as well. Droplet movement control and a new *channel-based* routing approach for MEDA are also presented.

4.1 Priority Setting

Typically, prior to solving a sub-problem, a priority solver assigns priorities to individual nets in the sub-problem. The routing algorithm routes each net based on their priority. Appropriate priority setting is essential to reduction of blockage of lower-priority droplets and prevention of failure of routing. To integrate the priority setting into the proposed routing algorithm, a new priority solver called path-based priority solver is proposed for initial priority settings. In this path-based priority solver, each net is routed

by a 2D A* search algorithm without considering other droplets. The following rules are then applied to each net:

- If a droplet passes through the sources of other droplets, other droplets are assigned with higher priorities.
- If a droplet passes through the sinks of other droplets, the droplet is assigned with higher priority.

The initial priorities determined by path-based priority solver, however, do not always achieve 100% routability. As will be described later in section 4.2.3, routability can be further improved by using dynamic routing algorithm.

4.2 Routing Algorithm

The routing algorithm is also an essential part of droplet routing. It is the procedure to determine paths for all the nets in a test case through block setting, net routing and dynamic routing. Algorithm 1 shows the overall droplet routing algorithm using dynamic routing, where the notations are listed in Table 4.1. The initial priority is decided by the selected priority solver (Algorithm 1, Line 2) based on the features of droplets. As the initial priority setting does not always result in 100% routability, the priorities are changed in the proposed dynamic routing approach (Algorithm 1, Line 3 to Line 21). Path finding and block setting algorithm vary with the types of nets (Algorithm 1, Line 7 and Line 9-17).

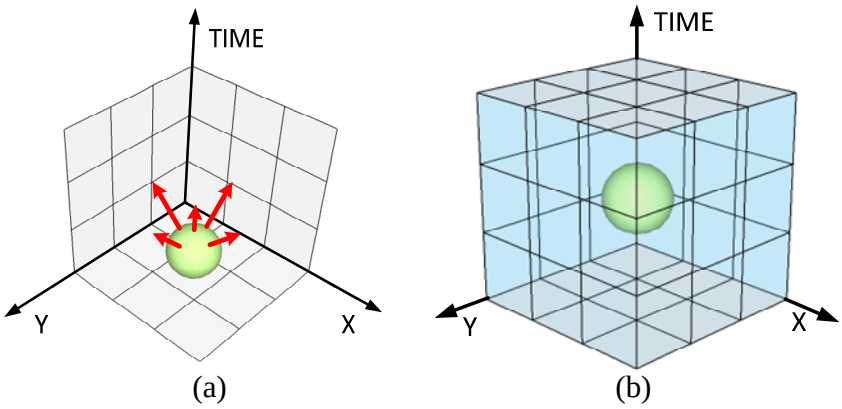
Table 4.1 Notation used in the algorithm.

D	Set of droplets	T_{di}	Sink position of d_i
d_i	Droplet i	$\hat{T}_{di}$	Temporary target position of d_i
$\hat{d}_i$	Related droplet of d_i for a 3-pin net	c	A cell on the routing plate
σ	Priority sequence of droplets	$A_{di,c}$	Arrival time of d_i to cell c
π_{di}	The priority indicator of droplet d_i	B_c	The bounding box area of cell c
G	A 3D map of DMFB	$\overline{A}$	Time constraint of routing
$\tilde{d}_i$	Pin type of d_i	P_{di}	Routed path of d_i
R	Set of routed paths		

Algorithm 1 Overall Algorithm

1 begin
2 $\sigma \leftarrow$ apply priority solver based on the features of D
3 repeat
4 $D \leftarrow$ sort D in descendent order according to σ
5 foreach d_i in D do
6 if $\tilde{d}_i \in$ 2-pin
7 $P_{d_i} \leftarrow$ FindPathAndMarkBlock(d_i , T_{d_i} , G)
8 elsif $\tilde{d}_i \in$ 3-pin
9 if $P_{\hat{d}_i} \neq \phi$
10 $P_{d_i} \leftarrow$ FindPathAndMarkBlock(d_i , $\hat{T}_{d_i}$, G)
11 $P_{d_i} \leftarrow$ Update path to T_{d_i} for d_i
12 $P_{\hat{d}_i} \leftarrow$ Update path to $T_{\hat{d}_i}$ for $\hat{d}_i$
13 MarkBlockForThreePin(P_{d_i} , $P_{\hat{d}_i}$, G)
14 else
15 $\{\hat{T}_{d_i}, \hat{T}_{\hat{d}_i}\} \leftarrow$ Set temporary destinations for d_i and $\hat{d}_i$, respectively
16 $P_{d_i} \leftarrow$ FindPathAndMarkBlock(d_i , $\hat{T}_{d_i}$, G)
17 endif
18 endif
19 $R \leftarrow$ update P_{d_i}
20 endfor
21 until $P_{d_i} \neq \phi$
22 return R
23 end

In the proposed routing algorithm, droplet routing on an electrode array is extended to a 3D space. The *x* and *y* axis correspond to the original definition of 2D electrode array. The *z* axis corresponds to the time steps. The time constraint limits the maximum value of z. Therefore the routing problem can be considered as routing all the nets in a 3D container. Consider a droplet located at (*x*, *y*) at time step *t*, which is represented as (*x*, *y*, *t*). If only vertical and horizontal movements are allowed, there are 5 available positions at the next time step: (*x*+1, *y*, *t*+1), (*x*-1, *y*, *t*+1), (*x*, *y*+1, *t*+1), (*x*, *y*-1, *t*+1) and (*x*, *y*, *t*+1). Figure 4.1(a) shows a 3D space that is converted from a 3×3 array with constraint of 3 time steps. Five arrows around the droplet indicate the possible positions for a droplet at the next time step.



(a) (b)

Figure 4.1 3D conversion of a routing problem. (a) Five available positions for the droplet at the next time step. (b) A cube defines the fluidic constraints.

As shown in Figure 4.1(b), the fluidic constraints imposed by the bounding width with $B = 1$ can be represented by a cube. If a droplet is located at (*x*, *y*, *t*), a 3×3 block cube centered by this droplet cannot be entered by other droplets. If a path is found for a droplet with higher priority, block cubes are generated along this path. Droplets with lower priorities have to avoid entering these block cubes. DMFB routing is similar to a

3D path finding problem. In computer games like the real-time strategy game [34], it is also a common and critical problem for a Non-Player Character (NPC) to find a shortest path to its destination without collisions with enemies. A* search algorithm is widely used in this type of games due to its high performance and promising results. Inspired by the solution to the robot motion planning problem, the proposed routing algorithm applies 3D-A* search algorithm to find the path for each net. There are two major advantages. First, 3D-A* search algorithm is able to route different sizes of droplets in a 3D space, which is important to droplet routing for highly configurable DMFB architecture such as MEDA. Secondly, the trend of path and moving directions can be customizable compared to other routing algorithms [11, 17-23].

4.2.1 Block Setting Algorithm

A block is defined as an area that cannot be entered by droplets. In addition to the blocks caused by existing modules on an electrode array, blocks are also generated by the paths of higher priority droplets. A block setting algorithm, which is an important part of the routing algorithm, is dependent on the types of nets due to the differences of fluidic constraints and the differences between 2-pin and 3-pin nets. To use 3D-A* search algorithm to find a path of each net, a block setting algorithm must consider the types of nets.

Block setting for paths: Figure 4.2 shows the generated blocks by the movement of a droplet in 2D and 3D views. To meet fluidic constraints, block cubes are created along the path of a higher priority droplet so that lower priority droplets are not allowed to enter these areas at certain specific time steps. Symbol *S* is referred to source and symbol *T* is

referred to target hereafter. Note that Figure 4.2(c) shows the block setting of a 2×2 droplet in MEDA, in which, the size of blocks is determined by the size of the droplet.

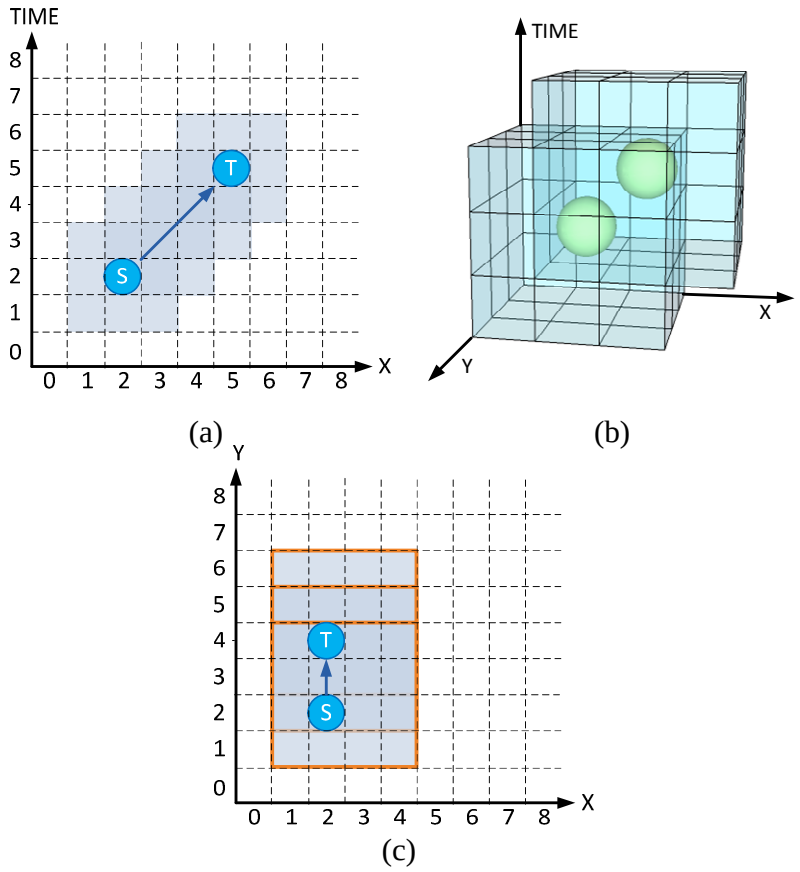


Figure 4.2 (a) Shaded area is the blocks generated by the movement of a droplet in 2D view. (b) The blocks generated by the movement of a droplet in 3D view in one time step. (c) The blocks generated by a 2×2 droplet in MEDA.

3D-A* search algorithm is able to automatically handle the stalling operation and some congestions for the low-priority net. Droplets are allowed to share cells at different time steps. For instance, lower priority droplet can pass through shared cells ahead of higher priority droplets; otherwise the lower priority droplet will be moved to a temporary location or wait for a certain time steps until congestion area is freed to pass through. Algorithm 2 shows the block setting algorithm for 2-pin nets. Once a droplet

reached its sink, all the 3D block cubes along the path except the sink are marked (Algorithm 2, Line 2). For a general droplet, blocks with the same size of the bounding boxes of its sink are generated along the time axis from the arrival time till the end of experiment (Algorithm 2, Line 9). For waste droplets, there is no more block after time *t* since they will be moved to a waste reservoir at the sink at time *t*. During routing, a higher priority droplet may pass through the sink of a lower priority droplet. A lower-priority droplet arrived at its sink first may block higher priority droplets. As a result, it is necessary to create a block to prevent the lower priority droplet from entering its sink (Algorithm 2, Lines 3-7) before higher priority droplets pass through the position.

Algorithm 2 MarkBlockForTwoPin

1 foreach c in $P_{d_i} \setminus T_{d_i}$ do
2 $G \leftarrow$ generate B_c at the corresponding time step of *c*
3 foreach d_j in D do
4 if $c = T_{d_j}$
5 $G \leftarrow$ generate B_c from time step 0 to the corresponding time step of *c*
6 endif
7 endfor
8 endfor
9 $G \leftarrow$ generate $B_{T_{d_i}}$ from $A_{d_i, T_{d_i}}$ to $\overline{A}$

Block setting for modules: The synthesis stage produces different modules placements for each sub-problem [35]. These modules, existing throughout each experiment, cannot be used for routing. The block setting algorithm must generate blocks for these modules from time step 0 to the maximum time step in a 3D space.

4.2.2 Net Routing

2-pin nets and 3-pin nets need to be processed differently. A 2-pin net, which involves one droplet, defines the source and sink of a droplet. A 3-pin net defines two

droplets that need to be merged into a larger droplet for mixing or dilution. As shown in Figure 4.3, traditionally, one of these two droplets is routed to its sink first, and then the other droplet is routed to the same sink. This approach will move the first droplet out from its sink when the other droplet approaches the sink position. The unnecessary move of the first droplet may cause problem in practical use. A realistic approach to handle 3-pin net is implemented in the proposed routing algorithm. Similar to the traditional approach, the 3-pin net is divided into two 2-pin nets. Instead of routing the two droplets to their common sink sequentially [21], they will be routed to their own temporary destinations which are adjacent to the sink. It should be noted that the priority of these two nets is not necessary in a consecutive order. These two droplets will be merged at the time when the second droplet arrives at its temporary destination, which greatly increases the routing flexibility.

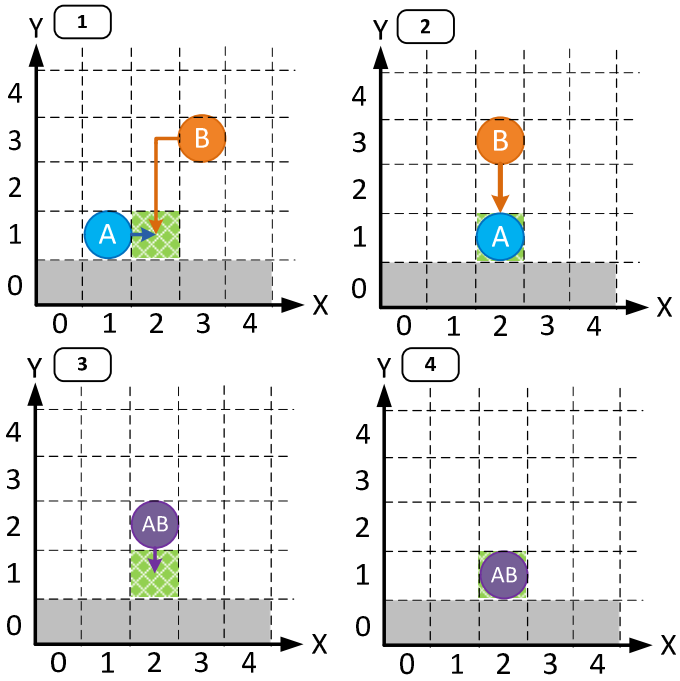


Figure 4.3 Problematic 3-pin routing.

To find the temporary destinations for two droplets (Algorithm 1, Line 15), the availability of four positions around the sink are checked. The example in Figure 4.4(a) shows that there are three available temporary destinations for a 3-pin net. Based on these positions, the temporary destinations are chosen for two droplets. Table 4.2 describes the rules that are used to decide the temporary destinations for a droplet. If two or more temporary positions are available for the droplet, the closer temporary destination for the droplet will be used by the routing algorithm.

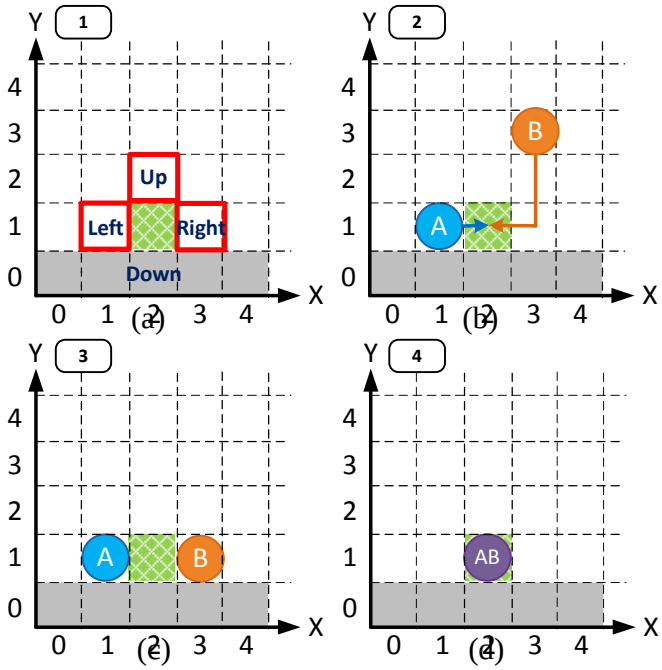


Figure 4.4 (a) Three available temporary destinations for two droplets. (b) to (d) The merging procedure by the proposed approach.

Table 4.2 Temporary Destinations Decision Rules

Up & Down	**Left & Right**	**Up & Down & Left & Right**	**Others**
Calculate the paths from droplet to Up and Down positions. Pick the closer one.	Calculate the paths from droplet to Left and Right positions. Pick the closer one.	Calculate the paths from droplet to Left, Right, Up and Down positions. Pick the closest one.	Merging Violation
Set the temporary destination for the other	Set the temporary destination for the other	Set the temporary destination for the other	

Block setting algorithm (Algorithm 3) for 3-pin nets requires special care. As illustrated in Figure 4.5, the first droplet with higher priority is routed to its temporary destination first and generates blocks (shaded area) at this position according to the bounding box. This approach lets the droplet wait for the second droplet to arrive at its own temporary destination without the interruption of irrelevant droplets. When both droplets arrives their temporary destinations, the blocks generated by two droplets around their own temporary destinations are trimmed according to the longer arrival time of two droplets. The two droplets are routed to their common sink at the next time step, which generates a block around the sink. For example, in Figure 4.5, d_A is routed to its temporary destination first at time t = 2 as its priority is higher than d_B, and then blocks on the temporary destination of d_A are generated till the end of experiment. Once d_B is routed to its own temporary destination at time t = 3, the longer arrival time A between d_A and d_B is 3 (Algorithm 3, Line 1). Then the blocks generated on the temporary destination of d_A are modified by trimming the blocks from time t = 3 (Algorithm 3,

Line 2). Finally blocks are generated on the common sink of d_A and d_B till the end of experiment (Algorithm 3, Line 3).

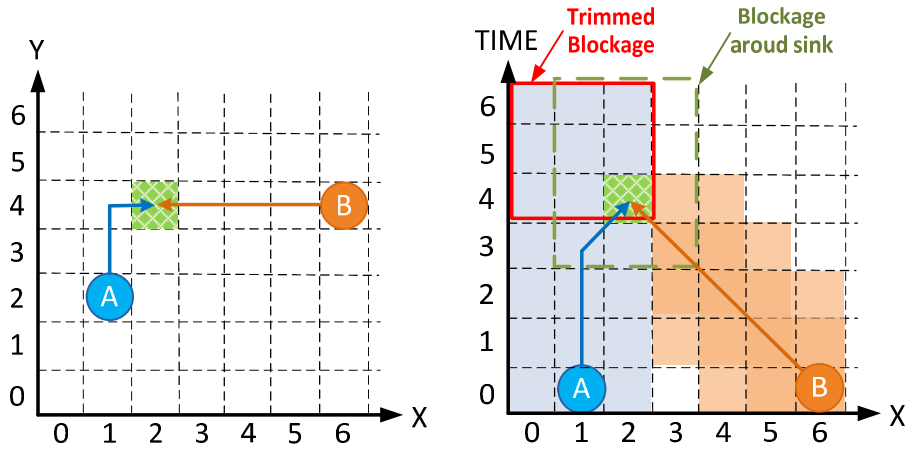


Figure 4.5 The block trimming procedure for 3-pin net, d_A will wait until d_B arrives at its own temporary destination for 1 time step.

Algorithm 3 MarkBlockForThreePin

1 $A \leftarrow \max\{A_{d_i,\hat{T}_{d_i}}, A_{\hat{d}_i,\hat{T}_{\hat{d}_i}}\}$

2 $G \leftarrow$ remove $B_{\hat{T}_{d_i}}$ and $B_{\hat{T}_{\hat{d}_i}}$ from A to $\overline{A}$

3 $G \leftarrow$ generate $B_{T_{d_i}}$ from A to $\overline{A}$

4.2.3 Dynamic Routing

Algorithm 4 shows the dynamic routing approach, in which, the backslash symbol means removing a specific priority indicator from the priority sequence. The initial setting of priorities set by the priority solver does not always solve every sub-problem. To solve this problem, a dynamic routing approach is introduced to adjust the priority during routing. If a net cannot be routed, which implies it is blocked by higher priority

nets, this net is set to the highest priority. On the other hand, if a net with the highest priority cannot be routed, the net is set to lowest priority in order to route other droplets.

Algorithm 4 FindPathAndMarkBlock

1 $P_{d_i} \leftarrow$ find path for d_i on G using 3D A* search algorithm
2 if $P_{d_i} \neq \phi$
3 $G \leftarrow$ MarkBlockForTwoPin(P_{d_i} , G)
4 else
5 $\sigma = (i \neq 1 ? \{\pi_i, \sigma \setminus \pi_i\} : \{\sigma \setminus \pi_1, \pi_1\})$
6 break
7 endif

4.2.4 Movement Control

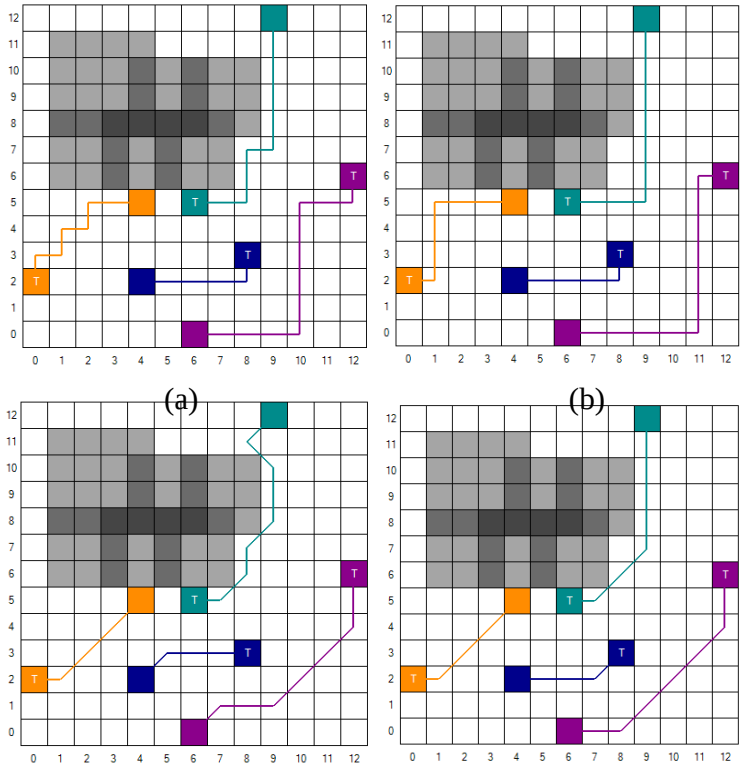
The router also features a droplet movement control policy to set the trend of path and moving directions. Diagonal routing can be turned on for droplets to move diagonally. A feature called “straightness-prone”, which means droplets tend to move to the same direction as the direction at the previous time step, is also implemented in the routing algorithm.

Zigzag movement: As shown in Figure 4.6(a), with the diagonal movement and straightness-prone features turned off, droplets move in zigzag pattern to mimic the diagonal movement for traditional DFMB devices that do not allow diagonal movements.

Diagonal movement: Movements in four diagonal directions are allowed in 3D-A* search algorithm with this feature turned on. Figure 4.6(c) shows an example of a diagonal movement without straightness-prone for a droplet.

Straightness-prone: This parameter can be used together with either one of the above two movements. Extra cost will be added when there is a change of moving

direction. Figure 4.6(b) and Figure 4.6(d) shows the paths of droplet movements with this parameter disabled and enabled, respectively.



(a) (b)

(c) (d)

Figure 4.6 (a) Diagonal movement disabled, Straightness-prone disabled. (b) Diagonal movement disabled, Straightness-prone enabled. (c) Diagonal movement enabled, Straightness-prone disabled. (d) Diagonal movement enabled, Straightness-prone enabled.

4.2.5 Channel-Based Routing

Thanks to the high configurability of MEDA, an innovative channel-based routing approach is made possible by moving a droplet from its source to a sink through a virtual narrow channel. This approach is able to handle some cases which are unroutable on traditional DMFBs and therefore effectively avoids the cross contamination as well as reduces total number of used cells.

Figure 4.7 shows that the sinks of two droplets are designated to sources of the other droplet. The droplets will collide with each other on traditional DMFBs. However, in MEDA DMFB, by enabling the channel-based routing, two droplets are squeezed into two narrow channels and able to reach their sinks without any interference.

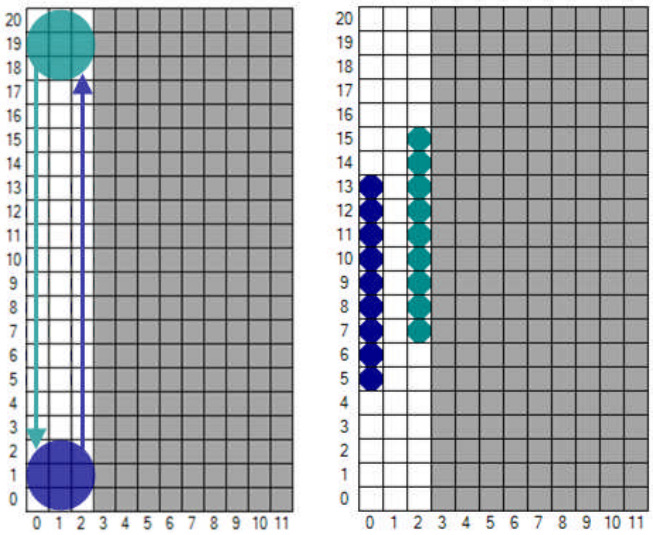


Figure 4.7 A sub problem is only routable by channel-based routing in MEDA.

In a channel-based routing algorithm, besides finding path for a droplet based on its reference point, a certain number of time steps need to be reserved for the "tail" of droplet, which is shown as the marked area in Figure 4.8. The number of reserved time steps t^{res} for a microelectrode is calculated by the following equation:

$$t^{res} = \lceil S_d / S_{ch} \rceil \tag{4.1}$$

in which, S_d is the area of a droplet, and S_{ch} is the area that a droplet can move in one time step. For example, for a 2×2 droplet in Figure 4.8, S_d is 4. If it is assumed that the droplet can only move for 1 microelectrode in one time step (i.e. $S_{ch} = 1$), the reserved time steps for microelectrode are 4 according to Eq. (4.1).

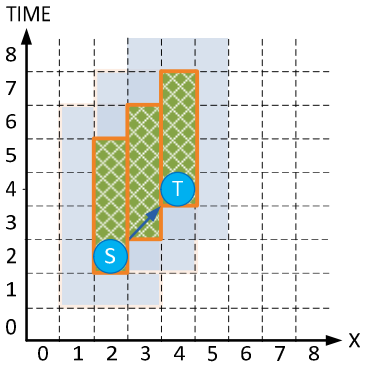


Figure 4.8 Green shaded grid represents the reserved time steps for the "tail" of a 2×2 droplet.

Chapter 5 Implementation of 3D Dynamic-Block-Based Droplet Router

The proposed DMFB droplet router is implemented in C# using Microsoft Visual Studio 2010 integrated development environment (IDE). Figure 5.1 shows the overview of the software architecture. The design of the software architecture follows three layers: data, logic and presentation. This chapter describes the implementation from these three aspects, focusing on the implementation of 3D dynamic-block-based droplet routing algorithm. The conversion of test cases from traditional DMFB architecture to MEDA architecture is then presented. The principles and implementation of the A* search algorithm are also discussed in this chapter.

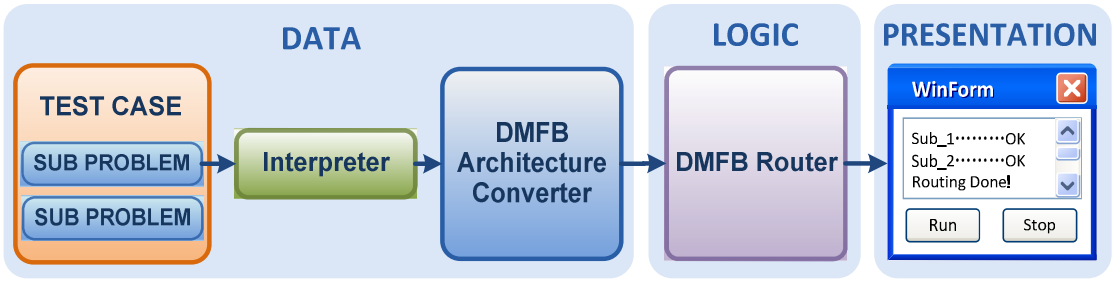


Figure 5.1 Overview of the software architecture.

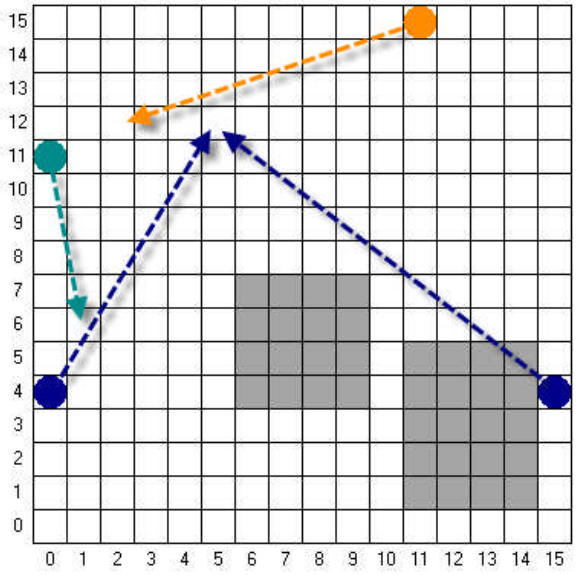
5.1 Data Layer

In this layer, the experiment description files are interpreted into the objects used in object-oriented (OO) language. The following contents present the syntax and structure of the input description file, the design of interpreter and the output of this layer.

1. Input description file

Figure 5.2(a) shows a segment of the input description file (in-virto-1, Sub Problem 2), in which the descriptions are interpreted line by line. The size of a DMFB electrode array is defined by ARRAY; the time constraint of each sub problems is defined by TIME and TIME CONSTRAINTS (TIME always equals to TIMECONSTRAINT + 1); and the number of sub problems is defined by NUMSUBPROBLEM. Each sub problem description section is wrapped between BEGIN SUBPROBLEM and END SUBPROBLEM, in which numbers of blocks and nets are defined by NUMBLOCKS and NUMNETS, respectively. The block position and size are defined after BLOCK.

```
ARRAY: 16 16
TIME: 21
TIMECONSTRAINT: 20
NUMSUBPROBLEMS: 11
...
BEGIN SUBPROBLEM 2
NUMBLOCKS: 2
BLOCK M1
6 4 9 7
BLOCK Dl2
11 1 14 5
NUMNETS: 3
NET S2_R2_M2
NUMPINS: 3
PIN S2
0 4
PIN R2
15 4
PIN M2_1
5 12
NET S1_Dl1
NUMPINS: 2
PIN S1
0 11
PIN Dl1_1
1 5
NET B_Dl1
NUMPINS: 2
PIN B
11 15
PIN Dl1_2
2 12
END SUBPROBLEM 2
...
```



(a) (b)

Figure 5.2 (a) A segment of the input description file. (b) visualization of sub problem 2.

Nets need special care as there are two kinds of nets: 2-pin and 3-pin, which are defined by NUMPINS. The source and the sink are defined by two consecutive PINs respectively (i.e., the first PIN is source; the second PIN is sink). For a 3-pin net which describes two droplets sharing a common sink, the first and second PINs describe the sources of two droplets, and the third PIN defines the position of the common sink.

Figure 5.2 shows that 11 sub problems need to be solved on a 16×16 DMFB assay for an in-vitro experiment. Figure 5.2(b) shows the visualization of the interpreted sub problem 2.

2. Interpreter Design

The interpreter works as a bridge between description files and objects, which converts plain-text descriptions into the objects used by OO language. Figure 5.3 shows the procedure of interpreting the experiment description into objects. As shown in Figure 5.4, the enumerator KEYWORDS_ENUM is a keyword library that defines all the syntaxes used for the experiment description. The interpreter reads the input file line by line, and once it detects any keyword defined in KEYWORDS_ENUM appears in the current line, the corresponding object is created and added into the root object TestBenchObject. As long as the interpreter reaches the end of description file, it returns TestBenchObject that contains all the information of an experiment.

```
While (Not InputFile.Eof )
If (InputFile.CurrentLine.Contain(KEYWORD))
        TestBenchObject = CreateObject(InputFile.CurrentLine, TestBenchObject)
InputFile.MoveNext
EndWhile
Return TestBenchObject
```

Figure 5.3 Procedure of interpreting descriptions to objects.

3. Output objects

Five classes are designed to model DMFB experiments. The class TestBench describes the properties of a DMFB array which also includes all the sub problems to be

solved. The class SubProblem contains all the nets and blocks information inside a subproblem. The relationship between these classes is shown in Figure 5.4.

Two major components in a DMFB experiment are DMFB electrode array and droplets. As shown in Figure 5.5, Planar and Droplet classes are defined to describe DMFB array and droplets, respectively. It should be noted that Planar is a class that is equipped with an electrode array and a time array, and the availability of each cell at a specific time can be requested from its instance. For example, when a droplet intends to move to a specific location at certain time, the routing algorithm should check the availability of the cell at this location to ensure the droplet can use this cell at next time step by requesting the Planar instance. The instance of Planar is created from the array information in the TestBench object by AssayConverter. Also all the SubProblem instances contained in a TestBench object are converted to a number of Droplet objects.

Droplet is a class that defines the properties of a droplet and its designated operation. IsDisgardable is set true if the sink of a droplet is a waste reservoir; IsForMixing is set true if two droplets share one common sink (i.e., 3-pin net); AllowMiniDroplet is set true if channel-based routing is permitted. And the moving area of fluidic sample in channel in one time step is defined by MinidropletWidth and MinidropletHeight. The routed path for each droplet is saved in RoutedPath.

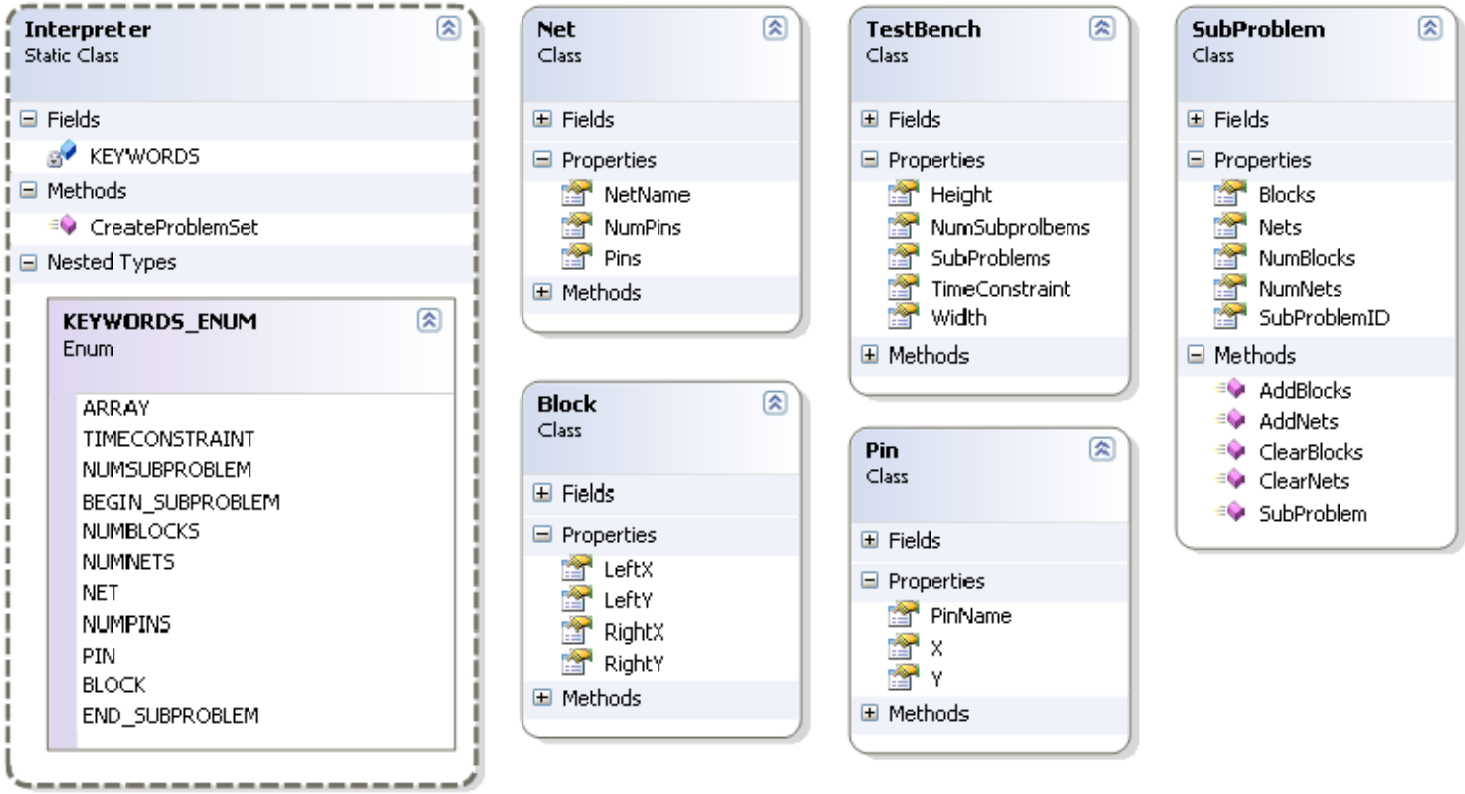


Figure 5.4 Classes defined to model DMFB experiment.

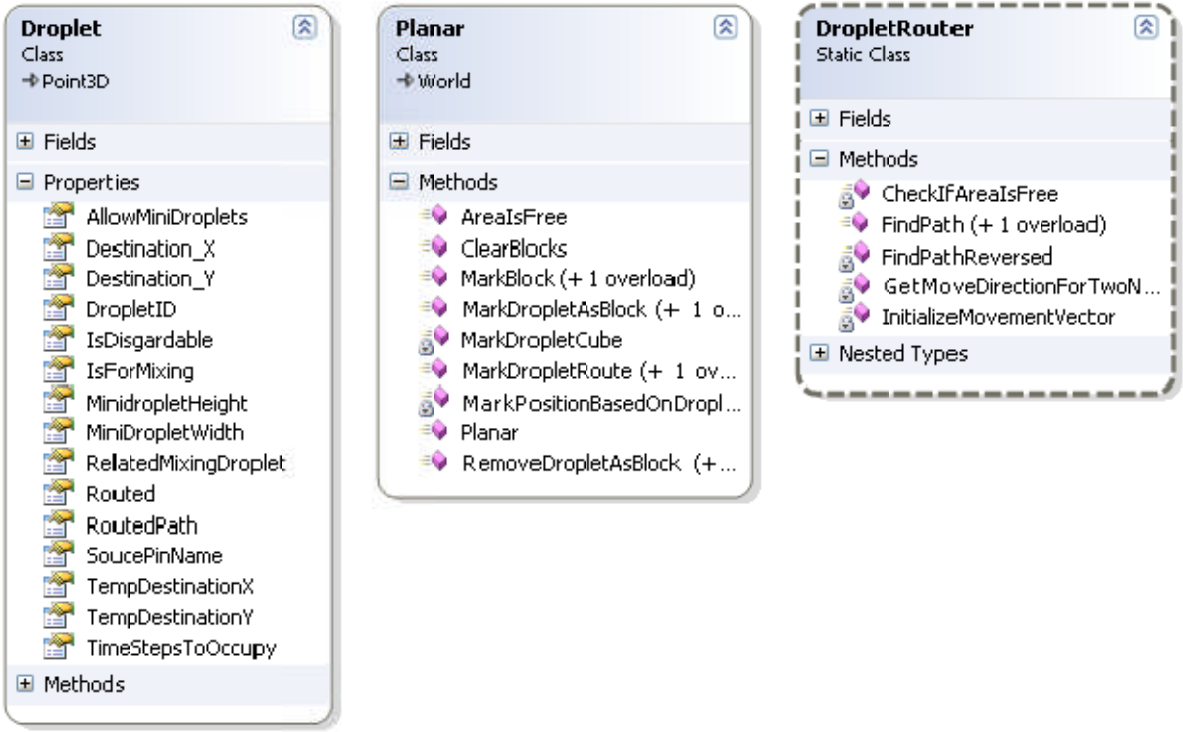


Figure 5.5 Classes defined for realistic description of DMFB experiment.

5.2 Logic Layer

This layer consists of priority solver and path finder. Priority solver organizes the routing sequence of nets by analyzing their features. The sorted nets are then passed to path finder.

The A* search algorithm [36], which is widely used in path finding problems, is utilized to find a path for each droplet. The A* search algorithm is an extension of Dijkstra's algorithm with heuristic feature and improved performance. Assume that a droplet needs to be moved from point A to point B as shown in Figure 5.6. The workflow of A* search algorithm is described as below [37]:

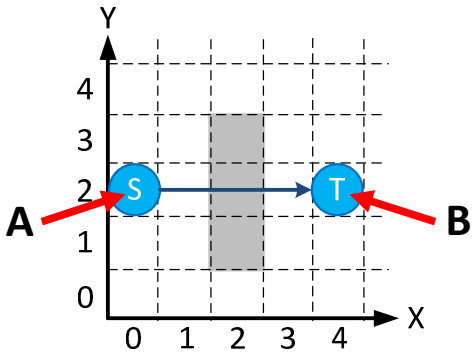


Figure 5.6 A droplet to be moved from point A to point B.

Step 1: Two data arrays *open list* and *close list* are defined. The *open list* contains all the points to be processed while *close list* contains the points have been processed. Starting with a point A, the program adds A to an *open list* and checks all the reachable points around point A by ignoring blocks. The program then links the parents of these reachable points to point A. Then point A is removed from *open list* and added to *close list*.

Step 2: The program evaluates the cost of all newly added points in the *open list* according to the following equation:

$$F=G+H \tag{5.1}$$

where *G* is the cost of moving droplet from point A to the given point and *H* is the estimated cost of moving from the given point to point B. Usually it is calculated by the Manhattan distance between these two points. The point with the minimum F (marked as current point) is removed from *open list* and added to *close list*.

Step 3: The program checks all the points around the current point, by ignoring blocks and points in the *close list*. If any points are not in the *open list*, their parents of these points are linked to the current point. For the points already in *open list*, their parents will be linked to the current point if the G cost of moving from current point to the point is less than their original G cost.

Step 4: Go to **Step 2**, until the destination point B is added to *close list*.

In the proposed routing algorithm, an extra axis is created to record time steps, thus A* search algorithm is extended to 3D space [34]. The number of nearby cells to search is increased from 8 to 26.

5.3 Presentation Layer

1. PlanarPicBox customized component

PlanarPicBox is a customized component which is inherited from Windows.Form.PictureBox class. It is designed specifically to show the status of droplets on a DMFB using GDI+. As shown in Figure 5.7, the routed paths and locations of

droplets can be illustrated by this component. Also the simulation of experiments can be visualized.

2. Main user interface

As shown in Figure 5.7, the main user interface (UI) consists of two panels: the left part shows the details of droplets in current sub problem, which includes droplet name, source, destination (sink), width, height, bounding box size, merging or not, and disposability; the right part shows the visualization of the current sub problem and contains the control panel. The control panel beneath the PlanarPicBox is used to choose the sub problem to be solved. The *simulate* button calls for the simulation window to visualize the movement of droplets in the current routed sub problem.

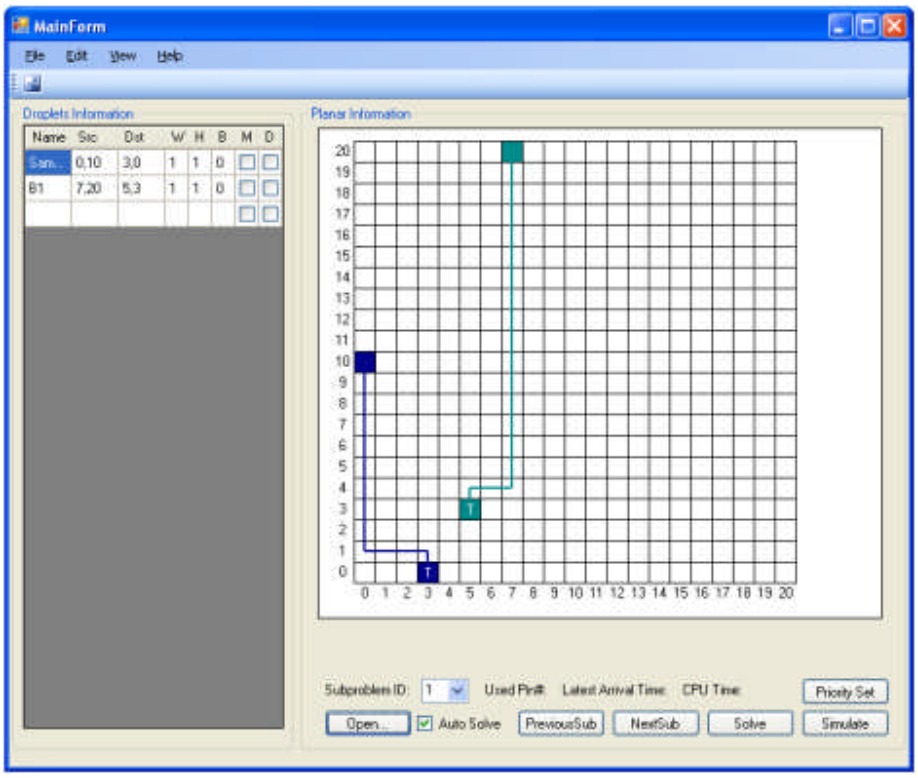


Figure 5.7 Screenshot of the main user interface, the status of droplets and electrode array is shown in PlanarPicBox.

3. Simulation platform

Once a sub problem is routed successfully, the movement of droplets is able to be visualized by the simulation platform. The *rewind* button navigates to the first frame of animation while the forward button jumps to the last frame. The *play* button can play or pause the animation. The scroll bar at the right of PlanarPicBox allows the user to navigate to any frame by dragging it. The speed control bar can control the speed of animation.

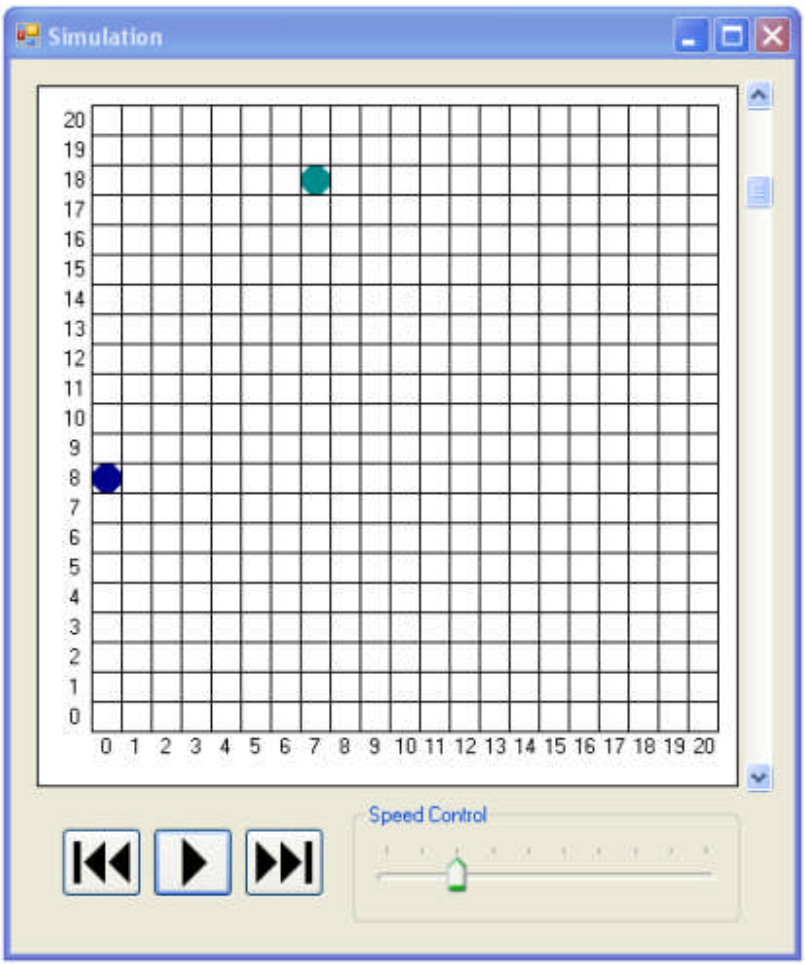


Figure 5.8 Screenshot of simulation platform.

5.4 Adjustable Parameters

In order to investigate the effect of different paths on routing performance, it is preferable to have parameters adjustable in the droplet routing algorithm. Three parameters, including *straightness-prone*, *diagonal movement* and *used cells preference*, are investigated. These three parameters affect the paths of droplets and, in turn, the routing performance.

1. Straightness-prone

When straightness-prone is turned on, the droplet to move towards the same direction as the previous time step. While A* search algorithm is evaluating the nearby cells, if the direction from current cell to reachable cell is not same as the previous direction, an extra cost will be added to the reachable cell in the cost function.

2. Diagonal Movement

Some recent DMFBs allow droplets to move diagonally. In traditional DMFBs, only 5 cells would be evaluated while path finding algorithm is searching reachable cells (i.e., Left, right, up, down and current cell). As shown in Figure 5.9, if diagonal movement is enabled, 4 extra cells will be evaluated (i.e., Left-up, right-up, left-down, and right-down).

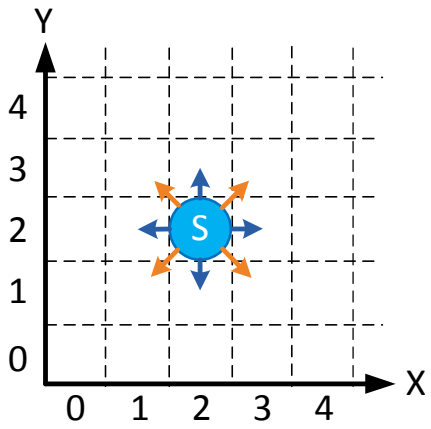


Figure 5.9 Four extra cells are checked in MEDA, which are marked as orange diagonal arrows.

3. Used Cells Preference

This parameter encourages droplets to reuse the used cells. A binary 2D bitmap is created to record the cells that were used by other droplets. During routing, if a reachable cell has not been used by other droplets, an extra cost will be added to this cell. Once a net has been routed, all the cells along its path will be marked as used on the bitmap.

5.5 Traditional DMFB to MEDA Conversion

This procedure is included in AssayConverter and can be enabled if necessary. Droplet width and height are customizable by setting the corresponding properties in the instance. In this thesis work, three sizes of droplets in MEDA are tested: 3×3, 5×5 and 7×7. For fair comparison, the test cases used in traditional DMFBs need to be converted to equivalent test cases for MEDA. Assuming that the droplet size is $M \times N$, the conversion follows Eq. (5.1) – (5.7), and Table 5.1 shows the notion used in MEDA conversion:

Table 5.1 Notation used in MEDA conversion.

t_{type}^{c}	Time constraint of an experiment
$\beta_{type}^{x/y}$	Left bound of block
$\gamma_{type}^{x/y}$	Right bound block
$\eta_{type}^{x/y}$	Pin position

$$t_{MEDA}^{c} = t_{Traditional}^{c} \times M \times N \tag{5.2}$$

$$\beta_{MEDA}^{x} = \beta_{Tradition}^{x} \times M \tag{5.3}$$

$$\beta_{MEDA}^{y} = \beta_{Tradition}^{y} \times N \tag{5.4}$$

$$\gamma_{MEDA}^{x} = (\gamma_{Tradition}^{x} + 1) \times M - 1 \tag{5.5}$$

$$\gamma_{MEDA}^{y} = (\gamma_{Tradition}^{y} + 1) \times N - 1 \tag{5.6}$$

$$\eta_{MEDA}^{x} = \eta_{Tradition}^{x} \times M \tag{5.7}$$

$$\eta_{MEDA}^{y} = \eta_{Tradition}^{y} \times N \tag{5.8}$$

Figure 5.10 shows a sub problem which is converted from a traditional DMFB to a 3×3 MEDA. The electrode array, modules and droplets are all increased by a factor of 3.

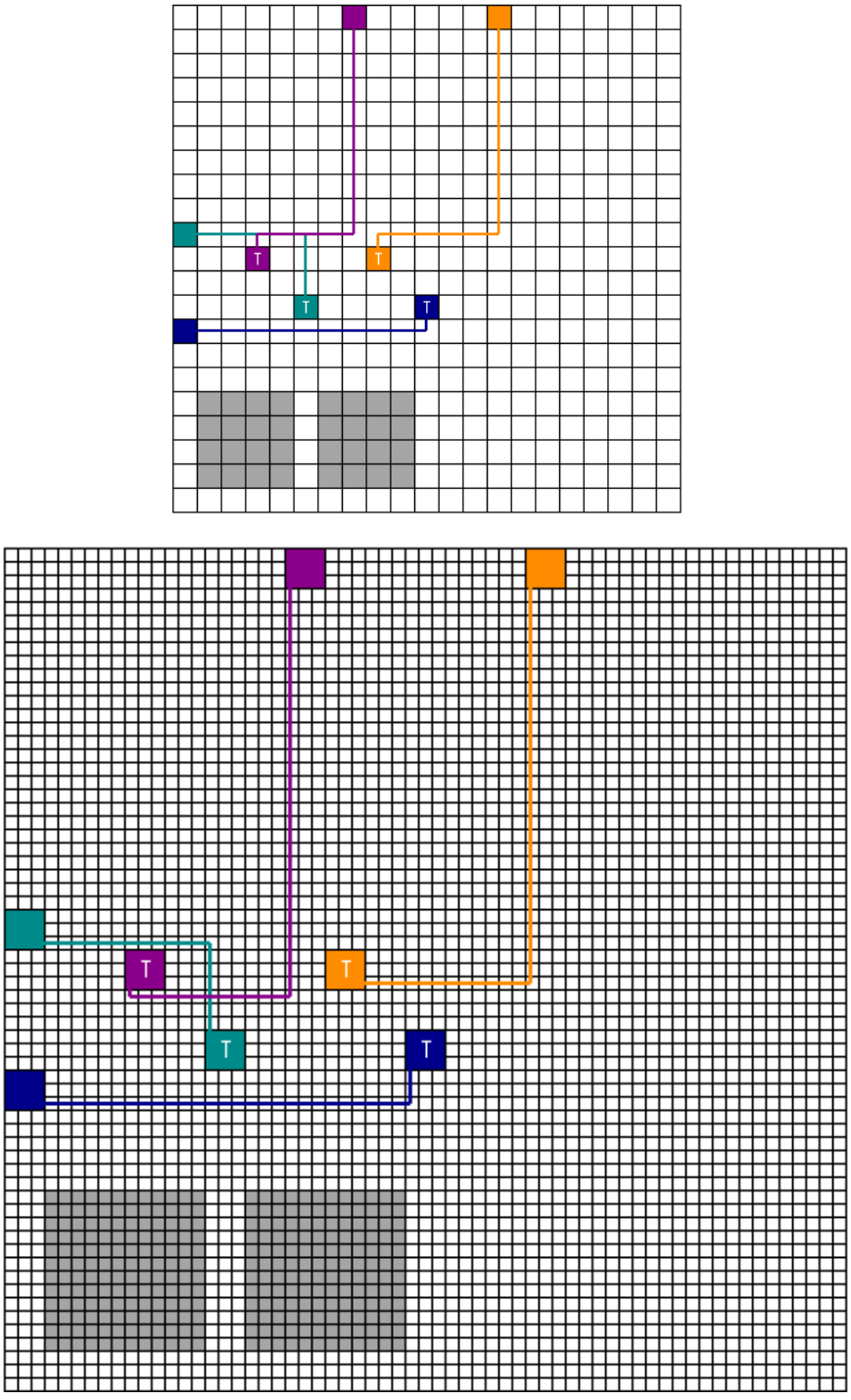


Figure 5.10 A sub problem in traditional DMFB and 3×3 MEDA architecture

Chapter 6 Results and Performance Comparison

To evaluate the performance of 3D dynamic-block-based droplet routing algorithm, a set of test cases were simulated with different combinations of adjustable parameters including different priority solvers. The effects of the parameters are investigated in terms of maximum latest arrival time, average latest arrival time and total number of used cells. Then the results with optimal parameters settings are compared with other existing routing algorithms. The conversion equation is given for fair comparison between traditional DMFBs and MEDA.

6.1 Conversion of Equivalent Results

To make fair comparison with traditional DMFBs, the results of MEDA are converted to equivalent results using Eq. (6.1-6.3). Table 6.1 shows the notation used in conversion equation. To simplify the routing problem, it is assumed that the width and height of droplet are both equal to W (i.e. $M=N=W$):

Table 6.1 Notation used in conversion equations.

t_{org}^{l}	Original maximum latest arrival time
t_{org}^{acc}	Original accumulative latest arrival time
Ω_{org}	Original total number of used cells
t_{eqv}^{l}	Equivalent maximum latest arrival time
t_{eqv}^{acc}	Equivalent accumulative latest arrival time
Ω_{eqv}	Equivalent original total number of used cells
W	Width or height of the droplet

Figure 6.1 shows the equivalent time steps and total number of used cells in a 3 ×3 channel-based MEDA DMFB. The reference point of droplet is used to indicate the current position of droplet. After the reference point has arrived at the sink, the "tail" of the droplet should also be moved to the sink following the path of reference point. The original time steps can be calculated as $t_{org}^{l} = t_{eqv}^{l} \cdot W + W^2 - 1$. For example, if a 3×3 droplet in MEDA moves for 2 equivalent time steps, the actual (original) time steps used in MEDA are $2\times3+3^2-1=14$. Inferred from the above equation, for channel-based routing results, equivalent maximum latest arrival time and accumulative latest arrival times are calculated by Eq. (6.1) and Eq. (6.2); the equivalent total number of used cells is calculated by Eq. (6.3).

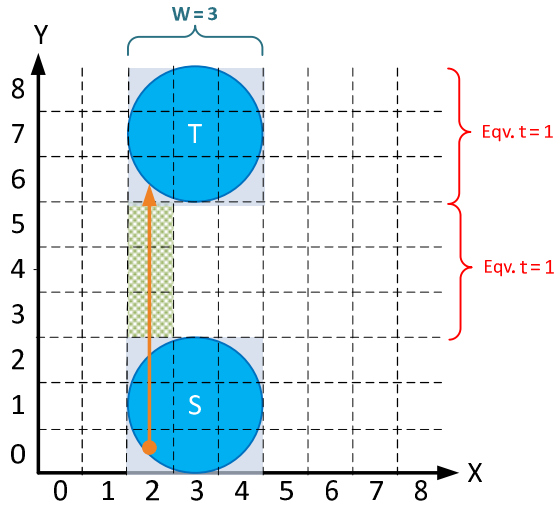


Figure 6.1 The equivalent time steps and original total number of used cells (green shaded grid) in a 3 ×3 channel-based MEDA DMFB

$$t_{eqv}^{l} = (t_{org}^{l} + 1 - W^2) / W \tag{6.1}$$

$$t_{eqv}^{acc} = (t_{org}^{acc} + 1 - W^2) / W \tag{6.2}$$

$$\Omega_{eqv} = \Omega_{org} / W^2 \tag{6.3}$$

Figure 6.2 shows the equivalent time steps and total number of used cells in a 3 ×3 non-channel-based MEDA DMFB, the original time steps can be calculated as $t_{org}^{l} = t_{eqv}^{l} \cdot W$, inferred from which, for non-channel-based routing results, equivalent maximum latest arrival time and accumulative latest arrival times are calculated according to Eq. (6.4) and Eq. (6.5); the equivalent total number of used cells is calculated according to Eq. (6.6).

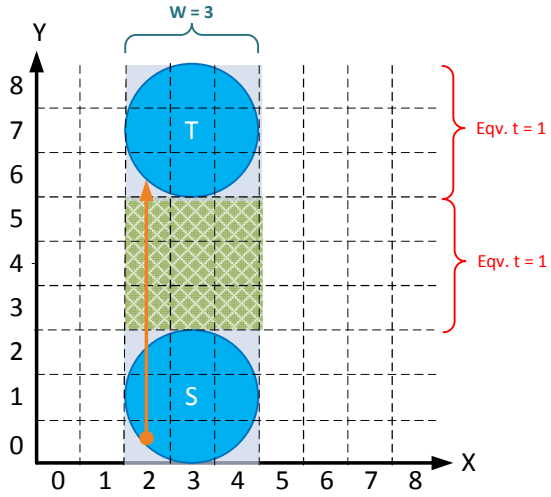


Figure 6.2 The equivalent time steps and original total number of used cells (green shaded grid) in a 3 ×3 non-channel-based MEDA DMFB

$$t_{eqv}^{l} = t_{org}^{l} / W \tag{6.4}$$

$$t_{eqv}^{acc} = t_{org}^{acc} / W \tag{6.5}$$

$$\Omega_{eqv} = \Omega_{org} / W^2 \tag{6.6}$$

6.2 Effect of Priority Solvers on Traditional DMFBs

Priority is an important factor for droplet routing algorithms. An inappropriate priority setting may cause lower priority droplets to be blocked by higher priority droplets, and in turn, may cause the failure of routing. Thus an appropriate priority setting is necessary to ensure the quality of routing. Both the fast routability and performance driven droplet routing algorithm [23] and the high performance droplet routing algorithm [21] implemented their own priority solvers and achieved high routability for the given benchmark problems. These two priority solvers (without considering concession zone)

are implemented and applied to the 3D-A* search algorithm. Also the same benchmark is used to make fair comparison with the proposed work. Table 6.2 shows the information of the benchmark suite which includes the size of DMFB, number of sub problems, number of nets and the time constraint of 4 test cases. Table 6.3 shows failed sub problems with different priority solvers. The number of failures is an indicator of the routability of 3D-A* search algorithm with different priority solvers. The path-based priority solver can achieve a better routability than those without using any priority solver. In most cases, the path-based priority solver (represented as "ours" in Table 6.3) combined with the priority solver of Cho's algorithm [21] results in the highest routability which can be seen from the total failures of different combinations of priority solvers in Table 6.3.

Table 6.2 Information of 4 test cases in benchmark suite III [23].

Benchmark Suite III				
Name	Size	#Sub	#Net	#T_{max}
in-vitro_1	16 × 16	11	28	20
in-vitro_2	14 × 14	15	35	20
protein_1	21 × 21	64	181	20
protein_2	13 × 13	78	178	20

Table 6.3 Failed sub problems with different priority solvers.

Test Case Name	None	Huang	Cho	Huang+Ours	Cho+Ours	Ours
5D, Straight						
Failures						
in-vitro_1	1	1	2	0	1	1
in-vitro_2	1	0	0	1	1	1
protein_1	4	5	6	5	5	4
protein_2	5	2	1	2	0	3
5D, No Straight						
Failures						
in-vitro_1	1	1	1	0	0	0
in-vitro_2	1	0	0	1	0	1
protein_1	4	5	6	5	5	4
protein_2	4	2	1	1	0	2
9D, Straight						
Failures						
in-vitro_1	0	1	0	2	1	1
in-vitro_2	1	0	0	1	0	1
protein_1	0	0	1	0	1	0
protein_2	3	2	2	2	2	2
9D, No Straight						
Failures						
in-vitro_1	0	1	0	2	1	1
in-vitro_2	1	0	0	1	0	1
protein_1	0	0	0	0	0	0
protein_2	4	2	3	2	2	2
Total	**30**	**22**	**23**	**25**	**19**	**24**

Table 6.3 indicates although the 3D-A* search algorithm does not always successfully route every test cases with these priority solvers; the path-based priority solver can improve the routability. Applying path-based priority solver onto the initial priority solver can reduce re-routing times in dynamic routing. Figure 6.3, Figure 6.4 and Figure 6.5 show t_{org}^{l}, t_{org}^{acc} and Ω_{org} of different priority solvers with dynamic routing enabled, diagonal movement and straightness-prone disabled. By comparing to the results of other combinations of priority solvers, it shows that path-based priority solver combined with dynamic routing results in the best t_{org}^{acc} and Ω_{org}, but does not work best in terms of t_{org}^{l}. The same conclusion can be made for other combinations of parameters such as the results of path-based priority solver with dynamic routing, diagonal movement and straightness-prone enabled.

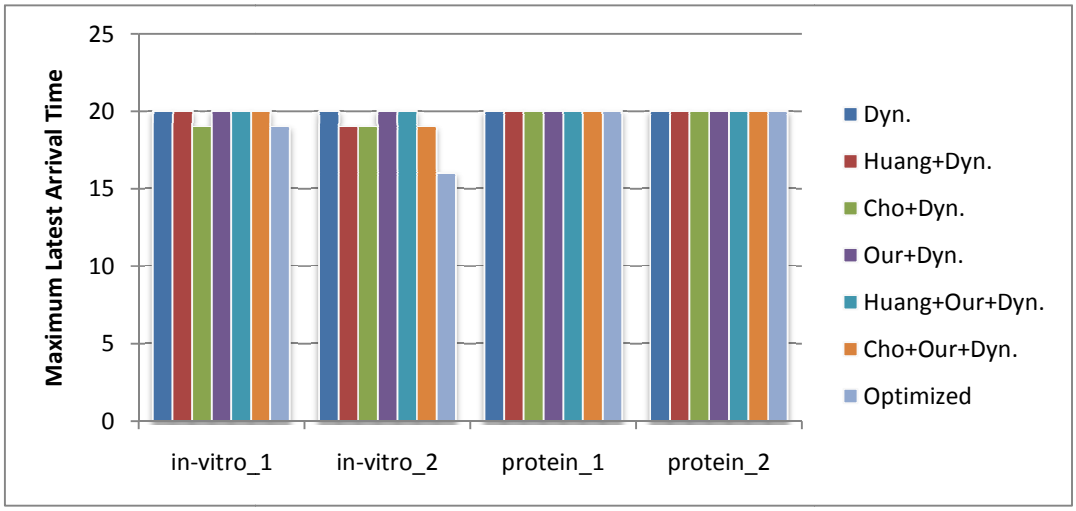


Figure 6.3 Maximum latest arrival time of different combinations of priority solvers.

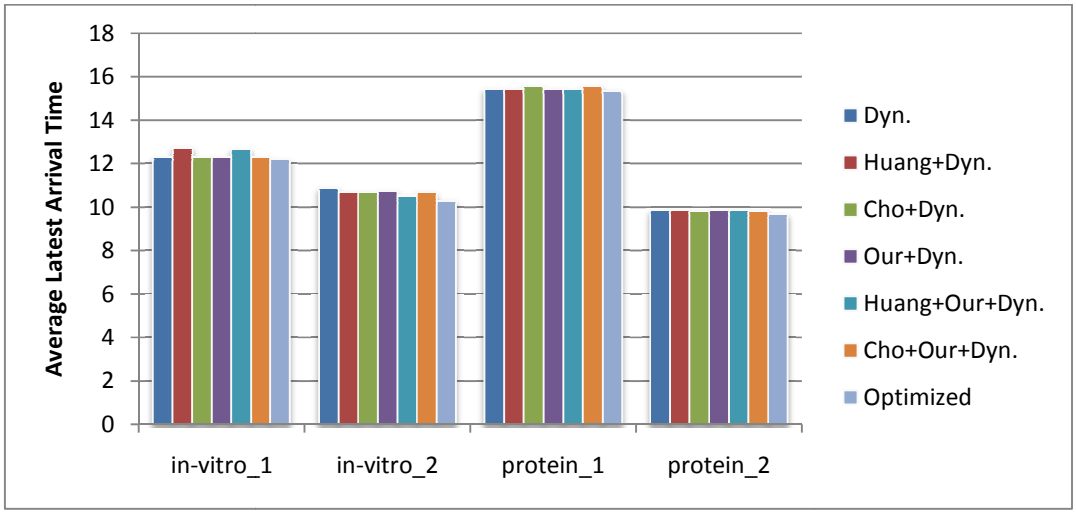


Figure 6.4 Average latest arrival time of different combinations of priority solvers.

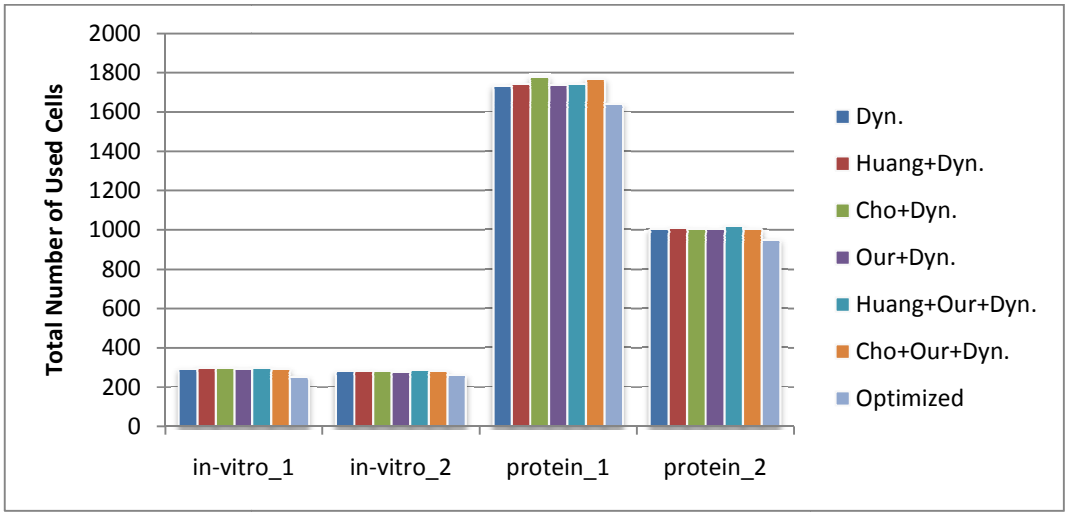


Figure 6.5 Total number of used cells of different combinations of priority solvers.

6.3 Effect of Straightness-Prone Parameter

The analysis of effect of straightness-prone parameter is based on the optimized results. The effect of straightness-prone on routability was evaluated by changing the switch of diagonal movement and straightness-prone with dynamic routing enabled. The abbreviations used in figures are show as follows:

- **N. D.:** No Diagonal movement
- **D.:** Diagonal movement
- **N.S.:** No Straightness-prone
- **S.:** Straightness-prone

Figure 6.6, Figure 6.7 and Figure 6.8 show t_{org}^{l} , t_{org}^{acc} and Ω_{org} by enabling or disabling of diagonal movement and straightness-prone parameter in traditional DMFB architecture. By comparing N.D., S. to N.D., N. S., and D., S. to D., N.S., it is concluded that for a traditional DMFB, straightness-prone can reduce t_{org}^{l} as well as t_{org}^{acc} from Figure 6.6 and Figure 6.7, but Ω_{org} will be increased by enabling this parameter in Figure 6.8.

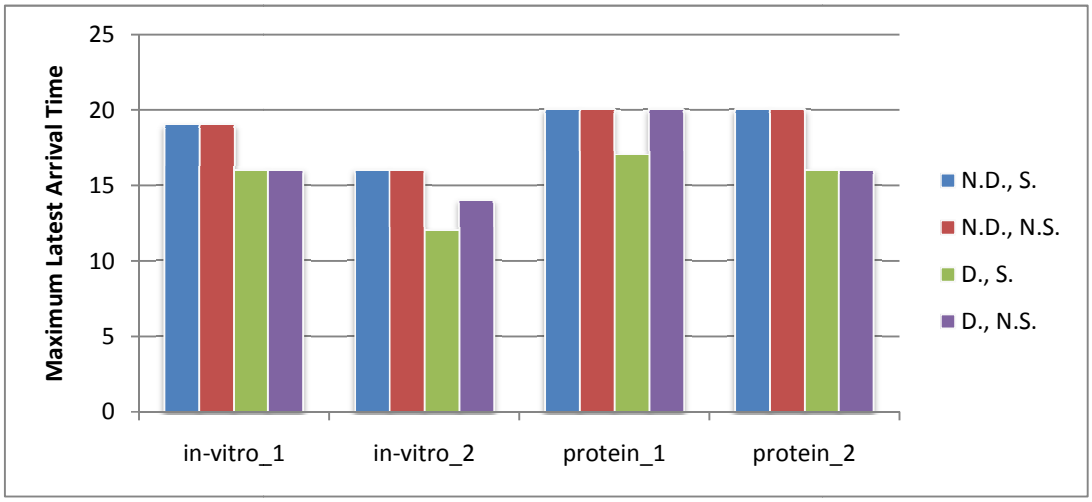


Figure 6.6 Maximum latest arrival time of optimized results.

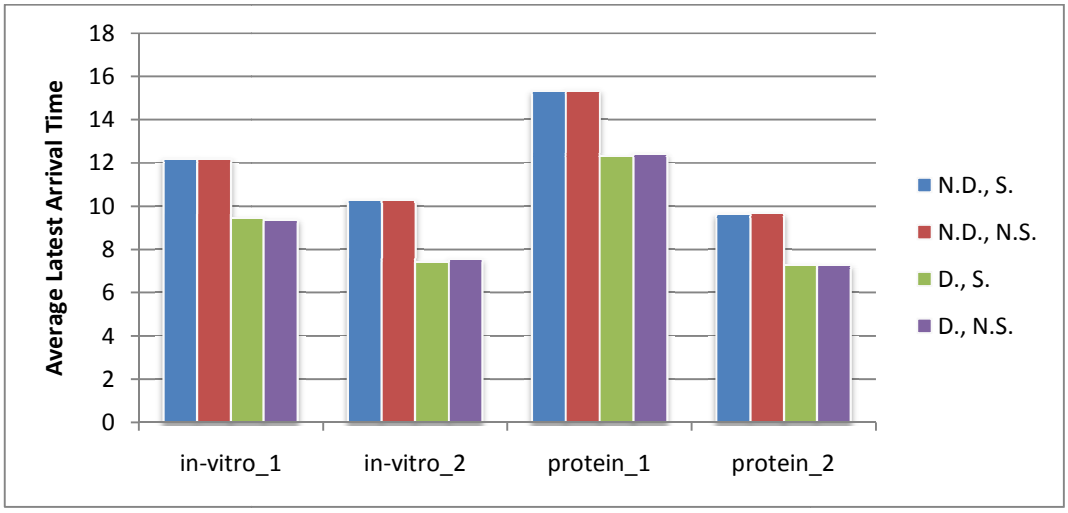


Figure 6.7 Average latest arrival time of optimized results.

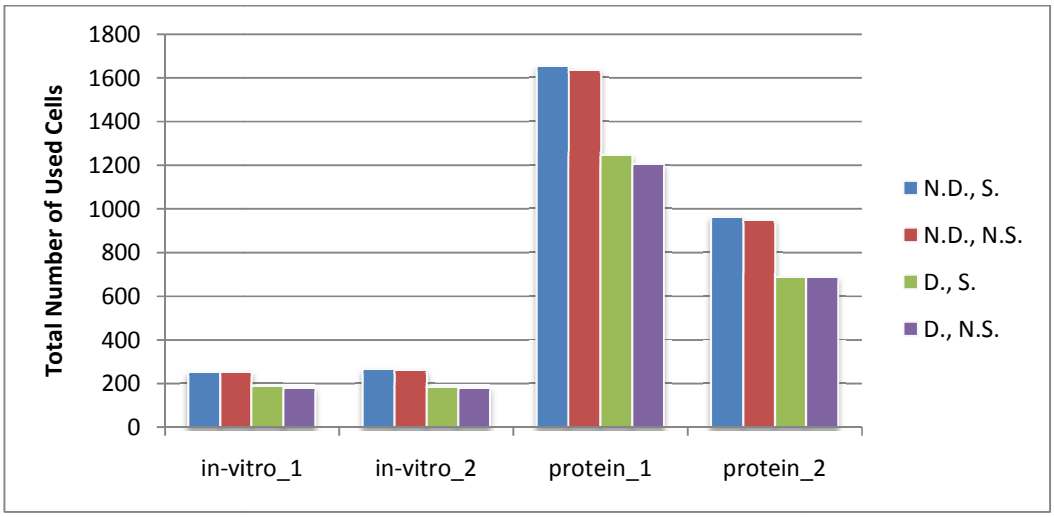


Figure 6.8 Total number of used cells of optimized results.

Figure 6.10-6.13 show the optimized routing results with different combinations of straightness-prone, diagonal movement and with used cells preference enabled in MEDA, while Figure 6.14-6.17 show the optimized results in MEDA without cell preference.

In MEDA, the overall conclusion is that t_{eqv}^{l} , t_{eqv}^{acc} and Ω_{eqv} will be increased by enabling straightness-prone parameter by investigating Figure 6.10-6.17 and Table 6.4. Although in Table 6.4 Ω_{eqv} gets slightly decreased by enabling this parameter, the overall

results show the increase of Ω_{eqv}. Two test cases failed when enabling straightness-prone parameters in 5×5 and 7×7 channel-based routing (See Figure 6.10 and Figure 6.14). This might be caused by the longer straight tail behind a droplet that blocks the way of other droplets.

Table 6.4 Examples of the effect of straightness-prone parameter.

Comparison Between	t_{eqv}^{l}		**Average** t_{eqv}^{acc}		Ω_{eqv}	
Figure 6.10 and Figure 6.11 (in-vitro-1, 3×3 No Channel, Used Cell Preference)	S., N.D.	N.S., N. D.	S., N.D.	N.S., N. D.	S., N.D.	N.S., N. D.
	17.33	17.33	11.79	11.79	243.33	249.33
Figure 6.12 and Figure 6.13 (in-vitro-1, 3×3 No Channel, Used Cell Preference)	S., D.	N.S., D.	S., D.	N.S., D.	S., D.	N.S., D.
	15.33	15.33	9.30	9.30	244.89	245.56
Figure 6.14 and Figure 6.15 (in-vitro-1, 3×3 No Channel, No Used Cell Preference)	S., N.D.	N.S., N.D.	S., N.D.	N.S., N.D.	S., N.D.	N.S., N.D.
	22.33	17.33	12.27	11.48	283.33	284.89
Figure 6.16 and Figure 6.17 (in-vitro-1, 3×3 No Channel, No Used Cell Preference)	S.,D.	N.S., D.	S.,D.	N.S., D.	S.,D.	N.S., D.
	15.33	15.33	9.15	9.12	252.67	259.67

6.4 Effect of Diagonal Movement

The analysis of effect of diagonal movement parameter is based on the optimized results. From Figure 6.6, Figure 6.7, Figure 6.8 by comparing N.D., S to D.,S, and N.D., N.S., to D.,N.S., it concludes that in traditional DMFB architecture, enabling diagonal movement can apparently improve t_{org}^{l}, t_{org}^{acc} and Ω_{org}. Also in MEDA, significant improvement can be seen in terms of t_{eqv}^{l}, t_{eqv}^{acc} and Ω_{eqv} by investigating Figure 6.10-6.17

and Table 6.5. Thus enabling diagonal movement is an efficient approach to take full advantage of DMFB.

Table 6.5 Examples of the effect of diagonal movement parameter.

Comparison Between	t_{eqv}^{l}		**Average** t_{eqv}^{acc}		Ω_{eqv}	
Figure 6.10 and Figure 6.12 (in-vitro-1, 3×3 No Channel)	S., N.D.	S., D.	S., N.D.	S., D.	S., N.D.	S., D.
	17.33	15.33	11.79	9.30	243.33	244.89
Figure 6.11 and Figure 6.13 (in-vitro-1, 3×3 No Channel, No Used Cell Preference)	N.S., N. D.	N.S., D.	N.S., N. D.	N.S., D.	N.S., N. D.	N.S., D.
	17.33	15.33	11.79	9.30	249.33	245.56
Figure 6.14 and Figure 6.16 (in-vitro-1, 3×3 No Channel, Used Cell Preference)	S., N.D.	S.,D.	S., N.D.	S.,D.	S., N.D.	S.,D.
	22.33	15.33	12.27	9.15	283.33	252.67
Figure 6.15 and Figure 6.17 (in-vitro-1, 3×3 No Channel, No Used Cell Preference)	N.S., N.D.	N.S., D.	N.S., N.D.	N.S., D.	N.S., N.D.	N.S., D.
	17.33	15.33	11.48	9.12	284.89	259.67

6.5 Effect of Used Cells Preference

The analysis of effect of used cells preference parameter is based on the optimized results. For both traditional and MEDA DMFB architectures, enabling this parameter will increase t_{eqv}^{l} and t_{eqv}^{acc} for a little bit while reduce Ω_{eqv} significantly by investigating Figure 6.10-6.17 and Table 6.6.

Table 6.6 Examples of the effect of used cells preference parameter.

Comparison Between	t_{eqv}^{l}		**Average** t_{eqv}^{acc}		Ω_{eqv}	
Figure 6.10 (With Used Cells Preference) and Figure 6.14 (Without Used Cells Preference) (in-vitro-1, 3×3 No Channel)	S., N.D.	S., N.D.	S., N.D.	S., N.D.	S., N.D.	S., N.D.
	17.33	22.33	11.79	12.27	243.33	283.33
Figure 6.11 (With Used Cells Preference) and Figure 6.15 (Without Used Cells Preference) (in-vitro-1, 3×3 No Channel, No Used Cell Preference)	N.S., N. D.	N.S., N.D.	N.S., N. D.	N.S., N.D.	N.S., N. D.	N.S., N.D.
	17.33	17.33	11.79	11.48	249.33	284.89
Figure 6.12 (With Used Cells Preference) and Figure 6.16 (Without Used Cells Preference) (in-vitro-1, 3×3 No Channel, Used Cell Preference)	S., D.	S.,D.	S., D.	S.,D.	S., D.	S.,D.
	15.33	15.33	9.30	9.15	244.89	252.67
Figure 6.13 (With Used Cells Preference) and Figure 6.17 (Without Used Cells Preference) (in-vitro-1, 3×3 No Channel, No Used Cell Preference)	N.S., D.	N.S., D.	N.S., D.	N.S., D.	N.S., D.	N.S., D.
	15.33	15.33	9.30	9.12	245.56	259.67

6.6 Effect of Channel-Based Routing for MEDA

By inner comparison of each individual figure from Figure 6.10 to Figure 6.17, the impacts of changing droplet sizes can be observed.

In non-channel-based routing, as the size of droplets increases, t_{eqv}^{l} and t_{eqv}^{acc} do not change too much, but Ω_{eqv} get increased.

In channel-based routing, t_{eqv}^{l} and t_{eqv}^{acc} get increased compared to non-channel-based routing, but Ω_{eqv} are dramatically reduced. As the size of droplets increases in channel-based routing, t_{eqv}^{l} and t_{eqv}^{acc} increases while Ω_{eqv} get further reduced. This phenomenon can be seen from the following example. As shown in Figure 6.9, two cases are investigated, in which, the droplet size of first one is 2×2 and the second one is 3×3. Both

of them move for two equivalent cells. According to Eq. (6.3), the equivalent used cells of the first case is 2/(2×2) = 0.5 while the second one is 3/(3×3)= 0.33. Thus the equivalent cells number decreases as the size of droplets increases.

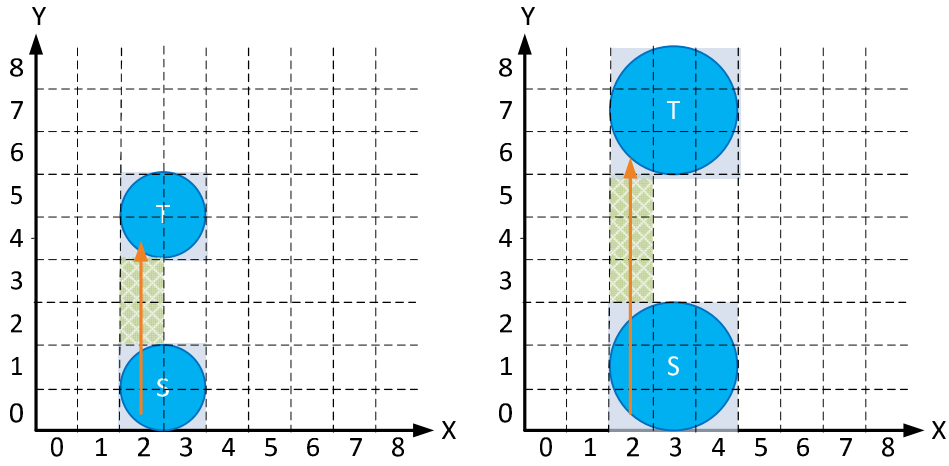
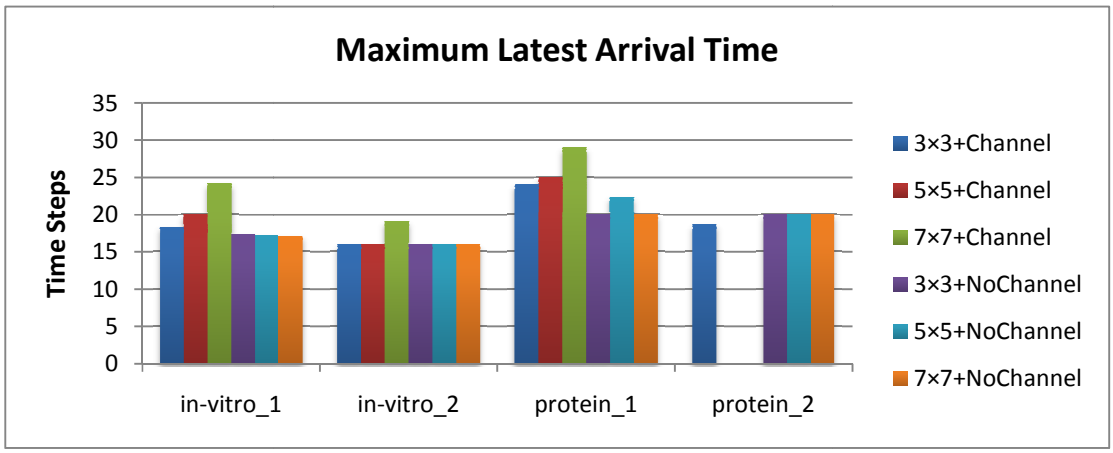
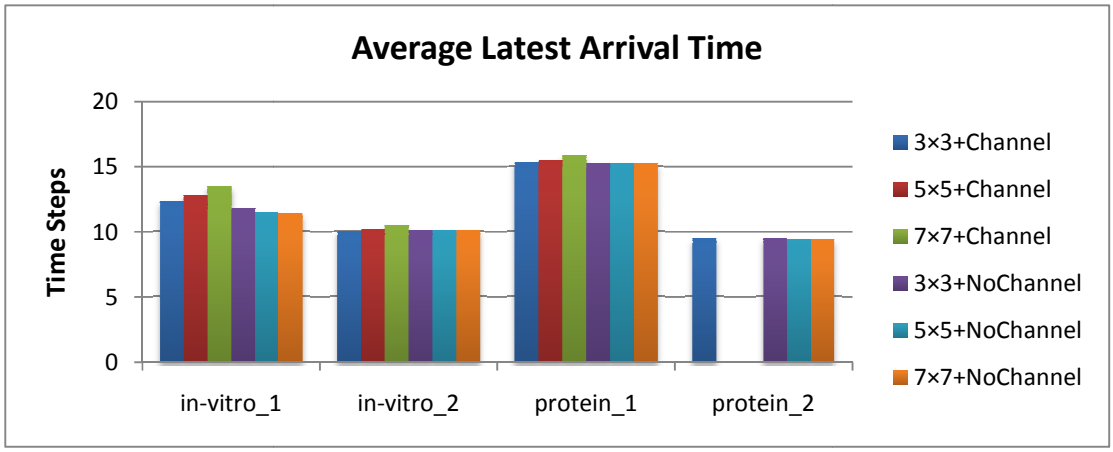


Figure 6.9 Two droplets with different sizes move for two equivalent cells, green shaded grid are counted as original used cells.

For some sub problems such as the one shown in Figure 4.7, enabling channel-based routing also can improve the routability. Since all the test cases used in this thesis work are converted from the test cases used in traditional DMFB architecture, it is unable to show the advantage of improved routability.





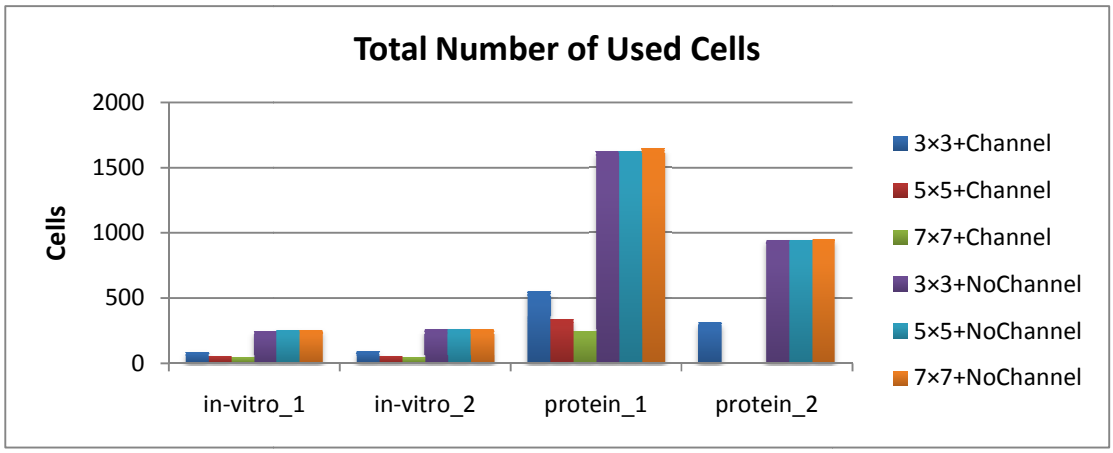


Figure 6.10 Straightness-prone, No diagonal movement, Used cells preference (MEDA).

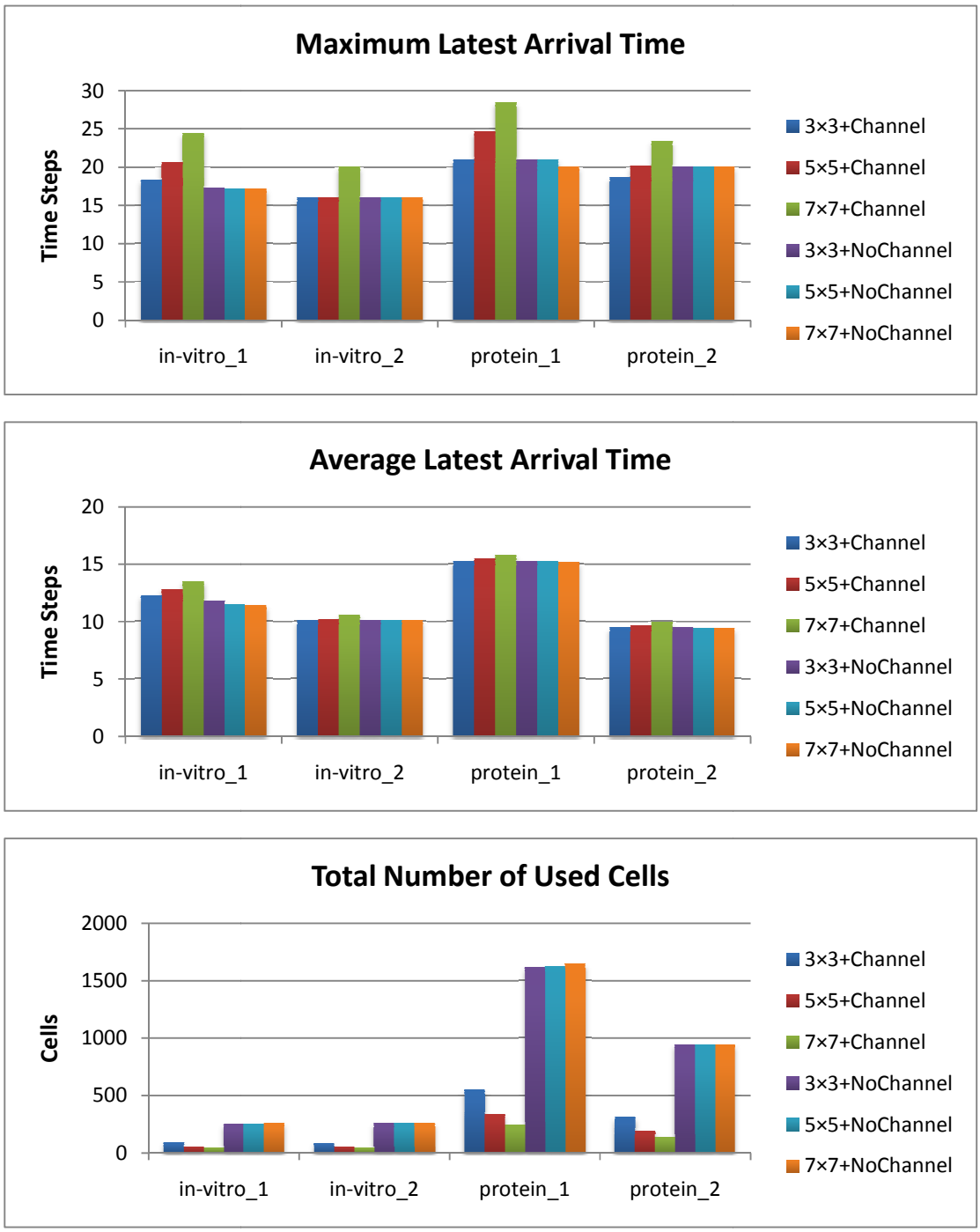


Figure 6.11 No Straigtness-prone, No Diagonal movement, Used cells preference (MEDA).

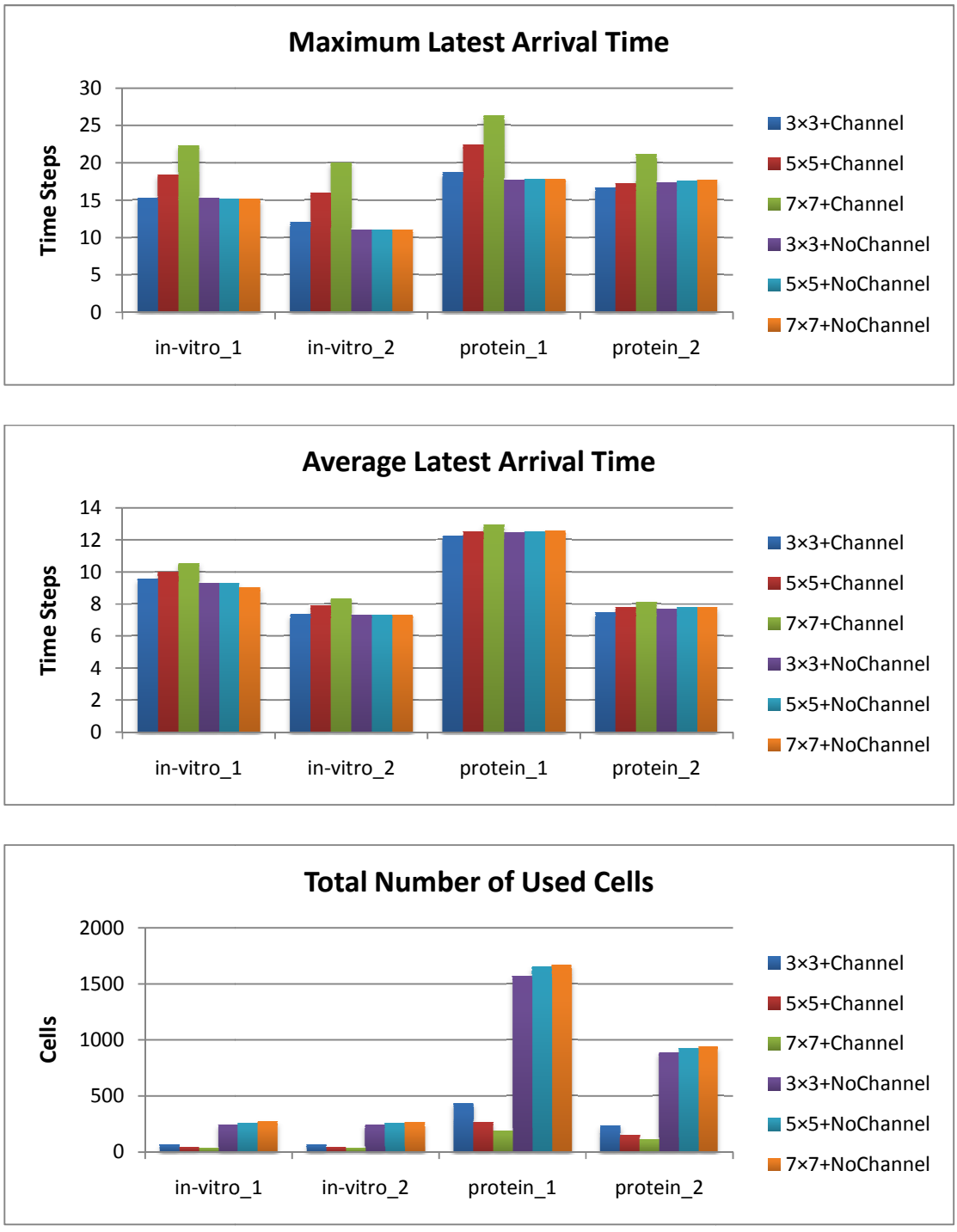
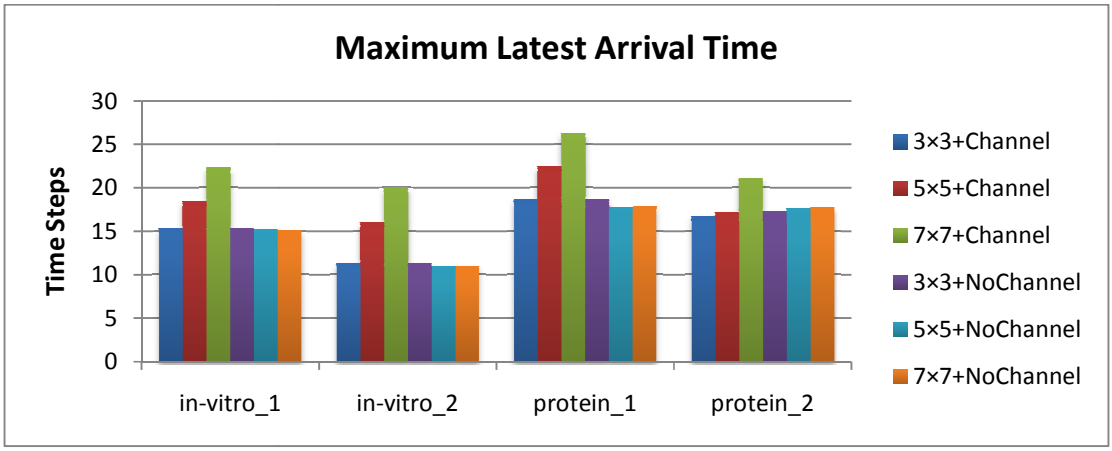
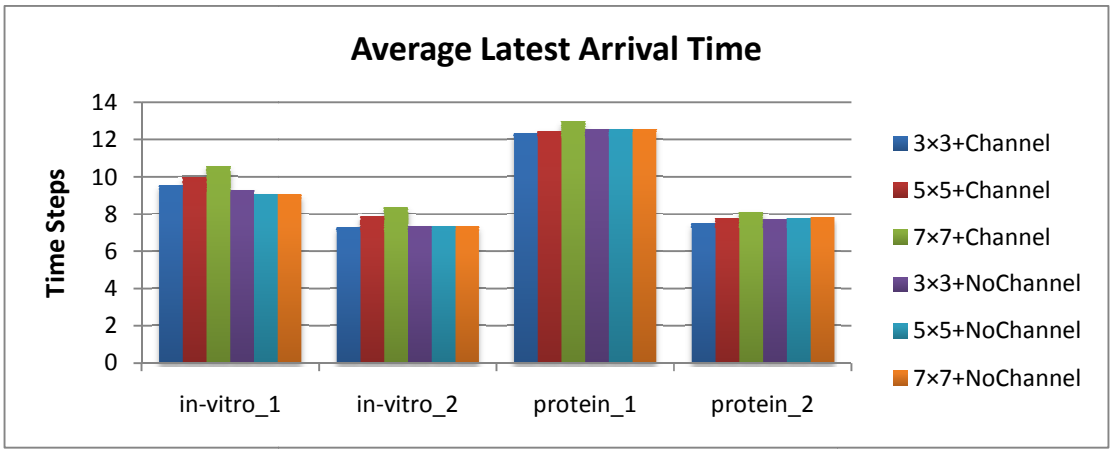


Figure 6.12 Straightness-prone, Diagonal movement, Used cells preference (MEDA).





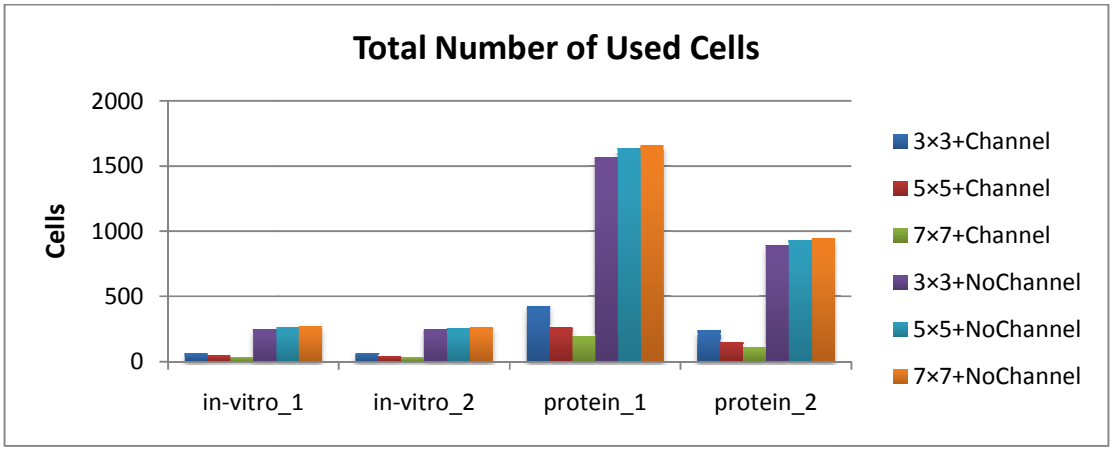
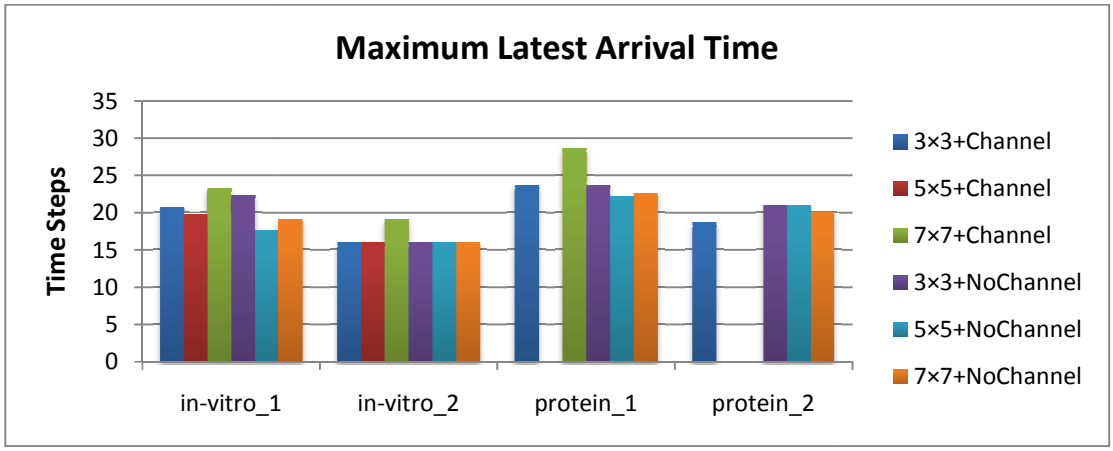
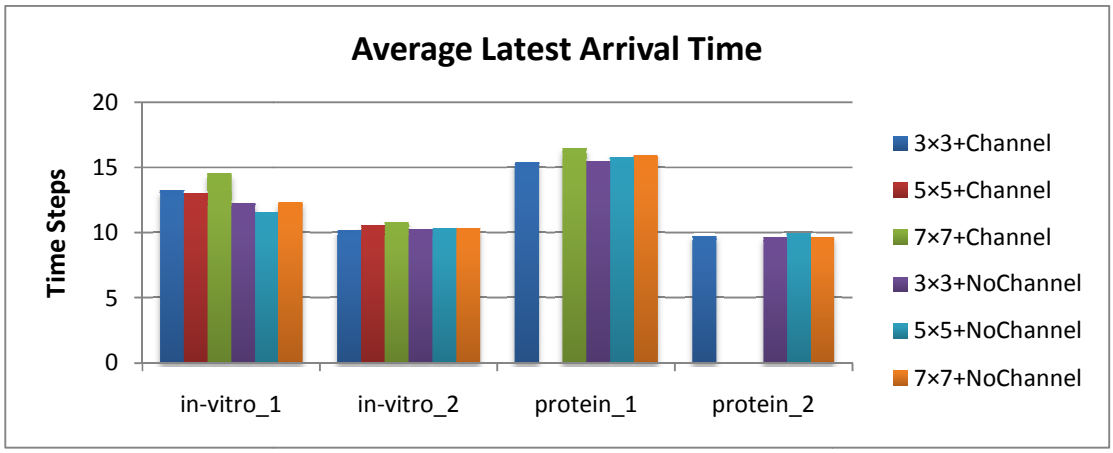


Figure 6.13 No Straightness-prone, Diagonal movement, Used cells preference (MEDA).





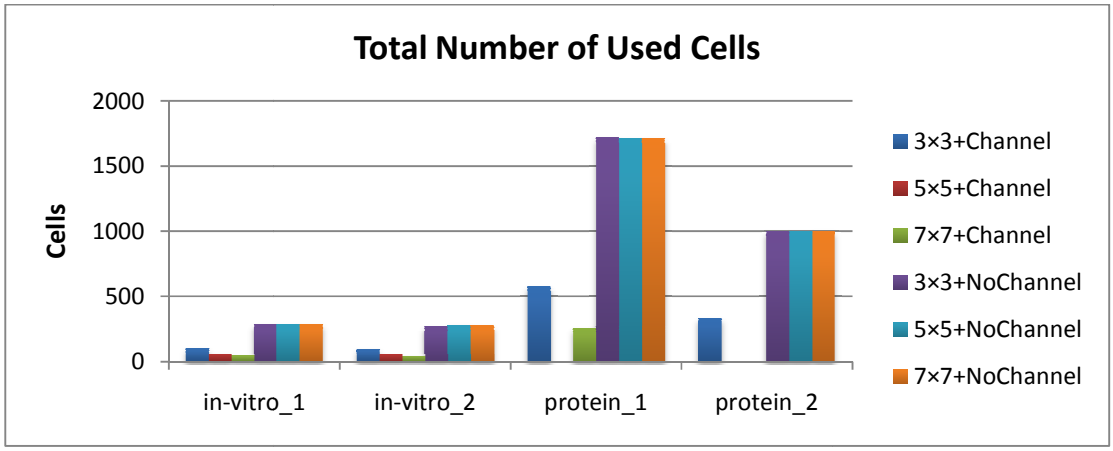
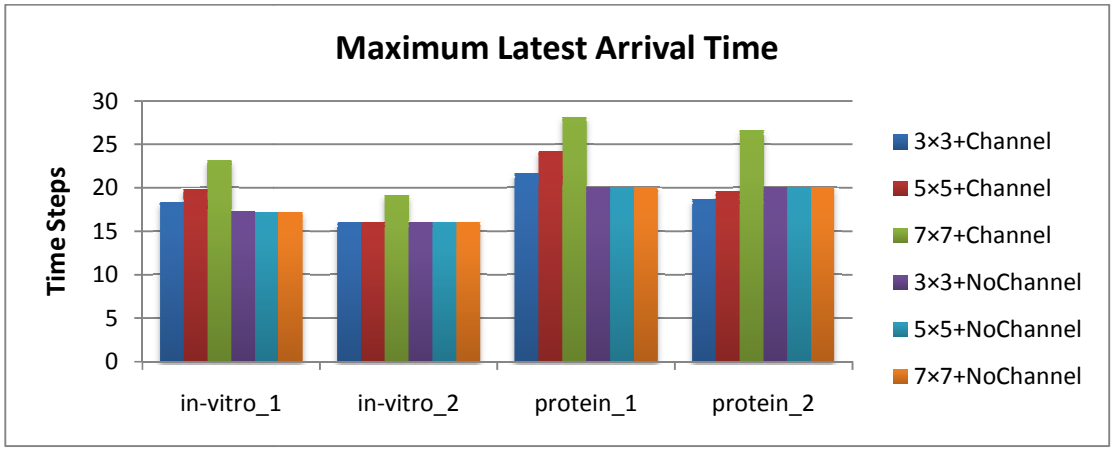
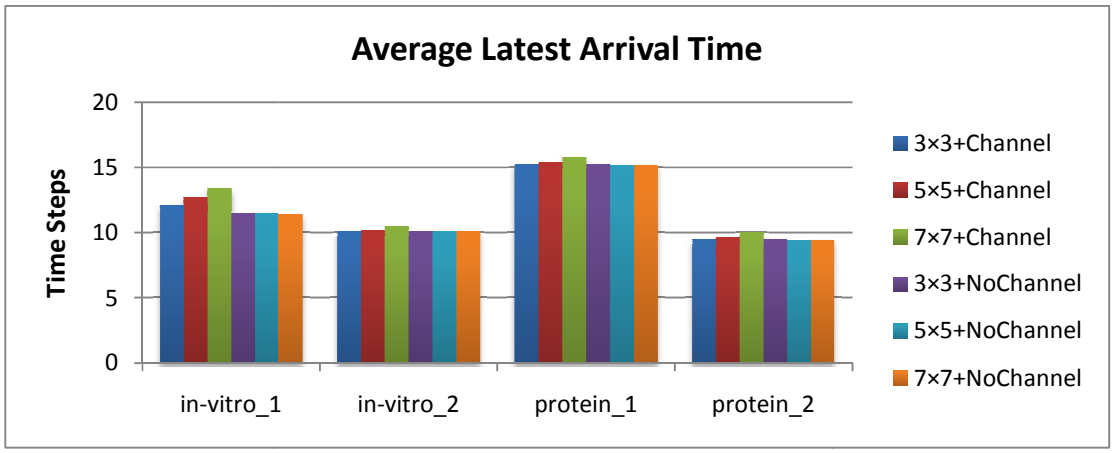


Figure 6.14 Straightness-Prone, No Diagonal movement No Used cells preference (MEDA).





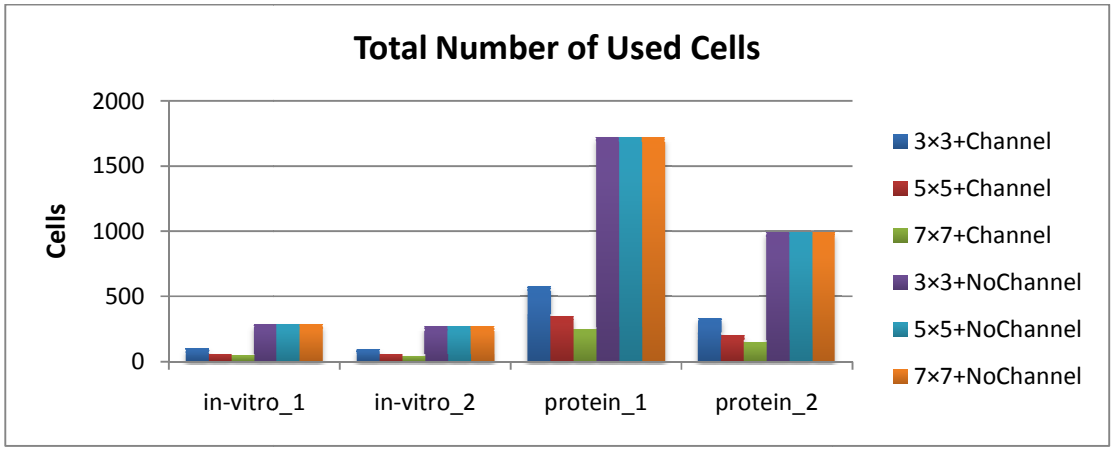


Figure 6.15 No Straightness-prone, No Diagonal movement, No Used cells preference (MEDA).

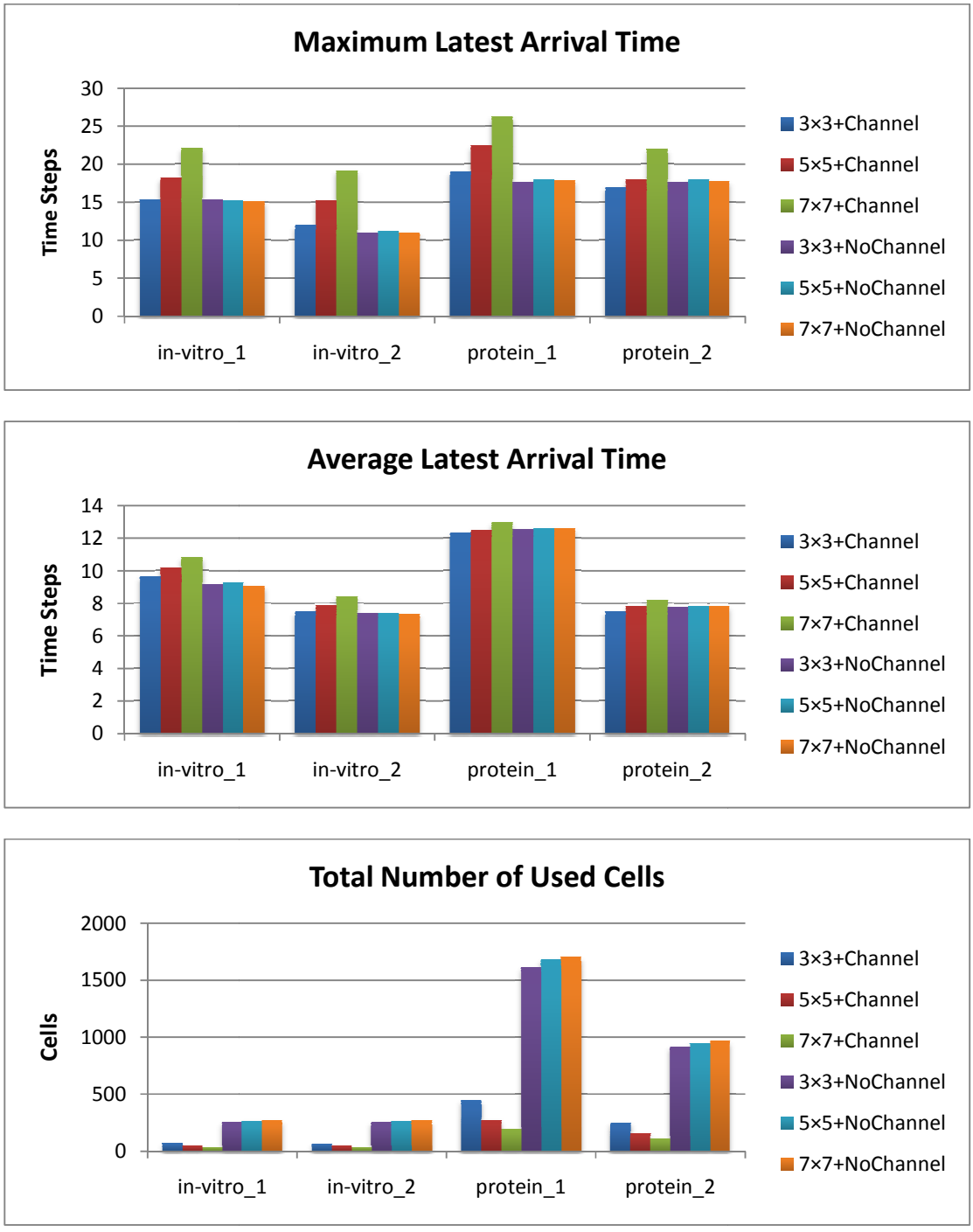
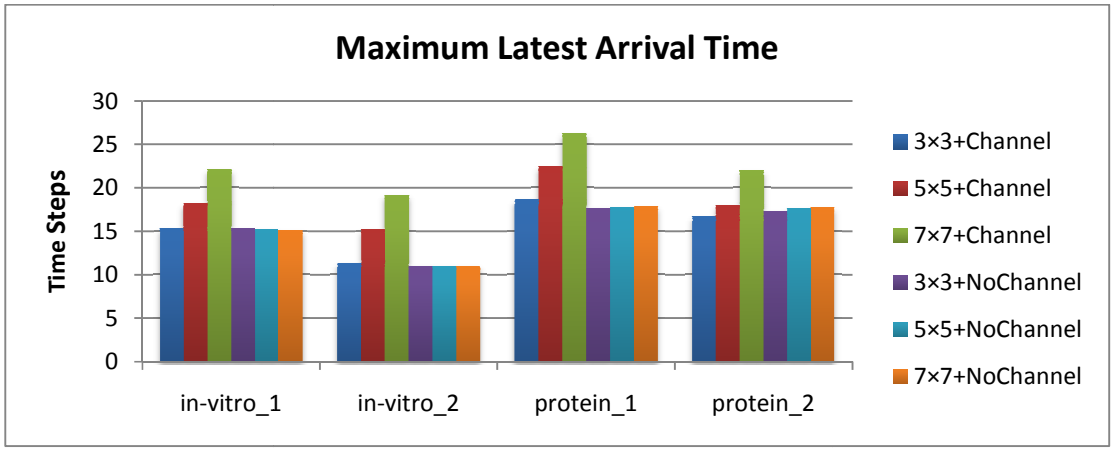
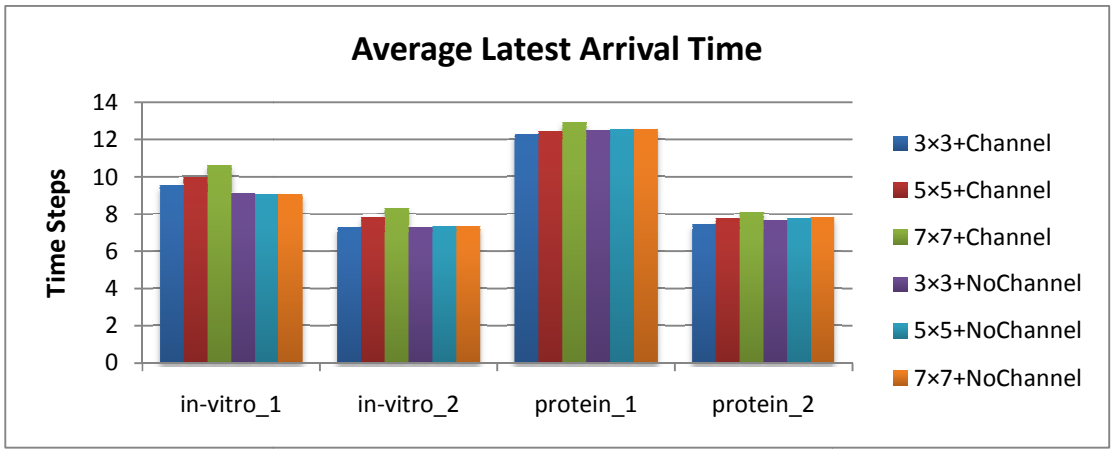


Figure 6.16 Straightness-prone, Diagonal movement, No Used cells preference (MEDA).





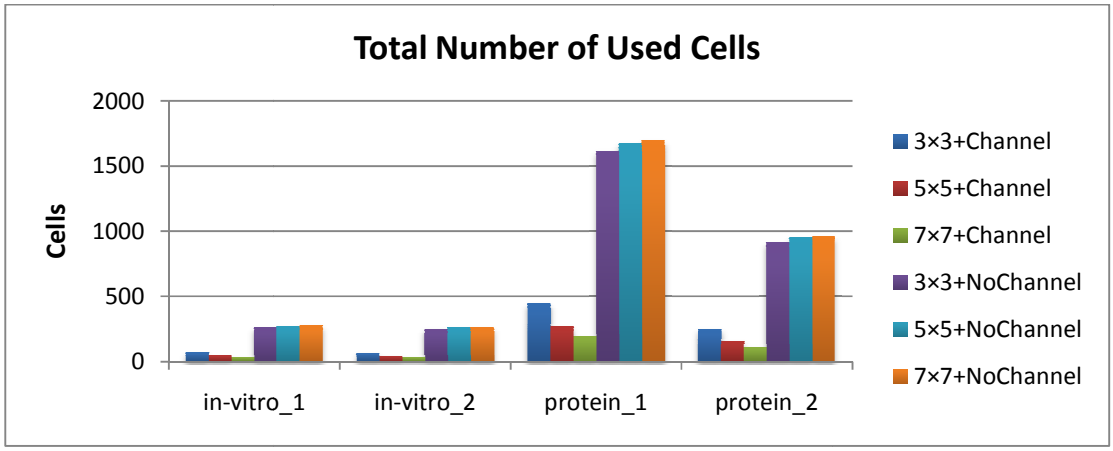


Figure 6.17 No Straightness-prone, Diagonal movement, No Used cells preference (MEDA).

6.7 Comparision with Previous Published Results

Table 6.7 and Table 6.8 show the comparison between the proposed algorithm and others in traditional DMFB architecture. The last row compares the proposed algorithm to so far the best algorithm[23]. It shows that the proposed algorithm has the highest parallelism. In terms of total number of used cells, the optimized results of the proposed algorithm are better than Cho's algorithm [21] and slightly higher than Huang's algorithm [23]. If the diagonal movement is enabled, t_{org}^{l} and t_{org}^{acc} can be dramatically improved while Ω_{org} can be greatly reduced.

In MEDA, channel-based routing and non-channel-based routing results are compared to the results in traditional DMFB architecture, respectively. Although Ω_{eqv} can be greatly reduced by increasing droplet size in channel-based routing, t_{eqv}^{l} and t_{eqv}^{acc} are increased as well. Furthermore, non-channel-based routing cannot benefit from the increase of droplet size, either. Thus the results of 3×3 (Assumed to be the smallest size of droplet in MEDA in this thesis) droplet are chosen to make comparison to other's results.

For channel-based routing, by the comparison between Table 6.9 and Table 6.8, it can be seen that t_{eqv}^{l} , t_{eqv}^{acc} and Ω_{eqv} are reduced by enabling channel-based parameter. Especially for Ω_{eqv} , it can be reduced by more than 50%. If diagonal movement is enabled, then t_{eqv}^{l} is decreased, t_{eqv}^{acc} is increased for a little bit and Ω_{eqv} still get greatly decreased comparing to traditional DMFBs.

For non-channel-based routing, by the comparison between Table 6.9 and Table 6.10, it concludes that t_{eqv}^{l}, t_{eqv}^{acc} and Ω_{eqv} get reduced for a little bit. If diagonal movement is enabled, then t_{eqv}^{l} is decreased, t_{eqv}^{acc} is increased for a little bit and Ω_{eqv} get increased as well.

Table 6.7 Results of Path-based + N.S. and Path-based + D., N.S. on traditional DMFBs.

BenchMark Suite	**Prioritized A* [17]**			**Two-Stage [11]**			**Ours (Path-Based + N.S.)**			**Ours (Path-Based + D., N.S.)**		
Name	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell
in-vitro_1	N/A	N/A	269	N/A	N/A	263	20.00	12.27	292	17.00	9.64	208
in-vitro_2	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	20.00	10.73	274	15.00	7.80	200
protein_1	FAIL	FAIL	FAIL	N/A	N/A	1735	20.00	15.44	1734	20.00	12.48	1322
protein_2	FAIL	FAIL	FAIL	FAIL	FAIL	FAIL	20.00	9.83	1001	16.00	7.44	718
Compared to Huang	N/A	N/A	N/A	N/A	N/A	N/A	**1.07**	**1.00**	**1.11**	**0.91**	**0.77**	**0.82**

Table 6.8 Optimized results of Path-based + N.S. and Path-based + D., N.S. on traditional DMFBs.

BenchMark Suite	**Cho**			**Huang**			**Ours (Optimized + N.S.)**			**Ours (Optimized + D., N.S.)**		
Name	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell
in-vitro_1	19	14.30	258	18	12.47	231	19	12.18	251	16	9.36	182
in-vitro_2	20	12.00	246	17	10.43	229	16	10.27	260	14	7.53	182
protein_1	20	16.55	1699	20	15.51	1588	20	15.34	1636	20	12.39	1205
protein_2	20	12.19	963	20	10.04	923	20	9.67	948	16	7.26	686
Compared to Huang	1.05	1.14	1.07	1.00	1.00	1.00	**1.00**	**0.98**	**1.04**	**0.88**	**0.75**	**0.76**

Table 6.9 Results of Path-based + N.S. and Path-based + D., N.S. in 3×3 channel-based MEDA.

BenchMark Suite	**Ours (MEDA + Path-Based + N.S.), Channel, 3×3**			**Ours (MEDA + Path-Based + D., N.S.), Channel, 3×3**			**Ours (MEDA + Optimized + N.S.), Channel, 3×3**			**Ours (MEDA + Optimized + D., N.S.), Channel, 3×3**		
Name	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell
in-vitro_1	27.33	13.73	90.00	18.67	9.97	65.33	18.33	12.24	87.00	15.33	9.55	64.44
in-vitro_2	23.00	11.00	91.22	17.00	7.96	66.22	16.00	10.11	85.67	11.33	7.29	64.00
protein_1	26.00	15.81	558.78	23.33	12.80	443.11	21.00	15.24	545.89	18.67	12.30	427.56
protein_2	27.00	10.01	315.56	21.67	7.79	240.78	18.67	9.49	309.44	16.67	7.48	239.11
Compared to Huang	**1.38**	**1.04**	**0.36**	**1.08**	**0.80**	**0.27**	**0.99**	**0.97**	**0.35**	**0.83**	**0.76**	**0.27**

Table 6.10 Results of Path-based + N.S. and Path-based + D., N.S. in non-channel-based 3×3 MEDA.

BenchMark Suite	**Ours (MEDA + Path-Based + N.S.), No Channel, 3×3**			**Ours (MEDA + Path-Based + D., N.S.), No Channel, 3×3**			**Ours (MEDA + Optimized + N.S.), No Channel, 3×3**			**Ours (MEDA + Optimized + D., N.S.), No Channel, 3×3**		
Name	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell	Max. A.	Avg. A.	#Cell
in-vitro_1	23.33	12.73	265.44	17.00	9.67	254.78	17.33	11.79	249.33	15.33	9.30	245.56
in-vitro_2	23.00	11.11	264.89	19.33	8.00	240.44	16.00	10.13	258.44	11.33	7.36	244.44
protein_1	21.67	15.48	1666.56	21.00	12.80	1600.89	21.00	15.27	1614.78	18.67	12.54	1563.00
protein_2	20.00	9.82	967.22	17.33	7.92	909.78	20.00	9.50	940.67	17.33	7.74	893.56
Compared to Huang	**1.17**	**1.01**	**1.06**	**1.00**	**0.79**	**1.01**	**0.99**	**0.96**	**1.03**	**0.84**	**0.76**	**0.99**

6.8 Summary

This chapter presents the effects of adjusting different parameters. First of all, by investigating different priority solvers and their combinations in traditional DMFB architecture, path-based priority solver combined with dynamic routing shows the best

results. Therefore, this combination of priority solvers is also applied to the droplet routing algorithm in MEDA. The effects of different parameters are summarized as follows:

- The straightness-prone parameter can reduce t^{l}_{org} and t^{acc}_{org} for a little bit in traditional DMFB architecture, but Ω_{org} is increased. For MEDA, straightness-prone parameter does not have positive effects. Thus enabling this parameter is not recommended except the paths of droplets need to be elegant.
- By enabling diagonal movement, $t^{l}_{org/eqv}$, $t^{acc}_{org/eqv}$ and $\Omega_{org/eqv}$ can be reduced for both traditional DMFB and MEDA architectures. It is a very effective approach to improve the simulation results if diagonal movement is permitted on DMFB.
- For both traditional DMFB and MEDA architectures, by enabling used cells preference parameter, the $\Omega_{org/eqv}$ can be apparently reduced. However $t^{l}_{org/eqv}$ and $t^{acc}_{org/eqv}$ get increased for a little bit. Since the side effect is not too much, enabling this parameter is recommended.
- As an exclusive approach for MEDA, channel-based routing can greatly improve routing results; especially can reduce Ω_{eqv} by more than 50%. And the larger droplets size is, the more cells can be reduced. The results of non-channel-based routing is similar to traditional DMFBs, therefore if droplet sizes are assumed to be same all the time, there is no advantage to use MEDA.

Chapter 7 Conclusions and Future Work

7.1 Conclusions

DMFB has been introduced in recent years and begins to replace many of applications of traditional continuous flow biochips. Nowadays, a new DMFB architecture called MEDA allows precise control and droplets with different sizes. For DMFB, droplet routing algorithm determines the effect usage of the system. This work presents a new droplet routing algorithms for both traditional and MEDA DMFB systems based on 3D-A* search algorithm with block setting, dynamic routing and diagonal moving features. An innovative approach to 3-pin net routing is proposed to avoid the unexpected merging during routing. For the MEDA, an exclusive channel-based routing approach is investigated.

As the proposed routing algorithm is categorized to sequential approach, the priority of nets needs to be decided prior to droplet routing in order to reduce interferences between nets. Cho's algorithm and Huang's algorithm showed high routability and effective cell usage, thus priority solvers of these two algorithms are implemented. A

path-based priority solver and a dynamic routing approach which is able to change priorities while routing, are also introduced. These priority solvers are combined with each other to complement 3D-A* search algorithm and the effects are investigated.

Parameters of the routing algorithm are adjustable; the straightness-prone parameter encourages the droplet to move towards the same direction as the previous time step; the used cells preference parameter encourages the droplet to move to the used cells; also the diagonal movement parameter can be enabled. For MEDA, the channel-based routing parameter is able to be enabled. Effects of combination of these parameters are investigated.

The simulation results show that path-based priority solver combined with dynamic routing result in the least maximum latest arrival time, average latest arrival time and total number of used cells in traditional DMFB architecture. Therefore, this combination of priority solvers is also applied to the routing algorithm for MEDA

Furthermore, the simulation results conclude that the straightness-prone parameter does not have many positive effects on both traditional DMFB and MEDA architectures; diagonal movement is a very effective approach to improve the results.

For MEDA, the proposed routing algorithm can not only handle this architecture directly, but also allow an innovative channel-based routing approach. The simulation results show that channel-based routing can achieve lower latest arrival and average latest arrival time and dramatically reduced total number of used cells. The larger droplet size is, the more cells can be reduced. The results of non-channel-based routing is similar to traditional DMFBs, therefore if droplet sizes are assumed to be same all the time, there is no advantage to use MEDA from routing point of view.

7.2 Future Work

If droplets are not properly placed in the synthesis stage, some cases may not be solved by the proposed algorithm. As shown in Figure 7.1, two droplets need swap position. However, this case can be solved by moving one droplet to a concession zone temporarily; waiting for another droplet reaches its sink, and moving the droplet out of concession zone to its own sink. This problem is very similar to the dead lock problem in SoKoBan puzzle (See Figure 7.2). The goal of the game is to simply move all the boxes to the designated positions. A robot which can quickly and automatically solve puzzles is demanded. Basically the robot detects dead lock first, and applies an approach called reverse searching to get rid of dead lock. This approach may also be able to solve congestion problem in droplet routing.

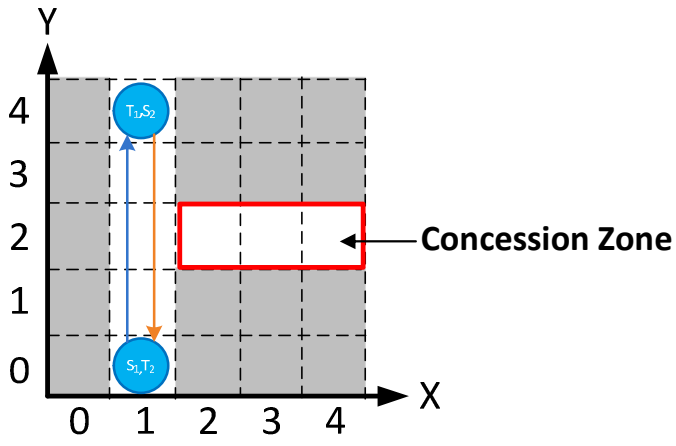


Figure 7.1 A case which cannot be solved by the proposed algorithm in traditional DMFBs.

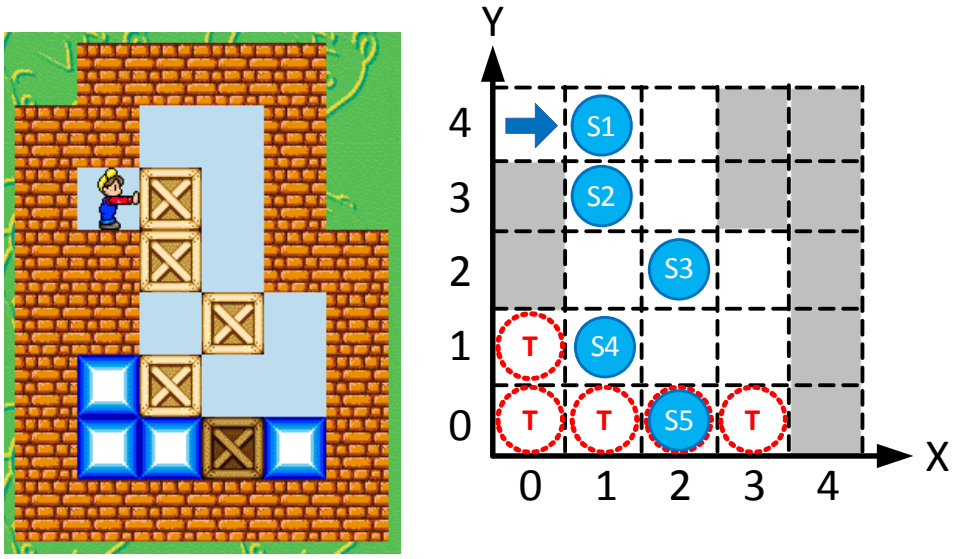


Figure 7.2 SoKoBan puzzle [38] and its representation.

The diagonal movement has not been implemented in others work. Therefore the comparison cannot be made between these algorithms and the proposed algorithm respect to the effect of diagonal movement. We open the door to the authors of existing algorithms to implement the diagonal movement feature and make comparison with our work.

References

[1] G. Whitesides, "The origins and the future of microfluidics," *Nature,* vol. 442, No. 7101, pp. 368-373, 2006.

[2] G. Wang, D. Teng, and S. Fan, "Digital microfluidic operations on micro-electrode array architecture.," *the 6th IEEE International Conference on Nano/Micro Engineered and Molecular Systems (NEMS)*, 2011.

[3] K. Chakrabarty, "Digital Microfluidic Biochips: A Vision for Functional Diversity and More than Moore," *2010 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2010.

[4] A. R. Boer, B. Bruyneel, J. G. Krabbe, H. Lingeman, W. M. A. Niessen, and H. Irth, "A microfluidic-based enzymatic assay for bioactivity screening combined with capillary liquid chromatography and mass spectrometry," *Lab on a Chip*, vol. 5, No. 11, pp. 1286-1292, 2005.

[5] J. Song, R. Evans, Y. Y. Lin, B. N. Hsu, and R. Fair, "A scaling model for electrowetting-on-dielectric microfluidic actuators," *Microfluidics and Nanofluidics*, vol. 7, No. 1, pp. 75-89, 2009.

[6] A. Guiseppi-Elie, S. Brahim, G. Slaughter, and K. Ward, "Design of a subcutaneous implantable biochip for monitoring of glucose and lactate," *Sensors Journal, IEEE,* vol. 5, No. 3, pp. 345-355, 2005.

[7] R. Fair, A. Khlystov, T. Tailor, V. Ivanov, R. Evans, P. Griffin, V. Srinivasan, V. Pamula, M. Pollack, and J. Zhou, "Chemical and biological applications of digital-microfluidic devices," *Design & Test of Computers, IEEE,* vol. 24, No. 1, pp. 10-24, 2007.

[8] E. Ottesen, J. Hong, S. Quake, and J. Leadbetter, "Microfluidic digital PCR enables multigene analysis of individual environmental bacteria," *Science,* vol. 314, No. 5804, p. 1464, 2006.

[9] V. Srinivasan, V. Pamula, and R. Fair, "An integrated digital microfluidic lab-on-a-chip for clinical diagnostics on human physiological fluids," *Lab on a Chip,* vol. 4, No. 4, pp. 310-315, 2004.

[10] T. Xu and K. Chakrabarty, "Droplet-trace-based array partitioning and a pin assignment algorithm for the automated design of digital microfluidic biochips," *the 4th International Conference on Hardware/Software Codesign and System Synthesis*, 2007.

[11] F. Su, W. Hwang, and K. Chakrabarty, "Droplet routing in the synthesis of digital microfluidic biochips," *2006 Design, Automation and Test in Europe*, 2006.

[12] T. Huang and T. Ho, "A two-stage ILP-based droplet routing algorithm for pin-constrained digital microfluidic biochips," *the 19th international symposium on physical design*, 2010.

[13] Z. Xiao and E. Young, "Crossrouter: A droplet router for cross-referencing digital microfluidic biochips," *2010 15th Asia and South Pacific Design Automation Conference*, 2010.

[14] T. Xu and K. Chakrabarty, "Broadcast electrode-addressing for pin-constrained multi-functional digital microfluidic biochips," *the 45th ACM/IEEE Design Automation Conference*, 2008.

[15] T. Xu and K. Chakrabarty, "A cross-referencing-based droplet manipulation method for high-throughput and pin-constrained digital microfluidic arrays," *2007 Design, Automation & Test in Europe Conference & Exhibition* 2007.

[16] S. Fan, C. Hashi, and C. Kim, "Manipulation of multiple droplets on N× M grid by cross-reference EWOD driving scheme and pressure-contact packaging," *The 16th IEEE Annual International Conference on Micro Electro Mechanical Systems*, 2003.

[17] K. Bohringer, "Towards optimal strategies for moving droplets in digital microfluidic systems," *2004 IEEE International Conference on Robotics and Automation*, 2004.

[18] Y. Zhao and K. Chakrabarty, "Cross-contamination avoidance for droplet routing in digital microfluidic biochips," *2009 Design, Automation & Test in Europe Conference & Exhibition*, 2009.

[19] T. Huang, C. Lin, and T. Ho, "A contamination aware droplet routing algorithm for digital microfluidic biochips," *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, 2009.

[20] X. Zhang, F. van Proosdij, and H. Kerkhoff, "A droplet routing technique for fault-tolerant digital microfluidic devices," *the 14th IEEE International Mixed-Signals, Sensors, and Systems Test Workshop*, 2008.

[21] M. Cho and D. Pan, "A high-performance droplet routing algorithm for digital microfluidic biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, No. 10, pp. 1714-1724, 2008.

[22] P. Yuh, C. Yang, and Y. Chang, "BioRoute: A network-flow-based routing algorithm for the synthesis of digital microfluidic biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, No. 11, pp. 1928-1941, 2008.

[23] T. Huang and T. Ho, "A fast routability-and performance-driven droplet routing algorithm for digital microfluidic biochips," *2009 IEEE International Conference on Computer Design*, 2010.

[24] C. Lin and Y. Chang, "ILP-based pin-count aware design methodology for microfluidic biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, No. 9, pp. 1315-1327, 2010.

[25] P. Yuh, S. Sapatnekar, C. Yang, and Y. Chang, "A progressive-ILP-based routing algorithm for the synthesis of cross-referencing biochips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, No. 9, pp. 1295-1306, 2009.

[26] T. Lengauer, *Combinatorial algorithms for integrated circuit layout*: John Wiley & Sons, Inc. New York, NY, USA, 1990.

[27] Antoine Deza, Chris Dickson, Tamas Terlaky, Anthony Vannelli, and H. Zhang. (2010). *Global Routing in VLSI Design: Algorithms, Theory, and Computational Practice*. Available: http://www.optimization-online.org/DB_FILE/2010/12/2852.pdf

[28] M. Sarrafzadeh and C. K. Wong, *An introduction to VLSI physical design*: McGraw-Hill, 1996.

[29] C. Lee, "An algorithm for path connections and its applications," *IRE Transactions on Electronic Computers,* No. 3, pp. 346-365, 1961.

[30] H. Zhou. *Introduction to VLSI CAD*. Available: http://users.eecs.northwestern.edu/~haizhou/357/lec5.pdf

[31] E. Shragowitz and S. Keel, "A global router on a multi-commodity flow model," *Interaction,* vol. 5, No. pp. 3-16, 1987.

[32] P. Raghavan, "Probabilistic construction of deterministic algorithms: approximating packing integer programs," *Journal of Computer and System Sciences,* vol. 37, No. 2, pp. 130-143, 1988.

[33] P. Raghavan and C. Tompson, "Randomized rounding: a technique for provably good algorithms and algorithmic proofs," *Combinatorica,* vol. 7, No. 4, pp. 365-374, 1987.

[34] D. Silver. (2005). *Cooperative pathfinding*. Available: http://www.cs.ucl.ac.uk/staff/D.Silver/web/Applications_files/coop-path-AIWisdom.pdf

[35] F. Su and K. Chakrabarty, "Architectural-level synthesis of digital microfluidics-based biochips," *2004 IEEE/ACM International Conference on Computer Aided Design*, 2005.

[36] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, No. 2, pp. 100-107, 1968.

[37] P. Lester. (2005). *A* Pathfinding for Beginners*. Available: http://www.policyalmanac.org/games/aStarTutorial.htm

[38] NBALANCE. (2000). *Sokoban Puzzle Guide of Windows 95*. Available: http://en.wikipedia.org/wiki/Sokoban

Appendix A

Routing results in traditional DMFB architecture.

Test Case Name	Dyn.	Huang+Dyn.	Cho+Dyn.	Our+Dyn.	Huang+Our+Dyn.	Cho+Our+Dyn.	Optimized
5D, Straight							
Maximum latest arrival time							
in-vitro_1	20.00	20.00	19.00	20.00	20.00	20.00	19.00
in-vitro_2	20.00	19.00	19.00	20.00	20.00	20.00	16.00
protein_1	20.00	20.00	20.00	20.00	20.00	20.00	20.00
protein_2	20.00	20.00	20.00	20.00	20.00	20.00	20.00

Average latest arrival time							
in-vitro_1	12.27	13.00	12.55	12.27	12.55	12.27	12.18
in-vitro_2	11.00	10.93	10.53	10.73	10.53	10.73	10.27
protein_1	15.45	15.41	15.56	15.45	15.41	15.56	15.31
protein_2	9.82	9.86	9.85	9.82	9.86	9.85	9.63
Total number of used cells							
in-vitro_1	291.00	283.00	284.00	291.00	292.00	288.00	251.00
in-vitro_2	277.00	280.00	277.00	278.00	270.00	271.00	265.00
protein_1	1731.00	1739.00	1770.00	1732.00	1738.00	1758.00	1651.00
protein_2	1023.00	1018.00	1025.00	1023.00	1018.00	1025.00	961.00
5D, No Straight							
Maximum latest arrival time							
in-vitro_1	20.00	20.00	19.00	20.00	20.00	20.00	19.00
in-vitro_2	20.00	19.00	19.00	20.00	20.00	19.00	16.00
protein_1	20.00	20.00	20.00	20.00	20.00	20.00	20.00
protein_2	20.00	20.00	20.00	20.00	20.00	20.00	20.00
Average latest arrival time							

in-vitro_1	12.27	12.73	12.27	12.27	12.64	12.27	12.18
in-vitro_2	10.87	10.67	10.67	10.73	10.53	10.67	10.27
protein_1	15.44	15.41	15.55	15.44	15.41	15.55	15.34
protein_2	9.83	9.86	9.82	9.83	9.86	9.82	9.67
Total number of used cells							
in-vitro_1	291.00	297.00	294.00	292.00	294.00	289.00	251.00
in-vitro_2	278.00	280.00	279.00	274.00	282.00	280.00	260.00
protein_1	1731.00	1739.00	1776.00	1734.00	1741.00	1767.00	1636.00
protein_2	1001.00	1006.00	1002.00	1001.00	1015.00	1002.00	948.00
9D, Straight							
Maximum latest arrival time							
in-vitro_1	20.00	19.00	19.00	17.00	16.00	17.00	16.00
in-vitro_2	17.00	16.00	14.00	15.00	14.00	14.00	12.00
protein_1	20.00	20.00	19.00	20.00	20.00	19.00	17.00
protein_2	16.00	18.00	17.00	16.00	16.00	17.00	16.00
Average latest arrival time							
in-vitro_1	9.91	9.73	9.82	9.64	9.45	9.73	9.45

in-vitro_2	8.13	8.20	7.73	7.80	7.60	7.73	7.40
protein_1	12.64	12.67	12.61	12.58	12.61	12.62	12.28
protein_2	7.49	7.47	7.42	7.44	7.40	7.37	7.26
Total number of used cells							
in-vitro_1	203.00	201.00	216.00	208.00	205.00	207.00	187.00
in-vitro_2	205.00	206.00	194.00	195.00	196.00	194.00	183.00
protein_1	1315.00	1327.00	1329.00	1311.00	1323.00	1329.00	1246.00
protein_2	724.00	730.00	726.00	718.00	721.00	721.00	687.00
9D, No Straight							
Maximum latest arrival time							
in-vitro_1	19.00	18.00	18.00	17.00	16.00	17.00	16.00
in-vitro_2	17.00	16.00	14.00	15.00	14.00	14.00	14.00
protein_1	20.00	20.00	20.00	20.00	20.00	20.00	20.00
protein_2	16.00	18.00	16.00	16.00	16.00	16.00	16.00
Average latest arrival time							
in-vitro_1	9.82	9.64	9.73	9.64	9.45	9.73	9.36
in-vitro_2	8.07	8.07	7.73	7.80	7.60	7.73	7.53

protein_1	12.55	12.58	12.59	12.48	12.52	12.59	12.39
protein_2	7.44	7.47	7.31	7.44	7.40	7.31	7.26
Total number of used cells							
in-vitro_1	206.00	205.00	213.00	208.00	207.00	208.00	182.00
in-vitro_2	203.00	201.00	197.00	200.00	202.00	197.00	182.00
protein_1	1327.00	1339.00	1331.00	1322.00	1326.00	1329.00	1205.00
protein_2	719.00	733.00	720.00	718.00	725.00	720.00	686.00

Appendix B

Optimized Routing results in MEDA with used cells preference. F means the corresponding test case is not routable.

Test Case Name	3×3+Channel	5×5+Channel	7×7+Channel	3×3+NoChannel	5×5+NoChannel	7×7+NoChannel
5D, Straight						
Maximum latest arrival time						
in-vitro_1	18.33	20.00	24.14	17.33	17.20	17.14
in-vitro_2	16.00	16.00	19.14	16.00	16.00	16.00
protein_1	24.00	25.00	29.00	20.00	22.40	20.00

protein_2	18.67	F	F	20.00	20.00	20.00
Average latest arrival time						
in-vitro_1	12.36	12.80	13.49	11.79	11.53	11.42
in-vitro_2	10.07	10.19	10.50	10.13	10.11	10.10
protein_1	15.35	15.49	15.85	15.26	15.29	15.23
protein_2	9.49	F	F	9.47	9.45	9.44
Total number of used cells						
in-vitro_1	85.33	54.96	40.12	243.33	254.80	254.43
in-vitro_2	88.11	54.64	39.43	257.22	262.84	261.00
protein_1	547.56	335.84	246.61	1621.33	1623.16	1643.16
protein_2	310.89	F	F	943.00	946.04	950.47
5D, No Straight						
Maximum latest arrival time						
in-vitro_1	18.33	20.60	24.43	17.33	17.20	17.14
in-vitro_2	16.00	16.00	20.00	16.00	16.00	16.00
protein_1	21.00	24.60	28.43	21.00	21.00	20.00
protein_2	18.67	20.20	23.43	20.00	20.00	20.00

Average latest arrival time						
in-vitro_1	12.24	12.85	13.52	11.79	11.53	11.42
in-vitro_2	10.11	10.20	10.56	10.13	10.11	10.10
protein_1	15.24	15.48	15.83	15.27	15.26	15.22
protein_2	9.49	9.67	9.99	9.50	9.45	9.44
Total number of used cells						
in-vitro_1	87.00	55.04	41.18	249.33	255.76	259.82
in-vitro_2	85.67	53.16	38.92	258.44	259.64	259.86
protein_1	545.89	333.92	245.18	1614.78	1625.32	1643.14
protein_2	309.44	192.32	140.73	940.67	940.48	945.02
9D, Straight						
Maximum latest arrival time						
in-vitro_1	15.33	18.40	22.29	15.33	15.20	15.14
in-vitro_2	12.00	16.00	20.00	11.00	11.00	11.00
protein_1	18.67	22.40	26.29	17.67	17.80	17.86
protein_2	16.67	17.20	21.14	17.33	17.60	17.71
Average latest arrival time						

in-vitro_1	9.58	10.02	10.57	9.30	9.29	9.05
in-vitro_2	7.33	7.89	8.34	7.31	7.32	7.32
protein_1	12.28s	12.50	12.94	12.49	12.54	12.56
protein_2	7.48	7.77	8.12	7.70	7.77	7.81
Total number of used cells						
in-vitro_1	64.33	40.76	31.12	244.89	257.56	268.94
in-vitro_2	62.78	39.00	29.88	243.67	256.80	263.35
protein_1	431.11	266.48	191.47	1565.67	1650.48	1671.94
protein_2	235.67	148.96	110.24	888.11	923.64	938.69
9D, No Straight						
Maximum latest arrival time						
in-vitro_1	15.33	18.40	22.29	15.33	15.20	15.14
in-vitro_2	11.33	16.00	20.00	11.33	11.00	11.00
protein_1	18.67	22.40	26.29	18.67	17.80	17.86
protein_2	16.67	17.20	21.14	17.33	17.60	17.71
Average latest arrival time						
in-vitro_1	9.55	10.00	10.57	9.30	9.07	9.05

in-vitro_2	7.29	7.89	8.34	7.36	7.32	7.32
protein_1	12.30	12.47	12.95	12.54	12.54	12.56
protein_2	7.48	7.77	8.12	7.74	7.77	7.81
Total number of used cells						
in-vitro_1	64.44	41.24	30.84	245.56	264.00	269.96
in-vitro_2	64.00	39.24	30.47	244.44	254.80	259.12
protein_1	427.56	260.84	189.73	1563.00	1630.96	1661.71
protein_2	239.11	150.28	110.76	893.56	931.00	945.86

Appendix C

Optimized Routing results in MEDA without used cells preference. F means the corresponding test case is not routable.

Test Case Name	3×3+Channel	5×5+Channel	7×7+Channel	3×3+NoChannel	5×5+NoChannel	7×7+NoChannel
5D, Straight						
Maximum latest arrival time						
in-vitro_1	20.67	19.80	23.29	22.33	17.60	19.14
in-vitro_2	16.00	16.00	19.14	16.00	16.00	16.00
protein_1	23.67	F	28.71	23.67	22.20	22.57
protein_2	18.67	F	F	21.00	21.00	20.00
Average latest arrival time						

in-vitro_1	13.21	12.98	14.51	12.27	11.58	12.38
in-vitro_2	10.20	10.55	10.78	10.24	10.37	10.35
protein_1	15.41	F	16.49	15.48	15.79	15.98
protein_2	9.74	F	F	9.62	9.98	9.66
Total number of used cells						
in-vitro_1	97.56	58.60	43.00	283.33	281.08	286.53
in-vitro_2	92.33	54.88	40.18	268.44	278.16	274.18
protein_1	576.44	F	252.45	1715.33	1707.56	1712.10
protein_2	332.00	F	F	999.33	999.00	997.73
5D, No Straight						
Maximum latest arrival time						
in-vitro_1	18.33	19.80	23.14	17.33	17.20	17.14
in-vitro_2	16.00	16.00	19.14	16.00	16.00	16.00
protein_1	21.67	24.20	28.14	20.00	20.00	20.00
protein_2	18.67	19.60	26.57	20.00	20.00	20.00
Average latest arrival time						
in-vitro_1	12.12	12.73	13.40	11.48	11.44	11.42

in-vitro_2	10.07	10.20	10.50	10.13	10.11	10.10
protein_1	15.26	15.43	15.81	15.23	15.22	15.21
protein_2	9.48	9.65	10.06	9.45	9.44	9.43
Total number of used cells						
in-vitro_1	99.00	59.84	42.88	284.89	284.72	284.65
in-vitro_2	90.00	54.96	39.10	267.56	266.16	265.92
protein_1	574.11	346.16	249.02	1719.56	1717.68	1716.76
protein_2	328.67	199.72	145.94	988.22	987.96	987.84
9D, Straight						
Maximum latest arrival time						
in-vitro_1	15.33	18.20	22.14	15.33	15.20	15.14
in-vitro_2	12.00	15.20	19.14	11.00	11.20	11.00
protein_1	19.00	22.40	26.29	17.67	18.00	17.86
protein_2	17.00	18.00	22.00	17.67	18.00	17.71
Average latest arrival time						
in-vitro_1	9.64	10.18	10.84	9.15	9.29	9.08
in-vitro_2	7.51	7.88	8.41	7.40	7.40	7.32

protein_1	12.32	12.52	12.96	12.58	12.61	12.60
protein_2	7.50	7.84	8.20	7.78	7.84	7.84
Total number of used cells						
in-vitro_1	69.44	41.88	31.00	252.67	264.88	267.92
in-vitro_2	65.56	41.28	30.47	254.78	262.88	267.98
protein_1	443.33	271.84	194.22	1614.22	1679.96	1705.00
protein_2	250.56	155.60	112.55	917.56	949.04	963.96
9D, No Straight						
Maximum latest arrival time						
in-vitro_1	15.33	18.20	22.14	15.33	15.20	15.14
in-vitro_2	11.33	15.20	19.14	11.00	11.00	11.00
protein_1	18.67	22.40	26.29	17.67	17.80	17.86
protein_2	16.67	18.00	22.00	17.33	17.60	17.71
Average latest arrival time						
in-vitro_1	9.52	9.98	10.62	9.12	9.07	9.05
in-vitro_2	7.29	7.84	8.29	7.31	7.32	7.32
protein_1	12.28	12.46	12.92	12.50	12.56	12.58

protein_2	7.44	7.75	8.13	7.68	7.76	7.80
Total number of used cells						
in-vitro_1	68.89	42.88	30.94	259.67	269.88	274.57
in-vitro_2	66.33	40.24	30.12	249.44	259.32	262.98
protein_1	447.89	270.64	193.92	1612.00	1672.48	1694.53
protein_2	251.44	154.92	112.47	916.00	949.60	961.00