Mathematical strategame theory and biomolecular mediated computing

Wang, Qi Wen

2014

Wang, Q. W. (2014). Mathematical strategame theory and biomolecular mediated computing. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/59222>

<https://doi.org/10.32657/10356/59222>

NANYANG
TECHNOLOGICAL
UNIVERSITY

MATHEMATICAL STRATEGAME THEORY
AND
BIOMOLECULAR MEDIATED COMPUTING

WANG QI-WEN

School of Mechanical and Aerospace Engineering

2014

MATHEMATICAL STRATEGAME THEORY
AND
BIOMOLECULAR MEDIATED COMPUTING

WANG QI-WEN

王岐闻

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2014

Abstract

Mathematical Strategame Theory (MST) and Biomolecular Mediated Computing (BMC) is a self-developed and fulfilled cross-disciplinary research field. The entire dissertation is made up of one theoretical work and one practical work.

The theoretical work of the dissertation is referring to Mathematical Strategame Theory. The self-created word “Strategame” is a combination of two words – the former part of the word “strategy” and the word “game”. The term “game” is originated from individual games in the framework of Parrondo’s paradox. Unlike conventional game theory, which involves strategic interactions among two or more players, Parrondo’s paradox describes a single player plays against a common opponent by following a series of predefined game rules. It has raised many attentions in media as it can be treated as a perfect demonstration of “beating the odds”. In addition to that, as claimed by its initiators, various effects in real life, where the randomness plays constructive roles, can be deemed as a Parrondian effect. On the opposite site, since the first appearance of Parrondo’s paradox, it has been continuously scrutinized by researchers, who pointed out the analogous paradoxical effect can be reproduced by using a relative simple game.

The objective of this part of the work is therefore lying on the investigation of Parrondo’s paradox and extending the framework of Parrondo’s paradox to other strategic applications. Several key findings have been raised through the investigation. The paradoxical effect generated by Parrondo’s paradox cannot be simply replaced by a relatively simple capital-dependent game. In addition, the Parrondo’s paradox can be mathematically analyzed in terms of a convex linear combination by retaining all the parameters inside a probability space. Based upon this theoretical foundation, it is able to construct other alternative paradoxes such as reverse Parrondo’s paradox, whereas two winning games can be combined in the generation of a losing game (compound game). Moreover, the paradoxical effect can be simply affected by three parameters: mixing sequences of two games in deterministically-mixed compound game, the predefined integer M (the switching mechanism lying behind capital-dependent Parrondo’s paradox), and non-linear combination of two games.

Although Parrondo’s paradox can be regarded as an appealing demonstration, which reveals the importance of developing the fuzzy logic gate, that is, a multi-valued logic which has a truth value

varying between 0 and 1, its potential application in contemporary financial market, however, is very restricted.

Nowadays, the performance of entire financial market is highly dependent on that of several huge financial institutes. Any problems occurs to any of these institutes may easily spell catastrophic consequences to majority of market participants, which may ultimately lead to the financial crisis. In the perspective of market participants, it is always optimal to seek out an avenue to reduce the potential risk of their investment. Even in the perspective of the government, it is always optimal to introduce one additional entity to contemporary financial market, which helps to redistribute and regulate the risks among these entities. Based on that, the developed conceptual model of option insurance is entirely different from the existing delta-neutral portfolio theory. Instead, it integrates the concepts of insurance and financial option. In addition to that, it is able to develop a simple mathematical model, which is based upon the inspirations obtained from Parrondo's paradox, to enhance its executional ability. By adopting the business strategy of matching, it is possible to determine the optimal pairs of call option and put option investors, which minimizes the potential loss of the option insurers while maximize its profitability.

The practical framework of the dissertation is referring to as Biomolecular Mediated Computing. As compared to conventional electronic mediated digital computing, biomolecular mediated computing is a relatively new computing paradigm, which employs artificially synthesized nucleic acids and analogs as information storing and processing substrate. At present time, biomolecular mediated computing remains at the stage of "proof of concept". However, there are several appealing features of biomolecular mediated computing. The main benefit of using biomolecular mediated computing is that different possible solution can be established all at once during the hybridization process in a parallel fashion. On the other hand, humans and most electronic computers have to solve the problem one process at a time. In addition to that, as information storing and processing substrate, DNA and its analogs have superior features for use in nanotechnology: minuscule size (diameter of 2nm), short structural repeat, and reasonable stiffness. Moreover, DNA and its analogs are cheap to obtain and energy-efficient.

The increasing capability of designing complex molecules and executional systems makes these models of computation increasingly of interest for many disciplines, including nanotechnology and biological engineering, as well as contributing to the understanding of various biochemical processes. At present stage, developing functional algorithm, which is able to exploit the massive parallelism, is one of the most important tasks. In this dissertation, three biomolecular have been proposed to tackle three classes of mathematical problems: automatic strategic assignment

apparatus, optimal route selection apparatus, and normative decision making apparatus. These toy models are crucial to map out the potential of future development of biomolecular mediated computing.

Acknowledgements

First and foremost, I want to thank my supervisor Associate Professor Jian-Jun Shu (舒建军). It has been an honor to be his Ph.D. student. His innovative way of thinking always inspires the new ideas in this highly crossed disciplinary study. I do appreciate all his contributions of time, concepts, and funding to make my Ph.D. experience productive and stimulating. The joy and enthusiasm he has for the research was motivational for me, even during tough times in the pursuit of Ph. D.

The members of the group have contributed immensely to my personal and professional time at Nanyang Technological University. The group has been a source of friendship as well as good collaboration. I am especially grateful for the person who stuck it out in graduate school with me: Kian-Yan Yong. We worked together on the development of biomolecular mediated computing devices.

Lastly, I would like to thank my family for all their love and encouragement. For my parents, Wei-Hua Quan (全薇华) and Wei-Jun Wang (王伟钧), who raised me with a love of science and supported me in all of my pursuits.

Wang Qi-Wen (王岐闻)

Nanyang Technological University

2014

Table of Contents

	<u>PAGE</u>
ABSTRACT	I
ACKNOWLEDGEMENTS	IV
TABLE OF CONTENTS	V
LIST OF FIGURES	IX
LIST OF TABLES	XIII
LIST OF SYMBOLS AND ABBREVIATIONS	XV
CHAPTER 1: INTRODUCTION	1
1.1 Background	2
1.2 Aims of This Study	7
1.3 Outline of This Dissertation	10
CHAPTER 2: LITERATURE REVIEW OF PARRONDO’S PARADOX	12
2.1 Background	13
2.2 Original Parrondo’s Paradox	16
2.2.1 Construction of Parrondo’s paradox	16
2.2.2 Working probabilities of original Parrondo’s paradox	19
2.2.3 Simulation of original Parrondo’s paradox	22
2.3 Dynamics of Capital-dependent Parrondo’s Paradox	25
2.3.1 Discrete-time Markov chain	25
2.3.2 Analysis of capital-dependent Parrondo’s paradox	30
2.4 Other Paradoxical Phenomenon	35
2.4.1 History-dependent Parrondo’s paradox	35
2.4.2 Cooperative Parrondo’s games	40
2.4.3 Buy-low-sell-random trading approach	40
2.4.4 Allison mixture	41
2.5 Issues surrounding Parrondo’s Paradox	42

CHAPTER 3: BEYOND PARRONDO’S PARADOX	46
3.1 Overview of Parrondo’s Paradox	48
3.2 Verification of Simple Capital-dependent Game	52
3.3 Mathematical Analysis of Parrondo’s Paradox	54
3.4 Alternative Parrondo’s Paradox	58
3.4.1 Reverse Parrondo’s paradox	58
3.4.2 Alternative Parrondo’s paradox	60
3.5 Effects of Other Parameters	69
3.5.1 Effect of predefined integer M	69
3.5.2 Effect of mixing sequence	71
3.5.3 Effect of non-linear combinations of two games	73
3.6 Conclusion	76
 CHAPTER 4: FINANCIAL OPTION INSURANCE	 78
4.1 Introduction	80
4.2 Business Model	83
4.2.1 Speculating insurance policies	83
4.2.2 Hedging insurance policies	88
4.3 Business Strategy of Matching	91
4.3.1 Three possible scenarios of matching	92
4.3.2 Matching strategy	94
4.3.3 Example	95
4.4 Option Insurance in Financial Market	98
4.4.1 Market acceptance	98
4.4.2 Option insurance profitability	99
4.4.3 Positive market effect	99
4.5 Conclusion	101
 CHAPTER 5: LITERATURE REVIEW OF BIOMOLECULAR COMPUTING	 102
5.1 Introductory Molecular Biology	105
5.1.1 A brief history of molecular biology	105

5.1.2 Two types of chemical bonds	106
5.1.3 Chemical structure of DNA and RNA	112
5.1.4 DNA topology	124
5.2 Theoretical Computer Science	140
5.2.1 Graph theory and combinatorial optimization problems	141
5.2.2 Boolean algebra	145
5.3 Developments of Biomolecular Mediated Computing	151
5.3.1 Limited-function biomolecular mediated devices	152
5.3.2 Biomolecular mediated logic circuits	160
 CHAPTER 6: BIOMOLECULAR STRATEGIC ASSIGNMENT APPARATUS	 169
6.1 Strategic Assignment Problem	171
6.2 Classical Solution	173
6.3 Methods	175
6.4 Implementation	179
6.5 Conclusion	186
 CHAPTER 7: BIOMOLECULAR OPTIMAL ROUTE SELECTION APPARATUS	 188
7.1 Optimal Route Selection Apparatus	190
7.2 Classical Solution	192
7.3 Methods	196
7.4 Implementation	198
7.5 Results and Discussion	206
7.6 Conclusion	208
 CHAPTER 8: BIOMOLECULAR NORMATIVE DECISION MAKING APPARATUS	 209
8.1 Normative Decision Making Apparatus	211
8.2 Methods	215

8.3 Information Encoding Scheme	219
8.4 Implementation	223
8.5 Conclusion	228
CHAPTER 9: CONCLUSIONS AND FUTURE RESEARCH	229
9.1 Contributions	230
9.2 Recommendations for Future Research	234
APPENDIX	237
REFERENCE	249

List of Figures

<u>FIGURE</u>	<u>PAGE</u>
1-1 Layout of the dissertation	10
2-1 Asymmetry of ratchet potential U_{ON} controlled by a single parameter α	13
2-2 Mechanics of Brownian flashing ratchet	14
2-3 Mechanics of titled Brownian flashing ratchet	15
2-4 Representation of flat and ratchet potential in the framework of Parrondo's paradox	16
2-5 Foundation of capital-dependent Parrondo's paradox	18
2-6 Capital-dependent Parrondo's paradox	23
2-7 Deterministically mixed games in capital-dependent Parrondo's paradox	24
2-8 Example of transition diagram	27
2-9 Example of transition probability matrix	28
2-10 Transition diagram of Parrondo's game A	30
2-11 Transition diagram of Parrondo's game B	31
2-12 Transition diagram of Parrondo's module game B	33
2-13 Foundation of history-dependent Parrondo's paradox	36
2-14 History-dependent Parrondo's paradox	37
2-15 Transition diagram of history-dependent Parrondo's paradox	38
2-16 Simplified transition diagram of game B of history-dependent Parrondo's paradox	39
2-17 Volatility pumping model	41
2-18 Probability space of capital-dependent Parrondo's paradox	44
3-1 Frameworks of capital-dependent and history-dependent Parrondo's paradox	50
3-2 Simulation of capital-dependent and history-dependent Parrondo's paradox	51
3-3 Simulation of simple capital-dependent game	52
3-4 Probability space of capital-dependent Parrondo's paradox	54

3-5	Probability space of reversed Parrondo's paradox	58
3-6	Simulation of reverse Parrondo's paradox	59
3-7	Probability space contains all probabilities of eight combinations of games	61
3-8	Game rule and simulation result of scheme #2	63
3-9	Game rule and simulation result of scheme #3	63
3-10	Game rule and simulation result of scheme #4	64
3-11	Game rule and simulation result of scheme #5	64
3-12	Game rule and simulation result of scheme #6	65
3-13	Game rule and simulation result of scheme #7	65
3-14	Game rule and simulation result of scheme #8	66
3-15	Probability space of original and modified Parrondo's paradox	70
3-16	Simulation of modified capital-dependent Parrondo's paradox	71
3-17	Simulation of Parrondo's paradox of different mixing sequences	72
3-18	Non-linear and linear combinations of two games in probability space	74
3-19	Simulation of non-linear combinations of two games	74
4-1	Risk and reward graph for call option and put option speculators	85
4-2	Transaction among three-entity structure with financial options and option insurance	86
4-3	A win-win situation for both speculators and third entity under different scenarios	87
4-4	Risk and reward graph for speculators with or without option insurance	88
4-5	Risk and reward graph for portfolio managers with and without option insurance	90
4-6	Three possible scenarios of matching between different moneyness of calls and puts	92
4-7	Demonstration of matching strategy among call option and put option speculators	96
5-1	Chemical structure of 2'-deoxyribose	112
5-2	Formation of nucleotide by a hypothetical double dehydration process	113
5-3	Purine and pyrimidine	114
5-4	Chemical structure of a single polynucleotide chain	115

5-5	The Watson-Crick base pairing	116
5-6	Minor and major groove in B-form DNA	117
5-7	Exposed chemical groups in major and minor grooves of A:T and G:C pairs	118
5-8	Anti-parallel orientation of double-stranded DNA	119
5-9	Structural differences between DNA and RNA	122
5-10	Various stem loop structure of RNA	123
5-11	The G:U base pair in RNA structure	123
5-12	Topological states of covalently closed, circular DNA (cccDNA)	126
5-13	A schematic representation of standard gel electrophoresis configuration	130
5-14	A schematic representation of creating ssDNA of specified sequence	133
5-15	A schematic representation of one cycle of PCR process to multiply dsDNA	134
5-16	A schematic description of cleaving mechanisms of restriction endonucleases	136
5-17	A schematic representation of two types of ends as a result of cleaving	137
5-18	A schematic comparison between two identical DNA solutions with and without the addition of DNA ligase after hybridization process	138
5-19	Non-directed graph and directed graph	141
5-20	Shortest path problem and shortest paths from vertex 1 to any other vertices in G	143
5-21	Minimal spanning tree problem and its corresponding solution	144
5-22	Assignment problem and its corresponding solution	145
5-23	Venn diagram of set x and its complement	146
5-24	Intersection and union of sets a and b	147
5-25	Circuit symbol and truth table of NOT function	148
5-26	Truth table for all possible two input bits Boolean functions	148
5-27	Circuit symbol and truth table of AND function	148
5-28	Circuit symbol and truth table of OR function	149
5-29	Circuit symbol and truth table of NAND and NOR function	149
5-30	Circuit symbol and truth table of XOR function	150
5-31	Circuit symbol and truth table of COIN function	150
5-32	Truth table of implication functions	150
5-33	Example of primal graph and its complementary graph	154
5-34	DNA representation of all possible non-negative one-digit binary integers	157

5-35	DNA representation of all possible nonnegative two-digit binary integers	157
5-36	DNA operation of binary addition of 11 and 01	158
5-37	The graph G_n which encodes two-bit numbers	162
5-38	A schematic representation of illusionary polymerase chain reaction in DNA-based logic AND gate construction	166
5-39	A schematic representation of toe-hold strand displacement method	167
6-1	Bipartite graph of Tian Ji's horse racing	171
6-2	The strategic assignment problem and its complementary graph	172
6-3	Solution of strategic assignment problem by employing classical algorithm	174
6-4	Demonstration of DNA encoding protocol	176
6-5	A typical example of possible paths generated at the end of ligation process	180
6-6	All paths starts from vertex a constitutes a tree diagram	183
6-7	Estimated pattern of DNA migration as a result of electrophoresis	184
6-8	Restriction enzyme ScaI cleaves dsDNA into two duplexes	184
6-9	Excessive edges versus insufficient edges in strategic assignment problem	187
7-1	Optimal route selection problem and its abstract graph G	190
7-2	Modified graph G and shortest paths from vertex 1 to other vertices in G	193
7-3	Modified graph G and shortest paths from vertex 6 to other vertices in G	195
7-4	Process flow chart of DNA-based optimal route selection apparatus	196
7-5	Schematic representation of edge encoding scheme	199
7-6	2D 12% native polyacrylamide gel image and corresponding optimal paths	207
8-1	Regional map of a strategic board game – risk	212
8-2	Expected utility of three options	213
8-3	Decision tree of regional military conflict	215
8-4	A schematic representation of toe-hold strand displacement method	216
8-5	Probability controlled by utilizing toe-hold mediated strand displacement mechanism	217
8-6	DNA sequence design motif	220

List of Tables

<u>TABLE</u>	<u>PAGE</u>
1-1 Relationship between quantities in Flashing Brownian ratchet and original Parrondo's paradox	2
1-2 Comparison between electronic-based and biomolecular mediated computing	5
2-1 Relationship between various quantities used in capital-dependent Parrondo's paradox and Brownian flashing ratchet	25
2-2 Game rules of game B of History-dependent Parrondo's paradox	35
3-1 Different combinations of two winning and/or losing games	60
3-2 Summary of simulation results corresponding to different schemes of combinations	66
4-1 Reimbursement clauses for speculators and portfolio managers	90
5-1 Second genetic code in major and minor groove of base pairs	117
5-2 Nearest neighbor thermodynamic parameters for Watson-Crick base pairs	121
5-3 Conversion table of decimal and binary number	156
5-4 Truth table of formula F	161
5-5 DNA in test tube and corresponding DNA sequences	163
6-1 DNA sequence design of vertices in set X	175
6-2 DNA sequence design of vertices in set Y	176
6-3 DNA sequence design of directed edges (i, j)	177
6-4 DNA sequence design of directed backtracking edges (j, i)	177
6-5 DNA sequence design of stop vertices	178
7-1 Vertices and corresponding single-stranded DNA (ssDNA) sequences	198

7-2	Edges and corresponding double-stranded DNA (dsDNA) sequences	200
7-3	DNA primers sequence design	200
7-4	Selected volume of individual DNA solutions	201
7-5	12% native polyacrylamide gel recipe	204
8-1	The probability of attacker winning a battle in risk	214
8-2	The concentration ratios of corresponding DNA sequences	223
8-3	Selected volume of individual DNA solutions	224

List of Symbols and Abbreviations

Chapter 1: Introduction

U_{on}	:	Ratchet potential
U_{off}	:	Flat potential
τ_{on}	:	Time ON
τ_{off}	:	Time OFF
ε	:	Biasing parameter
X_n	:	Capital amount
α	:	Potential slope

Chapter 2: Literature Review of Parrondo's paradox

U_{on}	:	Ratchet potential
U_{off}	:	Flat potential
α	:	The location along the axis, which is used to determine ratchet potential
L	:	Arbitrary length along the axis
P_{fwd}	:	Probability of moving forward
P_{bck}	:	Probability of moving backward
P_1	:	Winning probability of game A
P_2	:	Winning probability of scenario 1 of game B
P_3	:	Winning probability of scenario 2 of game B
M	:	Predefined integer, which
γ	:	Mixing parameter
q_1	:	Winning probability of compound game 1
q_2	:	Winning probability of compound game 2
ε	:	Biasing parameter
$[a, b]$	:	Mixing sequences (a: number of game A, b: number of game B)
P_{ij}	:	Transition probability from state i to state j

X_n	:	Random variable with Markov properties
$\mathbf{p}^n$	:	Transition matrix
$\mathbf{p}'(n)$	:	State distribution vector
$\boldsymbol{\pi}$	:	Steady distribution vector
D	:	Normalization constant
$\mathbf{I}$	:	Identity matrix
Y_n	:	Extracted random variable with Markov property
$P_{B,win}$	:	Winning probability of game B
E	:	Expected return

Chapter 3: Beyond Parrondo's Paradox

$C(t)$	:	Instantaneous capital at t number of games played
M	:	Predefined integer (switching mechanism of two scenarios of game B)
ε	:	Biasing parameter
p_1	:	Winning probability of game A
p_2	:	Winning probability of scenario 1 of game B
p_3	:	Winning probability of scenario 2 of game B
p_{c_1}	:	Winning probability of compound game where $C(t) \bmod M = 0$
p_{c_2}	:	Winning probability of compound game where $C(t) \bmod M \neq 0$
γ	:	Mixing parameter

Chapter 4: Financial Option Insurance

ITM	:	In-the-money
ATM	:	At-the-money
OTM	:	Out-of-the-money
S_0	:	Initial price of underlying stock
C	:	Premium price of particular call option
P	:	Premium price of particular put option
S_{t_1}	:	Price of underlying stock at exercise date of option (price increase)

S_{t_2}	:	Price of underlying stock at exercise date of option (price decrease)
EV	:	Expected value
K_C	:	Strike price of call option
K_P	:	Strike price of put option

Chapter 5: Literature Review of Biomolecular Computing

K_{eq}	:	Constant of equilibrium
$conc^A$	:	Concentration of atom A
$conc^B$	:	Concentration of atom B
$conc^{AB}$	:	Concentration of atom AB
G	:	Gibbs free energy
ΔS_{total}	:	Change of total entropy in a completely isolated system
ΔS_{ini}	:	Change of internal entropy
ΔS_{ext}	:	Change of external entropy
Q	:	Thermal energy
T	:	Temperature of isolated system
ΔH	:	Change of enthalpy
R	:	Universal gas constant
ATP	:	Adenosine triphosphates
ADP	:	Adenosine diphosphates
mer	:	Unit for single-stranded DNA in terms of number of nucleotide(s)
$ssDNA$	:	Single-stranded DNA
$dsDNA$	:	Double-stranded DNA
bp	:	Unit for double-stranded DNA in terms of number of base pairs
T_m	:	Melting temperature in Kelvin (K)
ΔS	:	Standard entropy
ΔT	:	Standard enthalpy
C_T	:	Total concentration of DNA strand
N	:	Arbitrary constant
$mRNA$	:	Messenger RNA
$tRNA$	:	Transfer RNA

$cccDNA$	:	Covalently-closed circular DNA
Lk	:	Linking number
Tw	:	Twist
Wr	:	Writhe
Lk^0	:	Linking number of fully relaxed cccDNA
σ	:	Super-helical density
α, β	:	Certain DNA sequences
V	:	Set of vertices
E	:	Set of edges
(i, j)	:	Non-directed edge between vertex i and vertex j
e_{ij}	:	Non-directed edge between vertex i and vertex j
O_{ij}	:	Oligonucleotide
D_{ij}	:	Relative concentration of one specific oligonucleotide
C_{ij}	:	Cost

Chapter 6: Biomolecular Strategic Assignment Apparatus

X	:	Subset of vertices
Y	:	Subset of vertices
$(i, j), e_{ij}$	:	Non-directed edge between vertex i and vertex j
G	:	Graph
G'	:	Complementary of graph G
M	:	Matching set of edges
E	:	Edge
P	:	Path
$ssDNA$	:	Single-stranded DNA
$dsDNA$	:	Double-stranded DNA
m	:	Total number of vertices in X
n	:	Total number of vertices in Y

Chapter 7: Biomolecular Optimal Route Selection Apparatus

$ssDNA$	:	Single-stranded DNA
$dsDNA$	:	Double-stranded DNA
GPS	:	Global positioning system
G	:	Graph
i, j	:	Any vertices in graph G
(i, j)	:	Directed weighted edge from starting vertex i to terminating vertex j
PL	:	Permanent label
TL	:	Temporary label
L_k	:	Length/weight of edge
$\widetilde{L}_k$	:	Temporary label of length of edge
$\emptyset$	:	Null set
k	:	Label of the edge
T_m	:	Melting temperature

Chapter 8: Biomolecular Normative Decision Making Apparatus

O_i	:	A set of predefined options
U	:	Expected utility
P_j	:	Probability of each possible outcome
x_j	:	Particular outcome
NT	:	Abbreviation of Northwest Territory
ON	:	Abbreviation of Ontario
QU	:	Abbreviation of Quebec
$ssDNA$	:	Single-stranded DNA
$dsDNA$	:	Double-stranded DNA
α, β	:	One specific region of single-stranded DNA
α', β'	:	Complementary region of target region of single-stranded DNA
E_{ij}	:	Directed edge from vertex i to vertex j

Th_{p_j}	:	Threshold DNA
T	:	Terminating vertex
TTi	:	Test tube number i

Chapter 1

Introduction

Mathematical Strategame Theory (MST) and Biomolecular Mediated Computing (BMC) is a self-defined and fulfilled cross-disciplinary research, which involves knowledge from previous seemingly unrelated disciplines. As described by the title of research, the entire framework can be further separated into two sections, theoretical framework and practical framework.

Mathematical Strategame Theory is the theoretical framework of entire research. The self-created word “strategame” is a combination of two words – the former part of the word “strategy” and the word “game”. The term “strategy” reveals the developed mathematical model is always associated with the situations where strategically thinking is demanded. The term “game” is originated from the individual games in Parrondo’s paradox, which is believed to be a special complementary of contemporary game theory. This part of the dissertation is set out to investigate the reason behind the counterintuitive phenomenon produced by Parrondo’s paradox, and extend the inspirations obtained from Parrondo’s paradox to other potential situations.

On the other hand, Biomolecular Medicated Computing is the practical framework of entire research. It is a relatively new computing paradigm as compared to contemporary electronic mediated computing. It employs nucleic acids and analogous biomolecular structures as information storing and processing substrate to tackle computational problems. This part of the dissertation is aiming to create several conceptual designs of biomolecular mediated devices, which is analogous to limited-function early computers. These developed devices may provide some insights to future research direction of potential biomolecular computer.

In this chapter, it begins with an overall view of the entire research (section 1.1). The objective and the scope of the work are described in aims of this study (section 1.2). In the last section (section 1.3), the outline of the entire dissertation is clarified.

1.1 Background

Mathematical Strategame Theory

The theoretical framework of the dissertation, Mathematical Strategame Theory, is closely related to a special complementary of contemporary game theory [1], namely, Parrondo's paradox. It is named after its initiator Spanish physicist Juan Manuel Rodriguez Parrondo. The original insight of Parrondo's paradox is to mathematically formulate a counterintuitive physical phenomenon observed during the operation of a mechanical device called flashing Brownian ratchet [2 – 3] in terms of a special combination of two coin-tossing games, which is referred to as game A and game B. This is the most original version of Parrondo's paradox, which is also treated as capital-dependent Parrondo's paradox [4 – 5].

As specified in [6], the relationship between quantities in capital-dependent Parrondo's paradox and flashing Brownian ratchet can be simply described by using Table 1 – 1.

Quantity	Flashing Brownian Ratchet	Original Parrondo's paradox
Source of potential	Electrostatic, Gravity	Rules of the games
Switching	U_{on} and U_{off} applied	Games A and B played
Switching durations	τ_{on} and τ_{off}	a and b
Duration	Time	Number of games played
Biasing	Macroscopic field gradient	Parameter ε
Transport quantity	Brownian particles	Capital of the player
Measurable output	Displacement x	Capital amount X_n
External energy	Switching U_{on} and U_{off}	None
Potential shape	Depends on α	Probabilities P_1, P_2, P_3 and integer M
Mode of analysis	Fokker-Planck equation	Discrete-time Markov chain

Table 1 – 1: *Relationship between quantities in Flashing Brownian ratchet and original Parrondo's paradox [5 – 6]*

In capital-dependent Parrondo's paradox, game A is a simple zero-order memoryless game, which simply states the individual trials are not affected by the results obtained from previous trials. Game B is a condition-based game, which consists of two scenarios. Each scenario is represented by a biased coin. Deciding which coin to be played is merely depending on whether

the instantaneous capital of the player is divisible or non-divisible by a predefined integer. The concept of original Parrondo's paradox specifies that both game A and game B are losing games in the perspective of the gambler as the capital is decreased with respect to the advance of number of games played. However, a winning expectation is produced when two games are played in a mixed manner.

The systematic approach of Parrondo's paradox was instituted by Gregory P. Harmer and Derek Abbott by proving the concepts of capital-dependent Parrondo's paradox [4 – 5], formulating the mathematical expressions associated with Parrondo's paradox [6 – 7], and extending the capital-dependent version to history-dependent version together with its original initiator [8]. As inspired by Parrondo's paradox, the concept has been further extended into the cooperative version in 2001 [9]. One year after, the history-dependent Parrondo's paradox has been successfully transferred into a quantum game by performing SU operations [10 – 11]. At the same time, the strategy of Parrondo's paradox was extended into a typical financial application, whereas the developed toy model, namely buy-low-sell-random trading approach, was claimed to outperform the conventional buy-and-hold trading approach [12]. In addition to that, many other paradoxical phenomena, which are based upon the framework of Parrondo's paradox, have been proposed by various researchers [19 – 25]. Most recently, a completely new form of Parrondo's paradox, namely Allison mixture, has been developed by A. Allison. As specified in [26], it simply states the resulting sequence generated by mixing two sequences of random numbers is not completely random unless certain conditions are fulfilled.

Parrondo's paradox has raised many attentions in media as it has tremendous potentials in describing strategy of turning two unfavorable situations into a favorable one [27]. However, the concept has been continuously scrutinized by researchers since its first appearance [28 – 32].

Biomolecular Mediated Computing

The practical framework of the dissertation, Biomolecular Mediated Computing, is a new computing paradigm, which employs artificially synthesized nucleic acids and analogous biomolecular structures as information storing and processing substrates to tackle computational problems.

Since the first electronic mediated general purpose microcomputer was introduced at the beginning of 1970s, improving the computational ability has always been the top-ranked concern among researchers from many related disciplines all over the world. In order to confront the growing demands for its computational ability, processing speed as well as parallelism, the size of individual transistor elements has to be reduced, such that, allows additional individual processing elements to be packed onto the same sized chip. The increasing packing density always led to problems, including power consumption and heat dissipation. Moreover, the employment of silicon-based material in manufacturing digital computers always has negative health and environmental influence during its manufacturing and recycling processes [33]. Despite that, the entire semiconductor industry is quickly approaching its physical constraints as predicted by Moore's Law [34].

Therefore, the researchers from various disciplines are engaged in exploring alternatives to contemporary electronic-based digital computer. Among various intriguing approaches, biomolecular mediated computing seems to be the most feasible method due to its appealing features as a perfect nanomaterial, including its minuscule size, short structural repeat, and reasonable stiffness [35].

The entire field is depending on two important discoveries – the exact structure of DNA molecules in 1953 [36] and the fundamental biological mechanisms in late 1970s. The biomolecular mediated computing was arguably initiated by Leonard M. Adleman, who demonstrated the employment of artificially synthesized DNA as a simple computational device to solve a seven-vertex directed Hamiltonian path problem in 1994 [37]. The significance of his work is the discovery of various connections between biological and computational systems. The immediacy and feasibility of biomolecular mediated computing revealed by his work was absent in previous works proposed by other researchers. In addition to that, the major advantages and drawbacks of biomolecular mediated computing over electronic-based computing are compared in Table 1 – 2.

Characteristics	Electronic-based computing	Biomolecular mediated computing
Processing speed	10^8 to 10^{14} operations per sec	10^{14} to 10^{20} operations per sec (during the DNA hybridization phrase)
Energy	10^9 operations per Joule	2×10^{10} operation per Joule
Info. storage	1 bit per 10^{12} cubic nm	1 bit per cubic nm
Computing Architecture	Effective for single operation, multiple cores of CPU for multiple operations at one time (up to 4 operations)	Ineffective for single operation, but very effective for massive parallel operations

Table 1 – 2: *Comparison between electronic-based and biomolecular mediated computing*

Shortly after Leonard M. Adleman's major contribution, it was theoretically modified and extended to the Boolean logic operations. It was applied to solve a famous problem, namely, satisfactory problem [38]. Since that, the researches of biomolecular mediated computing can be broadly classified into two different categories: limited-function biomolecular mediated devices and biomolecular mediated circuits.

The first category of the researches focuses on the development of limited-function biomolecular mediated devices. The proposed conceptual models maximize the advantage of biomolecular. There are many mathematical problems which have been subsequently solved by the developed models, including maximal clique problem [39], addition problem [40], shortest path problem [41], Chinese postman problem [42], maximum matching problem [43], traveling salesman problem [44], maximum cut problem [45], bin-packing problem [46], and so on [47]. The only drawback is that their applications are usually problem specific. In other words, one developed conceptual model is only efficient in handling one specific class of mathematical problems.

The second category of the researches emphasize on the development of biomolecular mediated logic circuit, which is assembled by a series of functional logic gates. Analogous to electronic-based logic gates, these developed logic gates are capable of performing simple Boolean logic operations. The researchers, who are enrolled in this category of researches, generally believe that designing appropriate biomolecular logic circuits is a perfect starting point in the development of general-purpose biomolecular mediated practical computer. The formal report of molecular-based logic gate was firstly initiated by C. P. Collier in 1999 [48]. One year later, the first functional universal biomolecular logic gate was proposed by N. C. Seeman et al. [49]. In addition to that, there are several intriguing designs of biomolecular mediated logic gates have been perfectly

summarized in [50 – 52]. The development of surface-based techniques [53] and representing the output signal in terms of fluorescent signals can be deemed as a significant advancement of biomolecular mediated logic circuits. Such improvement was reflected in the work of A. Saghatelian, which is known as DNA-based photonic logic gates [54].

Since that, there are many conceptual models of biomolecular mediated logic gates have been proposed. In general, these conceptual models can be further classified into two categories: biochemically mediated logic circuits [55 – 62] and strand displacement mediated logic circuits [63 – 80].

1.2 Aims of this study

The key objectives of the study can be broadly summarized in terms of following four statements:

- Investigation of Parrondo's paradox and other generated paradoxical phenomenon
- Extending the inspirations obtained from Parrondo's paradox into the development of conceptual models, which can be employed in the financial industry.
- Development of conceptual biomolecular device in solving specific class of mathematical problems, which maximize the ability of parallel computing.
- Experimental demonstration of feasibility of developed biomolecular mediated apparatus.

By referring to the list of major objectives, the aims of this study become obvious. As mentioned in the beginning of the chapter, the scope of the work can be separated into two sections, namely theoretical framework (Mathematical Strategame Theory) and practical framework (Biomolecular Mediated Computing).

The theoretical framework is closely associated with Parrondo's paradox, a special complementary to contemporary game theory. It produces a typical counterintuitive phenomenon, in which two individual losing strategies can be combined in the production of a winning expectation. As claimed by its initiator, various effects in real life, where the randomness plays constructive roles, can be regarded as Parrondo's effect. However, there are several researchers who have pointed out that Parrondo's paradox is not paradoxical as its counterintuitive phenomenon can be generated by using a relatively simple game. Therefore, the entire work is broke down into several testing scenarios. At the beginning, it is going to testify whether the analogous paradoxical effect can be simply generated by a relatively simple capital-dependent game. After that, Parrondo's paradox is mathematically analyzed by leaving all parameters and rule of game inside the probability space. Beyond that, it is possible to generate other paradoxical effects by varying different combinations of two games. In addition to that, by varying different values of predefined integer M (switching mechanism), mixing sequences of deterministically mixed compound game, and the nonlinear combination of two games, it is aiming to verify whether the paradoxical effect can be affected by these changes. Besides that, the framework of Parrondo's paradox is extended into the applications of financial industry.

The practical framework of the study is emphasizing on the development of biomolecular mediated computing. As the cross disciplinary research is a new topic, a thorough guide of this field is provided, which broadly summarizes various definitions, related biological mechanisms, and biomolecular mediated computing. Based on the understanding of the discipline, there are three conceptual designs of biomolecular mediated devices known as strategic assignment apparatus, optimal route selection apparatus, and normative decision making apparatus.

The solution of assignment problem was firstly proposed by Shiyong Wang et al [43]. The central idea of developed model is to validate whether there exists an argument path P within a bipartite graph with given matching set M . The proposed model is both theoretically infeasible and fails to exploit the advantage of biomolecular mediated computing, that is, massive parallelism. The developed model requires the collection of double-stranded DNA which is previously attached to the surface. However, there is no existing laboratory technique, which is able to collect attached molecules from the surface, and hence results in the loss of information. In addition to that, the proposed model is over relying on the human involvement. Verifying whether there exists an augmenting path does not really solve the strategic assignment problem. Besides, the solution at the end of each verification process cannot be recycled.

In order to overcome these limitations presented in original model, a new strategic assignment apparatus is proposed. The proposed model provides instant result instead of manipulating the model to verify whether there exists an argument path. In other words, the proposed apparatus is capable of scaling up as it requires least amount of human involvement.

The biomolecular optimal route selection apparatus and normative decision making apparatus are new to the field. The optimal route selection apparatus is a function unit embedded in contemporary Global Positioning System (GPS). The emphasis of optimal route selection apparatus is to experimentally demonstrate the feasibility of biomolecular mediated computing. For most developed models in the field, one typical laboratory setup is only capable of solving one particular problem. However, in this case, it allows all information, which is represented in terms of encoding sequences, to be stored within the database (DNA solution). The only thing the user should manipulate is to provide the input, the starting vertex (forward primer sequence) and terminating vertex (reverse primer sequence), during the Polymerase Chain Reaction (PCR) while maintaining the remaining setups unmodified. One significant advantage of proposed apparatus is to allow the scaling up of the device by simply increase the size of its database.

The emphasize of biomolecular normative decision making apparatus is to reveal the feasibility that the synthesized DNA can be employed as information storage and processing substrate in making rational decision based upon expected utility hypothesis. The toe-hold mediated strand displacement mechanism, which was used by Lulu Qian in [65], is believed to be a perfect representation of probabilities leads to different outcomes. At present time, biomolecular mediated computing is still at the stage of “proof of concept”, the objective of these developed apparatus is aiming to map out potentials of such computing methodology for the future development of biomolecular mediated computing.

1.3 Outline of this dissertation

The outline of the entire dissertation is surrounded with two major frameworks, namely, Mathematical Strategame Theory and Biomolecular Mediated Computing. The overall structure and the interconnections between these two fields are visually summarized in Figure 1 – 1.

Figure 1 – 1: Layout of the dissertation

The entire dissertation is presented as follows. It begins with a brief introduction, Chapter 1, on the concepts of Mathematical Strategame Theory and Biomolecular Mediated Computing. The introduction broadly summarizes the background information of these two fields, described the objectives of the study and methodologies used to achieve these targets.

Part I, Mathematical Strategame Theory, consists of three chapters. Chapter 2 is the literature review of Parrondo's paradox. The idea of Mathematical Strategame Theory is introduced, followed by a thorough investigation of two versions of Parrondo's paradox and other paradoxical

phenomenon proposed by various researchers. Chapter 3 and Chapter 4 contain the research result on mathematical strategame theory. Chapter 3 investigates Parrondo's paradox by manipulating various parameters mentioned in its framework. Chapter 4 discusses possible application of this mathematical strategame theory to financial option insurance. A completely new conceptual model of financial option insurance is proposed. It integrates two concepts of financial derivative (option) and insurance by introducing one addition entity, known as the third entity, into the contemporary two-entity financial structure. To enhance the executorial ability of the option insurer, the third entity, a specifically designed mathematical model, which is referred to as business strategy of matching, is proposed.

Part II, Biomolecular Mediated Computing, consists of four chapters. It begins with a conclusive guide, as shown in Chapter 5, which provides the essential knowledge, including various definitions, and biological mechanisms, prior to the cross disciplinary study of Biomolecular Mediated Computing. In Chapter 6, conceptual model of biomolecular mediated device is described in handling a famous combinatorial optimization problem, known as strategic assignment problem. The selected case, which is used for demonstrate purpose, is analogous to the mathematical model developed for financial option insurance. In Chapter 7, a conceptual model of biomolecular mediated optimal route selection apparatus is proposed. The function of developed device is analogous to a functional unit embedded in contemporary Global Positioning System (GPS) navigator. Furthermore, the proposed model is experimentally demonstrated to obtain optimal path in an abstract graph. The detailed experimental procedures and the obtained results are presented in the chapter. In Chapter 8, the idea of biomolecular mediated computing is further conceptually extended into the discipline of decision theory. The novel-based apparatus is described in handling a regional military conflict of a famous board game, the risk®.

Chapter 2

Literature Review of Parrondo's Paradox

The study of probability can be traced back to 16th century. The first systematic treatment of probability was initiated by Gerolamo Cardano, an Italian mathematician, who attempted to analyze the game of chance.

Game of chance is generally referring to as the process that involves the random events or variables. The simplest demonstration of games of chance can be produced by tossing an unbiased coin in the generation of one of two possible outcomes – “head” and “tail”.

Parrondo's paradox named after Spanish physicist J. M. R. Parrondo, who had successfully generated a positively biased outcome by combining two negatively biased coin-tossing games, which is currently referring to as the Parrondo's games [3]. Parrondo's games demonstrated an apparently counter-intuitive phenomenon, which consists of two individual gambling games, namely, game A and game B. If any player plays either game A or game B individually against a common opponent by following specified game rules, the longer the gambler plays the more capital he loses as a result of playing. Therefore, in the perspective of the player, both game A and game B are losing games. However, if the player chooses an alternative strategy by playing both games in a periodical or totally random order, he starts to win the money by repeating the strategy in the long run. The generated phenomenon is referring to as the Parrondo's paradox [4 – 8].

This chapter is organized as follows: the inspirations, which lead to the development of Parrondo's paradox, are broadly described in section 2.1. The derivation of original Parrondo's paradox, capital-dependent Parrondo's paradox, is reproduced in section 2.2. The dynamics of capital-dependent Parrondo's paradox are mathematically analyzed in section 2.3. Based on the inspirations obtained from Parrondo's paradox, other paradoxical phenomenon were generated by various researchers, these generated phenomenon are selectively concluded in section 2.4. The issues, which proposed by the critics of Parrondo's paradox, are summarized in section 2.5.

2.1 Background

The ratchet and pawl device, also known as Brownian ratchet, was firstly introduced in the last century. It was formally analyzed by Polish physicist M. Smoluchowski [13] and popularized by American physicist R. Feynman [14]. The focus of the research in this specific discipline is to exploit the Brownian motion by converting it into a directed motion, without any use of macroscopic forces or gradients. Several workable mechanisms have been developed so far. The intriguing phenomenon generated by one of these mechanisms, Brownian flashing ratchet (or simply flashing ratchet) [15], inspires the development of Parrondo's paradox.

The entire system of flashing ratchet consists of two one-dimensional potentials, known as ratchet potential U_{on} and flat potential U_{off} . For ratchet potential U_{on} , it is determined by a single parameter α , where $\alpha \in [0, 1]$, as shown in Figure 2 – 1.

Figure 2 – 1: *Asymmetry of ratchet potential U_{on} controlled by a single parameter α*

For Brownian flashing ratchet, the ratchet potential U_{on} is asymmetric, whereas the corresponding parameter $\alpha < \frac{1}{2}$. In the equilibrium state, where the potential height is larger than the thermal noise, the Brownian particles are located in the potential minima of ratchet potential,

for instance, they are located at origin 0, as indicated in Figure 2 – 2 (a). Once the system potential is switched from ratchet potential U_{on} to flat potential U_{off} , the Brownian particles, which are trapped at potential minima is freely diffused along the flat potential. The concentration of Brownian particles can be described by using a normal distribution curve with its mean located at the potential minima, *i.e.*, the origin 0, as specified in Figure 2 – 2 (b). When the ratchet potential U_{on} is switched back again, there is a large percentage of Brownian particles moving forward and finally rest at the potential minima L . Such percentage, denoted by P_{fwd} , is proportional to the shaded area of normal distribution curve (in green), as shown in Figure 2 – 2 (b). At the same time, there also exists certain percentage of Brownian particles moving backward and ultimately rest at the potential minima $-L$. Such percentage, denoted by P_{bck} , is proportional to the shaded area of normal distribution curve (in orange), as indicated in Figure 2 – 2 (b). Since the state of asymmetry of ratchet potential is governed by a single parameter α , therefore, the percentage of Brownian particles moving forward is therefore larger than the probability of Brownian particles moving backward, *i.e.*, $P_{\text{fwd}} > P_{\text{bck}}$. In a simple words, as shown in Figure 2 – 2, the percentage of Brownian particle moving forward is greater than that of moving backward in each toggling of two potentials. Therefore, the net motion of Brownian particles is moving forward through toggling between two potentials, U_{on} and U_{off} .

Figure 2 – 2: *Mechanics of Brownian flashing ratchet*

Suppose these two potentials, ratchet potential U_{on} and flat potential U_{off} , are one-sided tilted to a calculated height, such that, once the state is held on either potential, ratchet potential U_{on} or flat potential U_{off} , the Brownian particles have a net motion of moving backwards, as shown in Figure 2 – 3 (a) and (b). However, once these two potentials are toggled periodically, a counter-intuitive phenomenon can be observed, that is, the net motion of Brownian particles are moving forward instead of backward, as indicated in Figure 2 – 2 (c).

Figure 2 – 3: Mechanics of titled Brownian flashing ratchet

The phenomenon generated by tilted Brownian flashing ratchet inspires the development of Parrondo's paradox [3]. Initially, the purpose of J. M. R. Parrondo was to mathematically represent the mechanics of titled Brownian flashing ratchet in terms of a combination of two simple coin-tossing games, which leads to the original version of Parrondo's paradox, also known as capital-dependent Parrondo's paradox [4 – 7]. In subsequent section, section 2.2, a derivation of original Parrondo's paradox is reproduced.

2.2 Original Parrondo's Paradox

At first glance, these two concepts, the titled Brownian flashing ratchet and capital-dependent Parrondo's paradox are seemingly dissimilar. In fact, there are several connections between these two concepts: the Brownian particles of flashing ratchet is analogous to the capital of the player in this specific gambling games, and the source of physical potential of flashing ratchet can be treated as the rule of the gambling games.

The entire derivation can be further divided into two subsections – framework construction (2.2.1) and the estimation of working ranges of probabilities of two gambling games (2.2.2).

2.2.1 Construction of Parrondo's paradox

As specified in Figure 2 – 2 and Figure 2 – 3, the Brownian flashing ratchet consists of two energy potentials – ratchet potential U_{on} and flat potential U_{off} . Suppose the forward movement of Brownian particles is analogous to the direction of capital winning. The connections between these two systems can be simply expressed in Figure 2 – 4.

Figure 2 – 4: Representation of flat and ratchet potential in the framework of Parrondo's paradox

For flat potential U_{off} in the flashing ratchet, as shown in Figure 2 – 4, the conversion process is relatively straightforward. In an unbiased condition, the flat potential simply allows the diffusion of Brownian particles equally in both directions. The concentration of the Brownian particles can be described by using a normal distribution curve with its mean located at the center of the flat potential. If the flat potential is inclined to the right, all diffused Brownian particles have tendency of moving forward, as a result, the mean of normal distribution curve is shifted to the right. On the contrary, if the flat potential is inclined to the left, all diffused Brownian particles have tendency of moving backward, such that, the mean of normal distribution curve is shifted to the left.

The flat potential U_{off} of flashing ratchet can be formulated by a simple zero-order memoryless coin-tossing game, namely game A, with a single coin (coin 1) of winning probability P_1 , and losing probability $1 - P_1$. Game A is said to be a winning game in the perspective of player if $P_1 > 0.5$. On the other hand, game A is said to be a losing game if $P_1 < 0.5$.

The ratchet potential U_{on} of flashing ratchet, as shown in Figure 2 – 4, is a typical saw-tooth shape. It can be considered as a piecewise continuous linear system, which consists of two repetitive segments, known as segment 1 and segment 2. The saw-tooth shape suggests two coins, or two scenarios, are required in the transformation of ratchet potential in terms of coin-tossing games, namely game B. Segment 1 of saw-tooth shape, as depicted in Figure 2 – 4, suggests one coin with strong losing probability. On the other hand, segment 2 of saw-tooth shape demands another coin with relatively weak winning probability. Suppose probabilities P_2 and P_3 are used to represent the winning probability of coin 2 and coin 3, respectively. The losing probability of coin 2 is therefore much greater than the winning probability of coin 3, *i.e.*, $1 - P_2 > P_3$.

In Brownian flashing ratchet, particles choose which segment to follow depends on their corresponding spatial locations [15]. In game B of original Parrondo's paradox, deciding which coin to use, coin 2 or coin 3, requires modular arithmetic. The modular arithmetic has one limitation: at any number of games played, the capital of gambler should always be an integer number. Therefore, the games in original Parrondo's paradox are traditional gambling games, in which each toss of coin produces two possible outcomes, either a win (results in a gain of one unit of capital) or a loss (results in a loss of one unit of capital). By following such assumption, the modular arithmetic can be formed over a range of capital. It is controlled by a predefined integer number, denoted by M . If the capital of gambler is divisible by M , scenario 1 is followed, that is,

coin 2 is played; otherwise, scenario 2 is followed, that is, coin 3 is played. This is the reason explains why original Parrondo's paradox is also called capital-dependent Parrondo's paradox.

In addition to that, the original Parrondo's paradox takes one additional assumption that the game is not stopped even if any instantaneous capital of the player reaches zero or even a negative value until a specified number of games have been played. This assumption plays an important role in the simulation of both versions of Parrondo's paradox. Even in reality, such assumption can be simply offset by providing the player sufficient amount of starting capital prior to the beginning of the game.

All these game rules as specified above establish the foundation of Parrondo's paradox. It can be simply represented by using two individual tree diagrams as depicted in Figure 2 – 5. Game A is a zero-order memoryless game, whereas only a single coin, coin 1 of winning probability P_1 , is played. Game B is a condition-based game, deciding which coin, coin 2 or coin 3, is played depending on the amount of capital a gambler owns prior to each game – if the capital is divisible by M , a predefined integer number, coin 2 is played; otherwise, coin 3 is played.

Figure 2 – 5: *Foundation of capital-dependent Parrondo's paradox*

In addition to the foundation as specified above, it is worth to mention that Parrondo's games cannot be classified in terms of traditional game theory, which generally involves the strategic decision making on behalf of the individual players in the situation. In capital-dependent

Parrondo's games, the players (or the gamblers of the game) are considered to play against a common opponent, for instance, the casino, and the adopted strategy is stated by rule of the game.

In summary, the framework of capital-dependent Parrondo's paradox has been concluded in the section. The subsequent task is to determine the range of working probabilities of three biased coins, that is, probabilities P_1 , P_2 , and P_3 , which is capable of generating such paradoxical effect.

2.2.2 Working probabilities of original Parrondo's paradox

As specified by game rule of capital-dependent Parrondo's paradox, both game A and game B are losing games if played isolatedly. However, if two games are played in any periodically alternating order, a winning expectation is produced counterintuitively [5].

For game A, in order to make it a losing game if played isolatedly, the winning probability of biased coin 1 must be smaller than the corresponding losing probability, *i.e.*, $P_1 < 1 - P_1$, where $P_1, 1 - P_1 \in [0, 1]$, it can be rearranged in equation 2.1.

$$\frac{1 - P_1}{P_1} > 1 \quad (2.1)$$

Similar to game A, game B is also a losing game in the perspective of player if played isolatedly. In original capital-dependent Parrondo's paradox [4 – 6], the value of predefined integer equals to 3, *i.e.*, $M = 3$. Therefore, the result of capital $C \bmod M$ is likely to be 0, 1, or 2. As specified in Figure 2 – 5, if the instantaneous capital is divisible by M, *i.e.*, $C(t) \bmod M = 0$, coin 2 is played, which is of winning probability P_2 and losing probability $1 - P_2$; if any instantaneous capital during the game is not divisible by M. *i.e.*, $C(t) \bmod M \neq 0$ (in this case, $C \bmod M = 1$ or 2), coin 3 is used, which is associated with winning probability P_3 and losing probability $1 - P_3$. In order to make game B also a losing game, the winning probability of game B must be less than its corresponding losing probability, *i.e.*, $(1 - P_2)(1 - P_3)(1 - P_3) > P_2P_3P_3$. It can be rearranged, resulting in equation 2.2.

$$\frac{(1 - P_2)(1 - P_3)^2}{P_2P_3^2} > 1 \quad (2.2)$$

If game A and game B are played by switching these two games randomly, it produces a winning expectation. In order to randomly mix game A and game B in any alternating order, one additional parameter, called mixing parameter, is introduced. The mixing parameter, denoted by γ , is defined as the probability of playing game A and it is in the range between 0 and 1, *i.e.*, $\gamma \in [0, 1]$. If the mixing parameter $\gamma = 0$, only game B is played; on contrary, if the mixing parameter $\gamma = 1$, only game A is played. In the original Parrondo's paradox, both game A and game B have equal chance to be selected, therefore, the mixing parameter $\gamma = 0.5$.

The randomized game¹, game C, due to the linear combination with game B, is also a condition-based game depending on whether any instantaneous capital is divisible by M. If the capital is divisible by M, the winning probability of randomized game, game C, is:

$$q_1 = \gamma P_1 + (1 - \gamma)P_2 \quad (2.3 \text{ a})$$

If the capital is not divisible by M, the winning probability of randomized game is:

$$q_2 = \gamma P_1 + (1 - \gamma)P_3 \quad (2.3 \text{ b})$$

As specified by rule of capital-dependent Parrondo's paradox, the randomized game, game C, is a winning game. In addition to that, both game B and game C are condition-based games. Therefore, the winning probability of game C must be greater than losing probability of game C, which can be mathematically described by using equation 2.4.

$$\frac{(1 - q_1)(1 - q_2)^2}{q_1 q_2^2} < 1 \quad (2.4)$$

In summary, in order to generate the paradoxical effect, which is delivered by capital-dependent Parrondo's paradox, the probability of three coins have to satisfy all three equations as stated above – equation 2.1 (game A is a losing game), equation 2.2 (game B is a losing game), and equation 2.4 (randomized game C is a winning game).

In the original Parrondo's paradox [4 – 6], instead of controlling three individual probabilities P_1 , P_2 , and P_3 separately, one additional biasing parameter, denoted by ε , is introduced to control all three probabilities at once. Without the involvement of biasing

¹ **Randomized game:** A mixed game of switching game A and game B, which is controlled by mixing parameter γ .

parameter ε , *i.e.*, $\varepsilon = 0$, both game A and game B are fair games² when played isolatedly. If the biasing parameter $\varepsilon > 0$, both game A and game B are losing games once they are played isolatedly. On contrary, if the biasing parameter $\varepsilon < 0$, both game A and game B are winning games once they are played isolatedly.

The selected probabilities of games in original capital-dependent Parrondo's paradox [5] are specified in equation 2.5 (a) – (c).

$$P_1 = \frac{1}{2} - \varepsilon \quad (2.5 \text{ a})$$

$$P_2 = \frac{1}{10} - \varepsilon \quad (2.5 \text{ b})$$

$$P_3 = \frac{3}{4} - \varepsilon \quad (2.5 \text{ c})$$

From now on, the problem of determining the working probabilities of original Parrondo's paradox becomes finding the actual range of biasing parameter ε . By substituting equation 2.5 (a) to 2.5 (c) into equation 2.1, 2.2, and 2.4, the resulting equations are equation 2.6 (a) – 2.6 (c) as indicated below.

$$\varepsilon > 0 \quad (2.6 \text{ a})$$

$$\varepsilon(80\varepsilon^2 - 8\varepsilon + 49) > 0 \quad (2.6 \text{ b})$$

$$320\varepsilon^3 - 16\varepsilon^2 + 229\varepsilon - 3 < 0 \quad (2.6 \text{ c})$$

It is no doubt to have $\varepsilon > 0$, as specified in equation 2.6 (a), as both game A and game B must be losing games as required by capital-dependent Parrondo's paradox. For equation 2.6 (b), $\varepsilon > 0$, the quadratic part of $80\varepsilon^2 - 8\varepsilon + 49$ is less than zero (*i.e.* $\Delta = b^2 - 4ac = (-8)^2 - 4 \times 80 \times 49 = -15616 < 0$), therefore, all existing roots are imaginary, which simply states, for any real number of ε , the equation is always satisfied, *i.e.*, for any value of ε , equation $80\varepsilon^2 - 8\varepsilon + 49 > 0$.

The equation 2.6 (c) can be solved numerically, which results in one real number and two imaginary numbers. Only the real number is retained, which approximately equals to 0.0131093. By combining the results obtained from equation 2.6 (a) to 2.6 (c), the working range of biasing parameter ε can be specified by using one single equation, that is, equation 2.7.

² **Fair game:** The definition of fair is a little looser in capital-dependent Parrondo's paradox. The fair game in Parrondo's paradox is considered to be a game has probability of moving up n states is equal to the probability of moving down n state when n becomes large.

$$0 < \varepsilon < 0.0131093 \quad (2.7)$$

As long as the selected biasing parameter ε is within the range, as specified in equation 2.7, the Parrondian effect is produced.

2.2.3 Simulation of original Parrondo's paradox

D. Abbott was delighted by J. M. R. Parrondo's invention. He collaborated with his colleague G. P. Harmer in the simulation of capital-dependent Parrondo's paradox, which is formally reported in [5 – 6].

The simulation follows the probabilities specified by equation 2.5 (a) – 2.5 (c). To ensure both game A and game B are losing games if played isolatedly, and the randomized game, game C, is capable of producing a winning expectation, the biasing parameter ε should be in the range as specified in equation 2.7. In original version of Parrondo's paradox, the selected biasing parameter is 0.005, *i.e.*, $\varepsilon = 0.005$. In order to ease the simulation, it takes the assumption that the game is not interrupted even if the capital reaches zero or even a negative number until specified number of games played and the gambling game starts with zero capital. In addition to that, a total 10,000 trails is simulated for each game, the resulting capital is averaged over 10,000 number of trails, and a totally 100 games is played. The simulation is regenerated by using MATLAB. The simulated result is presented in Figure 2 – 6.

There are totally two information has been revealed by observing Figure 2 – 6. The first one is that two curves that representing game A (in pink) and game B (in blue) have the downward trend with the advance of the number of games played, which confirms both game A and game B are losing games if they are played isolatedly by the gambler. However, game A and game B are slightly different – game B has a transient effect for small number of games played. Such transient effect is a direct consequence of capital dependency of game B. Such effect vanishes after approximately 20 – 30 numbers of games played. The second one is that once these two games are played in a mixed manner, in which both game A and game B have equal chance to be played, *i.e.*, $\gamma = 0.5$, the resultant, the mixed game (or randomized game C) is counter-intuitively a winning game as mixed game (in black) have a clearly upward trend with the advance of the

number of games played. Such generated paradoxical effect is always referring to as the Parrondian effect.

Figure 2 – 6: *Capital-dependent Parrondo's paradox*

In addition to that, in order to examine whether the different way of periodically mixing are likely to affect the result obtained in the simulation, game A and game B are likely to be played in a deterministic way. The periodically mixing means that game A and game B are played with a specified mixing. In the following simulation, game A and game B are mixed in terms of $\{AABBAABBAABB\dots\}$ and $\{AAAABBBBBBAAAA\dots\}$, which can be expressed as $[a,b] = [2,2]$ and $[a,b] = [4,4]$, respectively. For comparison purpose, the randomly mixing is also retained within the simulation. Furthermore, in order to observe the long-term effects, a total 200 number of games are played as shown in Figure 2 – 7.

Figure 2 – 7: *Deterministically mixed games in capital-dependent Parrondo's paradox*

Deterministically mixing game A and game B is also capable of producing the analogous paradoxical effects. Quickly toggling between game A and game B, that is, game A and game B are mixed in terms of {AABBAABBAABB...}, is definitely outperformed randomly mixed game. Slowing down the toggling speed leads to the drop of capital during different number of games played. However, the upward trend still indicated the similar paradoxical effect is able to be generated. Therefore, it is possible to conduct the generated paradoxical effect is independent of the mixing sequences between game A and game B.

2.3 Dynamics of Capital-dependent Parrondo's Paradox

The similarity and difference of two mechanisms, Brownian flashing ratchet and Parrondo's paradox, are summarized in Table 2 – 1.

Quantity	Flashing Brownian Ratchet	Original Parrondo's Paradox
Source of potential	Electrostatic, Gravity	Rule of the games
Switching	Electromotive force, ON and OFF	Games A and B played
Duration	Time	Number of games played
Biasing parameter	Macroscopic field gradient	Biasing parameter ε
Transport quantity	Brownian particles	Capital of gambler
Measurable output	Moved distance	Amount of capital
External energy	ON and OFF triggering energy	None
Potential shape	Depends on α	Depends on probabilities
Mode of analysis	Fokker-Planck equation	Discrete-time Markov chain

Table 2 – 1: *Relationship between various quantities used in capital-dependent Parrondo's paradox and Brownian flashing ratchet*

The most significant difference between these two mechanisms, the Brownian flashing ratchet and Parrondo's paradox, is lying on their corresponding system dynamics. The Brownian flashing ratchet is continuous in time and space, as the particles are ubiquitous along the spatial location. However, the Parrondo's paradox is a typical discrete-time Markov chain, as the capital of the player at any state has to be an integer number as governed by modular arithmetic of capital-dependent Parrondo's paradox. This section begins with a brief introduction of discrete-time Markov chain (subsection 2.3.1) by selectively including the essential definitions and basic theories, which is required for the subsequent analysis of capital-dependent Parrondo's paradox in subsection 2.3.2.

2.3.1 Discrete-time Markov chain

The study of probability theory has a prolonged history. The boundary, which separates the probability theory into classic and modern probability theory, is on the state-dependency for each individual trials [16]. The classical probability theory is dealing with the independent trail process whereas the knowledge of previous trails does not affect its subsequent trials. It is obvious to

figure out the limitations of classic probability theory as it failed to describe the majority of real-life situations.

Due to this concern, A. A. Markov established the state-dependent probability theory, whereas the outcome of subsequent experiment is depending on the result from previous trial. Such state-dependent process is now regarded as the Markov process. In addition to that, Markov process can be further divided into discrete-time Markov process and continuous-time Markov process [17]. In this section, it is mainly dealing with one specific type of discrete-time Markov process, called Markov chain [18].

Markov chain is a discrete-time random process with the Markov properties, in which the subsequent random variable, let's say, X_{n+1} depends on previous random variable X_n rather than even earlier values $X_{n-1}, X_{n-2}, \dots, X_0$ of any random sequence. Such state dependency can be mathematically described by simply using one parameter called transition probability, denoted by P_{ij} . The entire state dependency can be described by using equation 2.8.

$$\begin{aligned} P[X_{n+1} = j \mid X_n = i, X_{n-1}, X_{n-2}, \dots, X_0 = i_0] \\ = P[X_{n+1} = j \mid X_n = i] \\ = P_{ij} \end{aligned} \quad (2.8)$$

where i and j are the corresponding state numbers.

The equation 2.8 has two layers of physical meanings. First of all, the value of random variable X_n contains all the past history that is required to predict the subsequent random variable X_{n+1} . Secondly, suppose at current state, the Markov chain is located in state i , it moves to next state j with a fixed transition probability P_{ij} . The property of transition probability P_{ij} can be mathematically captured by equation 2.9.

$$0 \leq P_{ij} \leq 1 \text{ and } \sum_{j=0}^{\infty} P_{ij} = 1 \quad (2.9)$$

Transition diagram

The discrete-time Markov chain can be graphically represented by a diagram called transition diagram. A transition diagram consists of two elements – nodes and directed arcs. The nodes represent the sample space of random variable X_n , where $n \in \mathbb{Z}_+$ and directed arcs are used to describe the transition process from state i to its subsequent state j .

A typical example, as described in Figure 2 – 8, is used to demonstrate the expression of a typical two-state discrete-time Markov chain in terms of a transition diagram. Suppose a system consists of two states, let's say, state 0 and state 1. The transition probability from state 0 to state 1 is a , the transition probability from state 1 to state 0 is b . Additionally, the transition probability for state 0 to remain in the same state is $1 - a$ and the transition probability for state 1 to remain in the same state is $1 - b$.

Figure 2 – 8: *Example of transition diagram*

Transition probability matrix

In addition to the transition diagram, the discrete-time Markov chain is can also be represented in terms of a transition probability matrix (or simply, transition matrix), denoted by bold letter P . The rows of transition matrix stand for the states at current step (step n), the columns of the transition matrix stand for the states at next step, which is step $n + 1$. The elements, which made up of transition matrix, are the transition probabilities P_{ij} from state i to state j in the next step. The transition matrix has its unique property: it is always a $N \times N$ square matrix, in which N is the total number of states.

The same example, as indicated in Figure 2 – 8, is used to demonstrate the properties of transition probability matrix P , as shown in Figure 2 – 9.

Subsequent state, $n + 1$

Current state, n

$$\mathbf{P} = \begin{matrix} \begin{matrix} 0 \\ 1 \end{matrix} & \begin{bmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{bmatrix} \end{matrix} = \begin{bmatrix} 1 - a & a \\ b & 1 - b \end{bmatrix}$$

Figure 2 – 9: *Example of transition probability matrix*

Chapman-Kolmogorov equation and n -step transition probability matrix

One of the reasons to express the discrete-time Markov chain in terms of the transition matrix is to study the behavior of the Markov process after n steps. The n step transition probabilities are governed by Chapman-Kolmogorov equation (equation 2.10).

$$P_{ij}(m + n) = \sum_{k=0}^k p_{ik}(m) p_{kj}(n) \quad (2.10)$$

The Chapman-Kolmogorov equation can be extended to the multiplication of transition matrix $P(m + n) = P(m) \times P(n)$. Such that, for a finite discrete-time Markov chain with transition matrix P , the n -step transition matrix $P(n)$ is:

$$\mathbf{P}(n) = \mathbf{P}^n \quad (2.11)$$

State distribution vector

The state distribution vector, as denoted by bold letter $\mathbf{p}$, is used to record the probabilities of the process in different steps. The vector $\mathbf{p} = [p_0, \dots, p_k]'$ is a state distribution vector if $\sum_{j=0}^k p_j = 1$, where each probability $p_j \in [0, 1]$. At initial step, $n = 0$, the corresponding state distribution vector is termed as initial state distribution vector, denoted by $\mathbf{p}(0)$. The state distribution vector $\mathbf{p}_j(n)$ at n step can be calculated by one of two possible methods:

- One iteration with n step transition probabilities:

$$\mathbf{p}'(n) = \mathbf{p}'(0)\mathbf{P}^n \quad (2.12 \text{ a})$$

- n iteration with one step transition probabilities:

$$\mathbf{p}'(n) = \mathbf{p}'(n-1)\mathbf{P} \quad (2.12 \text{ b})$$

Stationary/steady distribution vector

In addition to the state distribution vector, another important analysis of Markov chain can be achieved by studying the long-term behavior of state distribution vector, $\mathbf{p}(n)$, where n approaches infinite, *i.e.*, $n \rightarrow \infty$. The resultant vector, namely, stationary distribution vector $\boldsymbol{\pi}$ is defined by using equation 2.13.

$$\boldsymbol{\pi} = \lim_{n \rightarrow \infty} \mathbf{p}(n) \quad (2.13)$$

The stationary distribution of a discrete-time Markov chain occurs when the distribution reaches a steady state, *i.e.*, the distribution among states is no longer changed from one to another. The property of stationary distribution vector can be described by using equation 2.14.

$$\boldsymbol{\pi}'(n+1) = \boldsymbol{\pi}'(n)\mathbf{P} \quad (2.14)$$

There are several existing methods that can be used to solve the stationary distribution vector $\boldsymbol{\pi}$ if the transition probability matrix $\mathbf{P}$ is known. The simplest approach was proposed by P. Pyke, which can be described by using equation 2.15.

$$\pi = \frac{1}{D} \text{diag}[\text{cofac}(\mathbf{I} - \mathbf{P})] \quad (2.15)$$

where D is the normalization constant which adds up the diagonal elements of the cofactor of $\mathbf{I} - \mathbf{P}$. The function of diag returns with the main diagonal of the matrix. And the function of cofac results in the cofactors of the matrix.

2.3.2 Analysis of capital-dependent Parrondo's paradox

As mentioned above, the dynamics of capital-dependent Parrondo's paradox can be mathematically analyzed by using discrete-time Markov chain [18]. As specified by rule of the game, every single value of player's capital is analogous to individual states of discrete-time Markov chain. And the probabilities of these three coins involved in capital-dependent Parrondo's paradox are similar to the transition probabilities among different states of discrete-time Markov chain.

Game A

Game A of capital-dependent Parrondo's paradox can be simply represented by transition diagram as shown in Figure 2 – 10.

Figure 2 – 10: Transition diagram of Parrondo's game A

The initial capital can be started at any arbitrary n state of game A, where n can be any integer number. The capital moving from state n to state $n + 1$, as indicated in Figure 2 – 10, with transition probability P_1 , i.e., the player has probability of P_1 to win if game A is played isolatedly,

and as a result of winning, one unit of capital is awarded to the player. On the opposite side, the capital moving from state n to state $n - 1$ with transition probability of $1 - P_1$, *i.e.*, the player has probability of $1 - P_1$ to lose if game A is played isolatedly, and as a consequence of losing, one unit of capital is lost. In addition to that, it can be observed from Figure 2 – 10, the states can be extended to double infinite: negative infinite and positive infinite.

As mentioned in subsection 2.3.1, a typical discrete-time Markov chain can be mathematically expressed in terms of a transition probability matrix. In this case, game A of capital-dependent Parrondo's paradox can also be expressed in terms of a transition probability matrix, which contains values for the two diagonals that are offset from the main diagonal. The remaining elements of the transition probability matrix are all zeroes. The transition probability matrix of game A can be represented by equation 2.16.

$$P_A = \begin{bmatrix} 0 & P_1 & & & & \\ 1 - P_1 & 0 & P_1 & & & \\ & 1 - P_1 & 0 & P_1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 - P_1 & 0 & P_1 \\ & & & & 1 - P_1 & 0 \end{bmatrix} \quad (2.16)$$

Game B

Similarly, game B of capital-dependent Parrondo's paradox can also be represented by transition diagram of discrete-time Markov chain as shown in Figure 2 – 11.

Figure 2 – 11: Transition diagram of Parrondo's game B

The discrete-time Markov chain, which represents Parrondo's game B, contains n periodic subsystems of length of predefined integer M . The value of M is the same as that used in original Parrondo's paradox, where $M = 3$. The periodic subsystems are caused by state dependency (or

capital dependency) of Parrondo's game B. The probabilities leaving the states, which are the multiple of M , are either P_2 (winning probability of coin 2) or $1 - P_2$ (losing probability of coin 2). On the other hand, the probabilities leaving the states, which are not the multiple of M , are either P_3 (winning probability of coin 3) or $1 - P_3$ (losing probability of coin 3).

Similar to game A, game B of capital-dependent Parrondo's paradox can also be described in terms of a transition probability matrix. The distribution of elements within transition probability matrix of game B is identical to that of game A, which has only non-zero values for two diagonals that are offset from main diagonal, as specified in equation 2.17.

$$\mathbf{P}_B = \begin{bmatrix} 0 & P_3 & & & & \\ 1 - P_3 & 0 & P_3 & & & \\ & 1 - P_2 & 0 & P_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 - P_3 & 0 & P_3 \\ & & & & 1 - P_3 & 0 \end{bmatrix} \quad (2.17)$$

The state dependency of Parrondo's game B can also be observed from its transition probability matrix, as shown in equation 2.17. The winning probabilities and losing probabilities in every M th column are P_2 and $1 - P_2$, respectively.

The study of dynamics of capital-dependent Parrondo's paradox plays a vital role in the analysis. The dynamics of Parrondo's game A is really straightforward. Due to the state dependency, the dynamics of Parrondo's game B can be extremely complicated. However, by observing the long-term behavior of Parrondo's game B, it is possible to figure out game B consists of many periodic subsystems in the length of predefined integer M . In reality, the behavior of each subsystem is totally identical. Therefore, the actual study of dynamics of Parrondo's game B can be simplified.

Instead of extending the analysis to double infinite, it is able to extract the subsystem from the original discrete-time Markov chain, which is still capable of representing the original system. The extracted subsystem can be simply described by equation 2.18.

$$Y_n \equiv X_n \bmod M \quad (2.18)$$

The subsystem of game B can be modeled by using a cyclic discrete-time Markov chain of M number of states and it is cyclic, which means once the state reaches the highest state, which

is state $M - 1$, it is likely to return to its initial state, that is, state 0. Such modified subsystem can be visualized in Figure 2 – 12.

By employing such method, the double infinitive discrete-time Markov chain of game B can be modified into an equivalent but much simplified finite discrete-time Markov chain. The original transition probability matrix now changes to a finite $M \times M$ matrix.

Figure 2 – 12: *Transition diagram of Parrondo's module game B*

Transient effect and steady state effect

The dynamics of capital-dependent Parrondo's paradox can be studied in terms of transient effect (or initial effect) and steady state effect (or long term effect).

The transient effect of capital-dependent captures the movements of capital with respect to a small number of games played. It is possible to study the transient effect of game A and game B by simply observing Figure 2 – 6. For game A, the transient effect is almost negligible as there is very limited fluctuation of the curve (in pink) which represents the relationship between instantaneous capital and number of games played.

For game B, however, the transient effect is significant. As shown in Figure 2 – 6, the curve (in blue) which indicates the relationship between capital and the number of games played has very

large fluctuations at the beginning 20 to 30 number of games. Such phenomenon is caused by the nature of capital-dependency of Parrondo's game B.

As mentioned in subsection 2.3.1, the long-term effect of Parrondo's paradox is determined by obtaining the corresponding stationary distribution vector π . Instead of using the double infinite transition probability matrix as specified in equation 2.17, it is possible to modified it into an finite transition probability matrix as shown in equation 2.18 by using the modulo method.

$$\mathbf{P}_B = \begin{bmatrix} 0 & P_2 & 1 - P_2 \\ 1 - P_3 & 0 & P_3 \\ P_3 & 1 - P_3 & 0 \end{bmatrix} \quad (2.19)$$

The cofactor of the matrix $\mathbf{I} - \mathbf{P}$ can be expressed as follows:

$$\text{cofac}(\mathbf{I} - \mathbf{P}_B) = \begin{bmatrix} 1 + P_3(P_3 - 1) & P_3 - 1 & (P_3 - 1)^2 + P_3 \\ -P_2P_3 + P_3 - 1 & 1 + P_3(P_2 - 1) & (P_3 - 1) - P_2P_3 \\ P_2P_3 - (P_2 - 1) & -P_2P_3 + P_2 - 1 & 1 + P_2(P_3 - 1) \end{bmatrix} \quad (2.20)$$

Therefore, the stationary distribution vector of game B can be depicted by equation 2.21.

$$\pi^B = \frac{1}{D} \begin{bmatrix} 1 - P_3 + P_3^2 \\ 1 - P_3 + P_2P_3 \\ 1 - P_2 + P_2P_3 \end{bmatrix} \quad (2.21)$$

where the normalization constant, $D = 3 - P_2 - 2P_3 + 2P_2P_3 + P_3^2$.

Similarly, the stationary vector of game A can be simply determined by setting $P_2 = P_3 = P_1$. Such that, the equation 2.21 can be converted to equation 2.22.

$$\pi^A = \frac{1}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (2.22)$$

2.4 Other Paradoxical Phenomenon

As inspired by the framework of Parrondo's paradox, that is, two losing strategies are possible to be combined into the production of winning expectation, other paradoxical phenomenon has been subsequently proposed by various researchers [8 – 12, 19 – 26]. In this section, the generated paradoxical phenomenon is selectively reviewed.

2.4.1 History-dependent Parrondo's paradox [8]

The original Parrondo's paradox is capital-dependent as the game of rules relies on modulo arithmetic, which is merely depending on the instantaneous capital of the gambler. This specific property, in certain perspective, significantly limits the possible applications of the concept in the real life situations. It may be desirable to create other variations of Parrondo's paradox. One of the variations was later developed by J. M. R. Parrondo. In this variation of game B of Parrondo's paradox, choosing which scenario to follow is merely depending on the outcomes from previous two games. Therefore, this variation of Parrondo's paradox is regarded as the history-dependent Parrondo's paradox.

In history-dependent Parrondo's paradox, game A remains unchanged as that of capital-dependent Parrondo's paradox. Game B is modified depending on the results from previous two games, which has four possible combinations, LL, LW, WL, WW (W: win, L: lose). Therefore, game B has four different scenarios and therefore four respectful coins, namely coin 2, coin 3, coin 4, and coin 5. The actual rule of game B of history-dependent Parrondo's paradox can be summarized in Table 2 – 2.

Before last ($t - 2$)	Last ($t - 1$)	Current (t) Coin No.	Probability of winning P_{win}	Probability of losing P_{loss}
Loss	Loss	2	P_2	$1 - P_2$
Lose	Win	3	P_3	$1 - P_3$
Win	Loss	4	P_4	$1 - P_4$
Win	Win	5	P_5	$1 - P_5$

Table 2 – 2: Game rules of game B of History-dependent Parrondo's paradox

Similarly, the history-dependent Parrondo's paradox can be represented by using a simple tree diagram as shown in Figure 2 – 13. Game A remains unchanged to that of the capital-dependent Parrondo's paradox. Game B is remodeled to a game which contains four possible scenarios.

Figure 2 – 13: *Foundation of history-dependent Parrondo's paradox*

The winning probabilities of four coins of history-dependent Parrondo's paradox are determined by using the same methodology as specified in subsection 2.2.2. The selected probabilities are specified by equation 2.23 (a) to 2.23 (e).

$$P_1 = \frac{1}{2} - \varepsilon \quad (2.23 \text{ a})$$

$$P_2 = \frac{9}{10} - \varepsilon \quad (2.23 \text{ b})$$

$$P_3 = \frac{1}{4} - \varepsilon \quad (2.23 \text{ c})$$

$$P_4 = \frac{1}{4} - \varepsilon \quad (2.23 \text{ d})$$

$$P_5 = \frac{7}{10} - \varepsilon \quad (2.23 \text{ e})$$

Similarly, the biasing parameter ε is introduced to control all the probabilities of game B. Once biasing parameter $\varepsilon = 0$, both game A and game B are fair games if played isolatedly. If $\varepsilon > 0$, both game A and game B are losing games, and if $\varepsilon < 0$, both game A and game B are winning games.

The simulations are created by using the probabilities as stated in equation 2.23 (a) to 2.23 (e). The same value of biasing parameter $\varepsilon = 0.005$ is used in the simulation. The simulation can be reproduced by using MATLAB. The result is presented in Figure 2 – 14.

Figure 2 – 14: *History-dependent Parrondo's paradox*

The analysis of dynamics of history dependent Parrondo's paradox is much similar to that of capital-dependent Parrondo's paradox. Game B of history-dependent Parrondo's paradox can be expressed in terms of transition probability diagram, as shown in Figure 2 – 15.

Figure 2 – 15: *Transition diagram of history-dependent Parrondo's paradox*

Similarly, the transition probability diagram, as indicated in Figure 2 – 15, is not a standard discrete-time Markov chain, as the subsequent state is not merely depending on its previous state. In addition to that, such Markov chain is also leading to double infinite, which makes the entire analysis extremely complicated. The identical method, modulo arithmetic, that is used in the analysis of game B of capital-dependent Parrondo's paradox is used. The simplified transition probability diagram is specified in Figure 2 – 16.

Figure 2 – 16: *Simplified transition diagram of game B of history-dependent Parrondo's paradox*

Analogously, the simplified transition diagram of game B can be represented in terms of transition probability matrix, as indicated in equation 2.24.

$$\mathbf{P}_B = \begin{bmatrix} 1-P_2 & P_2 & 0 & 0 \\ 0 & 0 & 1-P_3 & P_3 \\ 1-P_4 & P_4 & 0 & 0 \\ 0 & 0 & 1-P_5 & 1 \end{bmatrix} \quad (2.24)$$

Once the transition probability matrix is finalized, the remaining characteristics can be therefore determined.

2.4.2 Cooperative Parrondo's games [9]

As inspired by Parrondo's history-dependent game, Toral developed a new family of games based on Parrondo's strategy, named as cooperative Parrondo's game. The cooperative Parrondo's game consists of N number of players arranged in a circle such that each player has two adjacent neighbors. A player randomly chooses among N players. Once a player wins in any particular game, it is labeled as "winner", otherwise, it is labeled as "loser". Game A remains identical to that of Parrondo's game. The winning probability of game B is depending on the state of the neighbors to the left and right of the player, which falls in one of the possible states known as {LL, LW, WL, WW}, each state consists of state of the left neighbor and right neighbor. The result of simulation demonstrates that this developed game also generate similar phenomenon from history-dependent Parrondo's paradox.

2.4.3 Buy-low-sell-random trading approach [12]

Maslov, S et al. proposed a toy model in studying the portfolio of a two theoretical stock. One stock is termed as low-risk stock, the daily fluctuations is relatively small. Another stock is named as high-risk stock, whereas the daily fluctuation is extremely large. It is proved that, if maintaining a 50:50 portfolio through portfolio rebalancing (*i.e.* takes the total cash, let's say, C and buy $C/2$ of high-risk stock and $C/2$ of low-risk stock), the expected return on investment is much higher as compared to the strategy of investing in either one of the two stocks as shown in Figure 2 – 17.

Figure 2 – 17: *Volatility pumping model*

2.4.4 Allison mixture [19]

Allison, A. et al proposed a new form of Parrondo's paradox, named as the Allison mixture. The idea is that, there are two sequences, namely, sequence 1 and sequence 2, which are totally random strings of 0s and 1s, and they are independent and have zero autocorrelation. However, if mix these two sequences in terms of a third sequence called sequence 3, it produces a counterintuitive result that the resulting sequence is not completely random.

2.5 Issues Surrounding Parrondo's Paradox

In order to have an unbiased understanding about the concept of Parrondo's paradox, it is necessary to hear the voices from both sides – supporters as well as critics. There are several controversial issues surrounding the Parrondo's paradox as specified by various researchers. These issues can be broadly summarized as follows.

1. Game B appears to be a winning game if played individually [28]

John L. Casti interpreted that game B is a winning game, which invalidates the requirement of Parrondo's paradox. From the comments of the original article, when the pre-defined integer $M = 3$, the chance of $X_n \bmod 3$, which is 0, 1, or 2, occurs with the equal possibility, $1/3$. Therefore, the chance of using a bad coin (coin 2) and a good coin (coin 3) is one to two. Therefore, with the biasing parameter $\varepsilon = 0$, the winning probability of game B is:

$$p_{B,win} = (2/3 \times 3/4) + (1/3 \times 1/10) = 16/30 \quad (2.25)$$

The result implies that game B is a winning game instead of a fair game. Such result is proved to be a mistake. This is mainly because of the author choosing the wrong analyzing method by considering the game statistically. The correct analysis is by using discrete-time Markov chains. The stationary distribution vector of game B has already created in equation 2.21. By substituting the winning probabilities of coin 2 and coin 3 into equation 2.21, the result reveals that the probability in each state is $5/13$, $2/13$, and $6/13$, correspondingly. Such that, with zero biasing parameter, the winning probability of game B is:

$$p_{B,win} = (5/13 \times 1/10) + (2/13 \times 3/4) + (6/13 \times 3/4) = 1/2 \quad (2.26)$$

As shown above, the winning probability of game B is $1/2$. Therefore, the game B is proved to be a fair game in the long run, when the biasing parameter $\varepsilon = 0$.

2. Parrondo's game should not be called game at all. And the definition of fairness in Parrondo's game is incorrect [29]

A. Costa et al. revealed two issues associated with Parrondo's paradox. One is whether the Parrondo's paradox using the same definition of game as that is used in game theory. And the second issue is concerning whether the definition of fairness in Parrondo's paradox is correct.

The widely accepted version of game theory's definition was proposed by Myerson. From the opinion, game theory attempts to mathematically capture the behavior in strategic situations, or games, in which an individual's success in making choices depends on the choices of the others. In simple words, the strategic decisions on behalf of the players in the game are compulsory elements in constructing a game. However, in Parrondo's paradox, the freedom of selecting which strategy (or strategies) to employ is eliminated in both versions of Parrondo's paradox. In other words, choosing which coin to play is not dependent on player's choice, it is governed by the rules of the game. Such that, the meaning of the word "game" in Parrondo's game is different from the traditional definition of the term "game" in game theory.

The author also mentioned the classical definition of fairness which was originally proposed by J. L. Doob [81]. Consider a gambler repeatedly plays a game and the gambler's capital after the n^{th} game is X_n , where $n \in \mathbb{Z}_+$, a fair game should satisfy the following condition:

$$E[X_{n+1} | X_0, X_1, \dots, X_n] = X_n \quad (2.27)$$

In other words, a game is fair if the stochastic process arising from its repeated play is a martingale, that is, the expected value after playing a game is the same as the present one. However, by using the definition of J. L. Doob, there is no doubt for game A to be a fair game. However, for game B, the probabilities of winning are $p_2 < \frac{1}{2}$ and $p_3 > \frac{1}{2}$. By using such definition, game B does not actually respond to a fair game, as when X_0 is a multiple of M , $E(X_1 | X_0) < X_0$ and when X_0 is not a multiple of M , $E(X_1 | X_0) > X_0$.

There is one thing for sure that the term "game" used in Parrondo's paradox is not identical to the definition of the "game" used in classical game theory. However, to the issue concerning the fairness of the game, as responded by Abbot [82], although game B does not satisfy the narrowest martingale definition of fairness, it is not very important in the context of the paradoxical nature of the games as game B is definitely lose when the biasing parameter $\epsilon > 0$.

In addition to the issues, the author uses a probability space to explain the secrets lying behind the Parrondo's paradox. There are one straight line and one curve generated in the probability space. The straight line represents game A by setting $p_2 = p_3 = p_1$. The winning probability of game A, p_1 , can be any points along the straight line. The curve is generated by following the equation 2.28.

$$\frac{p_2 p_3^2}{(1 - p_2)(1 - p_3)^2} = 1 \quad (2.28)$$

The physical meaning of the curve specifies that the winning probability of game B is identical to the losing probability of game B. Such curve equally divided the entire probability space into two separating regions. The region above the curve is named as the winning region (game B is a winning game). The region below the curve is the losing region (game B is a losing game). The simulated probability space is shown in Figure 2 –18.

Figure 2 – 18: *Probability space of capital-dependent Parrondo's paradox*

As specified in Figure 2 – 18, in order to make game A a losing game, the probability P_1 is selected along the straight line inside the losing region. Similarly, to make game B also a losing

game, the probability p_2 and p_3 is any point inside the losing region. If linking these two selected points by using a straight line, there is certain region of the line contained within the winning region. Finally, by varying the value of mixing parameter γ (*i.e.* changing the location of the point along the straight line), any points along this straight line falling into the winning region are the keys to produce a winning expectation of randomized game. Mathematically speaking, the randomized game is formed by a convex linear combination of two games (game A and game B), which is the secret behind the Parrondo's paradox.

3. The Parrondo's paradox is not paradoxical, as the Parrondian effect can be replicated by much simpler games [30 – 31]

T. K. Philips and A. B. Feldman mentioned that the Parrondo's result is not paradoxical. From authors' point of view, the individual games are losing only under a distribution that differs from that which is actually encountered when playing the compound game. In that sense, they proposed a simpler game to illustrate the insights. The proposed game has the following game rules:

- In game A, the player loses \$2 if his capital is an odd number, and loses \$1 if capital is an even number.
- In game B, the player is paid \$6 if the capital is an odd number, and loses \$7 if capital is an even number.

4. There are deterministic ways of playing Parrondo's game, which provide better winning probabilities under more general conditions, than randomization the process.[32]

The opinion proposed by R. Iyengar and R. Kohli can be treated as an extension of the version that specifies Parrondo's game is not a branch of traditional game theory.

From developers' point of view, the player's game is dissimilar to the physical system of Brownian flashing ratchet, the randomized combination of two games is not the most profitable choice for the player. The best strategy for the player to play is to selectively choose the game which offers the highest winning odds.

In addition, it is proved that the randomization cannot produce the winning combination of two individually losing games if game B is a first-order or zero-order Markov chain process.

Chapter 3

Beyond Parrondo's Paradox

Parrondo's paradox is a counterintuitive phenomenon where individually losing games can be combined in producing a winning expectation. The initial purpose of Parrondo's paradox was to mathematically formulate a physical phenomenon generated by flashing Brownian ratchet [3] in terms of two simple gambling games, namely, game A and game B [4]. Unlike conventional game theory, which involves strategic interactions among two or more players, Parrondo's paradox describes a single player, also referred to as the gambler, playing against a common opponent by following a series of predefined game rules. If the gambler plays either game A or game B isolatedly, the longer he plays the more capital he loses as a consequence of playing. However, once the gambler exploits the game rules by alternatively playing exactly the same two games in a periodic order. Counterintuitively, he starts to win the capital by repeating such strategy in a long run. Since that, several studies were made to demonstrate the concept of capital-dependent Parrondo's paradox [5-6], to formulate the expressions of the paradox [7] and to extend the capital-dependent Parrondo's paradox to a history-dependent version [8].

Parrondo's paradox has raised many attentions in media as the selected case is a perfect demonstration of "beating the odds" [4]. Based on the framework of Parrondo's paradox, the same concept has been further extended into other potential applications [9-12]. On the opposite side, the concept has been continuously scrutinized by many researchers since its first appearance [29-32].

In this chapter, it begins with a short summarization of the key concepts of two versions of Parrondo's paradox (section 3.1). It is further ventured into the investigation on whether the analogous paradoxical effect can be reproduced by using a relative simple capital-dependent game as claimed in [30] (section 3.2). In reality, all the parameters that used in Parrondo's paradox can be analyzed inside a probability space, which reveals the secret of the paradox (section 3.3). Based upon this foundation, it is possible to study whether the other combinations of two winning and/or losing games under the framework Parrondo's paradox are capable of generating analogous paradoxical effects (section 3.4). Furthermore, the study of the effects of predefined integer M ,

other alternative mixing sequences, and the non-linear combination of two games, are investigated (section 3.5). In the end, the issues associated with the paradox are discussed and the potential application of Parrondo's paradox is addressed (section 3.6).

3.1 Overview of Parrondo's Paradox

There are totally two versions of Parrondo's paradox, which refers to as capital-dependent and history-dependent Parrondo's paradox. The only difference between these two versions of paradox is lying on the corresponding switching mechanisms of game B.

For both versions, game A is identical, which is a zero-order memoryless gambling game of winning probability of p_1 and losing probability of $1 - p_1$. On the other hand, game B is a typical condition-based game, that is, before any game B is played, the instantaneous capital along the process of game is firstly identified to decide which scenario to be selected and played by the gambler.

For capital-dependent Parrondo's paradox, choosing which scenario to follow merely depends on whether its instantaneous capital along the game, denoted by $C(t)$, is a multiple of a predefined integer M or not. If capital is a multiple of M , *i.e.*, $C(t) \bmod M = 0$, scenario 1 is chosen to be played, in which the corresponding winning probability p_2 is much lower than the losing probability $1 - p_2$. Reversely, if the capital is not a multiple of M , *i.e.*, $C(t) \bmod M \neq 0$, scenario 2 is selected, in which the winning probability p_3 is slightly higher than the corresponding losing probability $1 - p_3$.

Deciding which scenario to be played, for history-dependent Parrondo's paradox, relies on the outcomes of previous two games. As the outcome of each game is either resulting in a win or a loss, there are totally four different combinations of results of previous two games: {lose, lose}, {lose, win}, {win, lose}, and {win, win}. Therefore, there are totally four different scenarios to be selected. Each scenario corresponds to one specific combination of results of previous two games.

In addition to that, all these probabilities are designed to be controlled by using a single parameter, which is known as the biasing parameter, denoted by ε . The central idea is that, by setting the biasing parameter $\varepsilon > 0$, both game A and game B are losing games in the perspective of the gambler, *i.e.*, instantaneous capital of the gambler $C(t)$ is continuously decreasing with the advancement of number t of games played if these two games are played isolatedly. The framework of capital-dependent and history-dependent Parrondo's paradox can be conclusively illustrated in Figure 3 – 1.

Based on the rules of games as specified in Figure 3 – 1 and by setting the value of biasing parameter ε and predefined integer M value to $\varepsilon = 0.005$ and $M = 3$, respectively, the simulation

can be generated by averaging the outcomes of 10,000 trials for each game, and totally 200 games are played, as shown in Figure 3 – 2.

Figure 3 – 2 reveals two essential information: for capital-dependent Parrondo's paradox, both game A (in blue) and game B (in pink) are losing games in the perspective of the gambler if these two games are played isolatedly: the instantaneous capital of gambler $C(t)$ decreases with the advance of number of games played. On the opposite side, once these two games are played in a mixed manner, in which both game A and game B have equal chance to be played, *i.e.*, $\text{Probability (game A)} = \text{Probability (game B)}$, the compound game (in black) is a winning game: the instantaneous capital of gambler $C(t)$ increases with the advance of number of games played.

Capital-dependent Parrondo's paradox **History-dependent Parrondo's paradox** **Figure 3 – 1:** Frameworks of capital-dependent and history-dependent Parrondo's paradox

Figure 3 – 2: Simulation of capital-dependent and history-dependent Parrondo's paradox

3.2 Verification of simple capital-dependent game

From various scrutinizes [29 - 32], there are several issues surround Parrondo's paradox since its first appearance. Some of these issues were responded by its initiators in [82]. The objective of this section is to resolve the remaining issues associated with Parrondo's paradox. The key objective is to testifying whether the identical paradoxical effect can be simply reproduced by replacing original framework of Parrondo's paradox with a relatively simple capital-dependent game as claimed in [30].

Analogously, the proposed framework consists of two isolate games – **game A**: the gambler is likely to lose \$2 if his capital $C(t)$ is an odd number, and lose \$1 if the capital $C(t)$ is an even number; **game B**: the gambler is paid \$6 if his capital $C(t)$ is an odd number, and lose \$7 if the capital $C(t)$ is an even number.

At first glance, the proposed approach seems to be plausible – both game A and game B are losing games in the perspective of the gambler, and the compound game is likely to produce a winning expectation. In order to verify the feasibility of proposed concept, the simulation based on the game rules of this capital-dependent game is firstly produced, as shown in Figure 3 – 3.

Figure 3 – 3: *Simulation of simple capital-dependent game*

As specified in Figure 3 – 3, analogous to original Parrondo’s paradox, game A is a losing game and the compound game is a winning game in the perspective of the gambler. However, game B is failed to be a losing game as with the advancement of number of games played, capital of the gambler increases instead. In reality, the trick employed in game B is relatively simple: the instantaneous capital of gambler $C(t)$ along the game has unequal chance to be an odd or even number. In simple terms, no matter the starting capital of gambler is an odd or even number, the resultant capital is becoming and subsequently maintaining as an odd number after first several rounds of game B is played.

To demonstrate the concept, suppose the gambler holds an odd-number starting capital, for instance, 9 units of capital before playing game B. With the advancement of number of games played, his capital becomes “ $9 \rightarrow 15 \rightarrow 21 \rightarrow 27 \rightarrow 33 \rightarrow \dots$ ”. As the capital is always an odd number, game B is a winning game in the perspective of the gambler. Suppose the gambler begins game B by carrying an even-number starting capital, let’s say, 10 units of capital. With the advancement of number of games played, his capital is likely to be “ $10 \rightarrow 3 \rightarrow 9 \rightarrow 15 \rightarrow 21 \rightarrow \dots$ ”. After one round of game B is played, the initial even-number capital becomes and maintains as an odd number for the remaining rounds of game B. Therefore, no matter the starting capital for gambler is an odd or even number, game B is always a winning game to the gambler. There is no doubt that playing game B alone offers higher returns than playing the compound game, *i.e.*, coupling game B with a losing game, game A, which is also reflected in Figure 3 – 3.

In summary, although the statement, as claimed in [30], which states the individual losing games can be deterministically combined in the formation of a winning game is mainly caused by different probability distribution of individual games and compound game, is able to be regarded as a persuasive explanation on the working principle of Parrondo’s paradox. Unfortunately, the proposed simple capital-dependent game is failed to reproduce the analogous paradoxical effect delivered by either versions of Parrondo’s paradox.

3.3 Mathematical analysis of Parrondo's paradox

In reality, the entire counterintuitive phenomenon, which is generated by the compound game, or randomly mixed game, of capital-dependent Parrondo's paradox, can be analyzed by simply placing all the parameters in one single probability space. The probability space, as indicated in Figure 3 – 4, consists of two individual elements: a straight line (in red) and a curve (in black).

Figure 3 – 4: *Probability space of capital-dependent Parrondo's paradox*

The curve, as shown in Figure 3 – 4, is defined by the game rules. In order to make game B a fair game, the winning probability of game B should equal to the losing probability of game B. Such relationship can be mathematically described as $p_2 p_3^M = (1 - p_2)(1 - p_3)^M$. In the selected case, original capital-dependent Parrondo's paradox, the predefined integer is selected to be 3, i.e., $M = 3$. Therefore, in order to make game B a fair game, the selected probabilities in scenario 1 and 2 must satisfy equation 3.1.

$$\frac{p_2 p_3^2}{(1 - p_2)(1 - p_3)^2} = 1 \quad (3.1)$$

In addition, equation (3.1) can be modified by simply expressing winning probability of scenario 1, p_2 , in terms of winning probability of scenario 2, p_3 . The resultant function is equation (3.2), which is the curve (in black) in Figure 3 – 4.

$$p_2 = \frac{p_3^2 - 2p_3 + 1}{2p_3^2 - 2p_3 + 1} \quad (3.2)$$

Such curve, in reality, has one additional feature. It basically divides the entire probability space into two separate regions: the region above the curve is termed as the winning region (in grey) as the winning probability of game B is higher than the corresponding losing probability, *i.e.*, $p_2 p_3^2 > (1 - p_2)(1 - p_3)^2$; the region below the curve is termed as the losing region (in yellow) as the winning probability of game B is lower than the corresponding losing probability, *i.e.*, $p_2 p_3^2 < (1 - p_2)(1 - p_3)^2$.

In summary, if the selected probabilities of game B, p_2 and p_3 , are falling into the winning region, game B is a winning game in the perspective of the gambler. On the other hand, if selected probabilities of game B, p_2 and p_3 , are lying inside the losing region, game B is a losing game to the gambler.

Similarly, by setting probabilities $p_2 = p_3 = p_1$, equation (3.1) can be converted into equation (3.3).

$$p_1^3 = (1 - p_1)^3 \quad (3.3)$$

By solving equation (3.3), it results in one real solution, $p_1 = 0.5$, and two imaginary solutions $p_1 = \frac{1}{2} - \frac{\sqrt{3}}{2}i$ and $p_1 = \frac{1}{2} + \frac{\sqrt{3}}{2}i$. In these solutions, only the real solution has the physical meaning, which reveals the winning probability of game A should always equal to the corresponding losing probability. Such relationship can be expressed in terms of the straight line (in red) in the probability space, as shown in Figure 3 – 4. It implies the winning probability of game A should always be selected along this straight line.

Similarly, if the winning probability of game A is less than 0.5, *i.e.*, the portion of straight line that falls inside the losing region, game A is a losing game in the perspective of the gambler. On the other hand, if the winning probability of game A is larger than 0.5, *i.e.*, the portion of straight line that falls into the winning region, game A is a winning game to the gambler. If the winning

probability is exactly 0.5, *i.e.*, the selected probability of game A is right on the intersection point of the straight line and the curve, game A is a fair game.

As specified in game rules of original capital-dependent Parrondo's paradox, game A is a losing game. Therefore, the winning probability of game A, p_1 , is selected along the straight line (in red) in the losing region. In original capital-dependent Parrondo's paradox, winning probability of game A is 0.495. Game B is also a losing game and therefore two winning probabilities of game B, p_2 and p_3 , must be any single point, (p_2, p_3) , located inside the losing region. The selected probabilities of original Parrondo's paradox are $p_2 = 0.095$ and $p_3 = 0.745$. Therefore, it is possible to plot these two points, $(p_1, p_1) = (0.495, 0.495)$ and $(p_2, p_3) = (0.095, 0.745)$, inside the probability space.

The compound game is formed as a convex linear combination of these two games, game A and game B, by introducing one additional parameter, which is termed as mixing parameter, denoted by γ . It is defined as the probability of selecting game A to be played by the gambler. The resulting compound game is analogous to game B of capital-dependent Parrondo's paradox, which is a condition-based game. Suppose the instantaneous capital along the game, $C(t)$, is divisible by predefined integer M , the winning probability of compound game, let's say, p_{c_1} , can be expressed by equation (3.4).

$$p_{c_1} = \gamma p_1 + (1 - \gamma) p_2 \quad (3.4)$$

On the other hand, if the capital $C(t)$ is not a multiple of M , the winning probability of compound game, p_{c_2} , can be expressed by equation (3.5).

$$p_{c_2} = \gamma p_1 + (1 - \gamma) p_3 \quad (3.5)$$

As all these probabilities are fixed once they are selected, the only method is to tweak the value of mixing parameter γ . In order to make the compound game a winning game, the selected mixing parameter must satisfy equation (3.6), that is, the winning probabilities of compound game is greater than the corresponding losing probability if the predefined integer $M = 3$.

$$\frac{p_{c_1} p_{c_2}^2}{(1 - p_{c_1})(1 - p_{c_2})^2} > 1 \quad (3.6)$$

Such method can also be represented in the same probability space, as shown in Figure 3 – 4, by linking these two points, (p_1, p_1) and (p_2, p_3) , using a straight connecting line (in blue). It is able to observe there is certain region of the line falling inside the winning region. The probabilities fall inside this region satisfy equation (3.6), which makes compound game a winning game to the gambler. By adjusting the value of mixing parameter γ , *i.e.*, changing the location of the point along the straight line, any selected points along this straight line falling into the winning region are the keys in producing a winning expectation. In original capital-dependent Parrondo's paradox, mixing parameter γ equals to 0.5, which is the middle point of the straight line (in blue). Such point is located inside the winning region. Therefore, the compound game is a winning game. This is believed to be the secret behind Parrondo's paradox.

3.4 Alternative Parrondo's Paradox

Based upon the theoretical foundation created in section 3.3, it is possible to construct several alternative designs. Such designs can be used to explain how the analogous paradoxical effects can be generated by simply manipulating these parameters in the probability space.

3.4.1 Reverse Parrondo's paradox

It is started by providing a relatively simple alternative design, namely, reversed Parrondo's paradox, that is, two winning games can also be combined in producing a losing expectation.

The reversed Parrondo's paradox is achieved by simply switching the winning probabilities with its losing probabilities. Similarly, the selected probabilities can be plot inside the same probability space, as shown in Figure 3 – 5.

Figure 3 – 5: *Probability space of reversed Parrondo's paradox*

In such case, all these probabilities of game A and game B are falling inside the winning region of probability space. Therefore, game A and game B are winning games in the perspective of the gambler. In this case, by setting the mixing parameter γ to 0.5, the middle point is falling inside the losing region. In this arrangement, a totally reversed paradoxical effect is predicted to be produced. In order to preserve the consistence, a simulation is generated by averaging the outcomes of 10,000 trials for each game, and a total 200 games are played, as shown in Figure 3 – 6.

Figure 3 – 6: Simulation of reverse Parrondo's paradox

As indicated in Figure 3 – 6, both game A and game B are winning games in the perspective of the gambler if played isolatedly. On the other hand, the compound game is a losing game in the perspective of the gambler. One intriguing feature can be observed from this simple design. By comparing these two figures, Figure 3 – 4 and Figure 3 – 5, it is able to discover the selected probabilities are symmetric about the center point (or the intersection point of the straight line and the curve). It is caused by game rule of game B, which is also symmetric about the center point.

3.4.2 Alternative Parrondo's paradox

In reality, there can be totally eight different combinations of two winning and/or losing games, including the combinations that have been used in the original and reversed Parrondo's paradox. All these schemes and the results of individual games can be summarized in Table 3 – 1.

Scheme	Game A	Game B		
#1	Lose	Lose	Scenario 1	Lose
			Scenario 2	Win
#2	Win	Lose	Scenario 1	Lose
			Scenario 2	Win
#3	Lose	Win	Scenario 1	Win
			Scenario 2	Win
#4	Lose	Lose	Scenario 1	Lose
			Scenario 2	Lose
#5	Win	Win	Scenario 1	Win
			Scenario 2	Win
#6	Win	Lose	Scenario 1	Lose
			Scenario 2	Lose
#7	Lose	Win	Scenario 1	Win
			Scenario 2	Lose
#8	Win	Win	Scenario 1	Win
			Scenario 2	Lose

Table 3 – 1: *Different combinations of two winning and/or losing games*

The corresponding probabilities of these eight different combinations of two winning and/or losing games can be simply presented in the same probability space, as indicated in Figure 3 – 7.

Figure 3 – 7: *Probability space contains all probabilities of eight combinations of games*

Due to this specific feature of point symmetry, the analysis can be simply restricted to one side of the probability space, *i.e.*, scheme #1, scheme #2, scheme #4, and scheme #6, in which scheme #1 is the original capital-dependent Parrondo's paradox.

In scheme #2, game A is a winning game as its winning probability is greater than 0.5, game B is a losing game, which is exactly the same as that of theme #1. In this case, the compound game is also a winning game as the center point of the connecting line between these two points is falling inside the winning region.

In scheme #4, game A is a losing game, which is the same as that of scheme #1, game B is also a losing game as probabilities of both scenarios are falling inside the losing region. There is no doubt that the compound game is also a losing game.

Finally, in scheme #6, game A is a winning game as that of scheme #2. Game B is a strong losing game as that of scheme #4. The resultant compound game is also a losing game.

Based upon this one-sided analysis, it is possible to determine the results of the compound game for the remaining schemes. The compound games of scheme #3 (reverse scheme of scheme #6), scheme #5 (reverse scheme of scheme #4), scheme #7 (reverse scheme of scheme #2), and scheme #8 (reverse scheme of scheme #1) are winning game, winning game, losing game and losing game, respectively.

Such prediction can be testified through a series of simulations. In order to preserve the consistence of simulation results, the same values of biasing parameter ε and predefined integer M , that is, $\varepsilon = 0.005$ and $M = 3$, are used for all these simulation. Analogous to original version (scheme #1), the simulations are generated by averaging the outcomes of 10,000 trials for each game, and 200 games are played. The simulations for these themes are presented from Figure 3 – 8 to Figure 3 – 14.

Scheme #2

Game A			
p_i (Win)		$1-p_i$ (Lose)	
$0.5 + \epsilon$		$0.5 - \epsilon$	
Game B			
Is capital C(t) a multiple of M?			
<u>Scenario 1 (Yes)</u>		<u>Scenario 2 (No)</u>	
p_2 (Win)	$1-p_2$ (Lose)	p_3 (Win)	$1-p_3$ (Lose)
$0.1 - \epsilon$	$0.9 + \epsilon$	$0.75 - \epsilon$	$0.25 + \epsilon$

Figure 3 – 8: Game rule and simulation result of scheme #2

Scheme #3

Game A			
p_i (Win)		$1-p_i$ (Lose)	
0.5 - ϵ		0.5 + ϵ	
Game B			
Is capital C(t) a multiple of M?			
<u>Scenario 1 (Yes)</u>		<u>Scenario 2 (No)</u>	
p_2 (Win)	$1-p_2$ (Lose)	p_3 (Win)	$1-p_3$ (Lose)
0.9 + ϵ	0.1 - ϵ	0.75 - ϵ	0.25 + ϵ

Figure 3 – 9: Game rule and simulation result of scheme #3

Scheme #4

Game A			
p_i (Win)		$1-p_i$ (Lose)	
0.5 - ϵ		0.5 + ϵ	
Game B			
Is capital C(t) a multiple of M?			
<u>Scenario 1 (Yes)</u>		<u>Scenario 2 (No)</u>	
p_2 (Win)	$1-p_2$ (Lose)	p_3 (Win)	$1-p_3$ (Lose)
0.1 - ϵ	0.9 + ϵ	0.25 + ϵ	0.75 - ϵ

Figure 3 – 10: Game rule and simulation result of scheme #4

Scheme #5

Game A			
p_i (Win)		$1-p_i$ (Lose)	
$0.5 + \epsilon$		$0.5 - \epsilon$	
Game B			
Is capital C(t) a multiple of M?			
<u>Scenario 1 (Yes)</u>		<u>Scenario 2 (No)</u>	
p_2 (Win)	$1-p_2$ (Lose)	p_3 (Win)	$1-p_3$ (Lose)
$0.9 + \epsilon$	$0.1 - \epsilon$	$0.75 - \epsilon$	$0.25 + \epsilon$

Figure 3 – 11: Game rule and simulation result of scheme #5

Scheme #6

Game A			
p_i (Win)		$1-p_i$ (Lose)	
0.5 + ϵ		0.5 - ϵ	
Game B			
Is capital C(t) a multiple of M?			
<u>Scenario 1 (Yes)</u>		<u>Scenario 2 (No)</u>	
p_2 (Win)	$1-p_2$ (Lose)	p_3 (Win)	$1-p_3$ (Lose)
0.1 - ϵ	0.9 + ϵ	0.25 + ϵ	0.75 - ϵ

Figure 3 – 12: Game rule and simulation result of scheme #6

Scheme #7

Game A			
p_i (Win)		$1-p_i$ (Lose)	
0.5 - ϵ		0.5 + ϵ	
Game B			
Is capital C(t) a multiple of M?			
<u>Scenario 1 (Yes)</u>		<u>Scenario 2 (No)</u>	
p_2 (Win)	$1-p_2$ (Lose)	p_3 (Win)	$1-p_3$ (Lose)
0.9 + ϵ	0.1 - ϵ	0.25 + ϵ	0.75 - ϵ

Figure 3 – 13: Game rule and simulation result of scheme #7

Scheme #8

Game A			
p_l (Win)		$1-p_l$ (Lose)	
0.5 + ϵ		0.5 - ϵ	
Game B			
Is capital C(t) a multiple of M?			
<u>Scenario 1 (Yes)</u>		<u>Scenario 2 (No)</u>	
p_2 (Win)	$1-p_2$ (Lose)	p_3 (Win)	$1-p_3$ (Lose)
0.9 + ϵ	0.1 - ϵ	0.25 + ϵ	0.75 - ϵ

Figure 3 – 14: Game rule and simulation result of scheme #8

Scheme	Description	Paradoxical effect
#1	Lose + Lose = Win	Very strong
#2	Win + Lose = Win	Strong
#3	Lose + Win = Win	N/A
#4	Lose + Lose = Lose	N/A
#5	Win + Win = Win	N/A
#6	Win + Lose = Lose	N/A
#7	Lose + Win = Lose	Strong
#8	Win + Win = Lose	Very strong

Table 3 – 2: Summary of simulation results corresponding to different schemes of combinations

Scheme #1, as shown in Figure 3 – 2, is the original Parrondo's paradox, which is able to produce very strong paradoxical effects: if game A and game B are played isolatedly, they are losing games in the perspective of the gambler; the compound game, however, is a winning game to the gambler. Analogous to Scheme #1, the reverse Parrondo's paradox, scheme #8, as shown in Figure 3 – 14, is also able to generate very strong paradoxical effects: if game A and game B are played isolatedly, they are winning games to the gambler; the compound game, on the opposite, generates a losing expectation.

By observing the simulation results of Scheme #2 and its reverse scheme, Scheme #7, it is able to identify both schemes are capable of generating relatively strong paradoxical effects. In Scheme #2, as shown in Figure 3 – 8, game A is a slightly winning game, game B is a completely losing game. Intuitively, the compound game should be a slightly winning game. However, the simulation result reveals the compound game is definitely outperformed game A, which results in a completely winning game. The similar situation also occurs in its reverse scheme, Scheme #7, as shown in Figure 3 – 13. The only difference is that playing the compound game results in an obvious inferior position than playing game A, a winning game, alone.

From the simulation results of Scheme #3, as shown in Figure 3 – 9, and Scheme #6, it is possible to observe both schemes are belonging to the trivial cases. In Scheme #3, game A is a losing game, and game B is a complete winning game in both scenarios – scenario 1 and scenario 2. The trend of compound game is greatly influenced by the trend of game B. As the compound game is the game where game A and game B has equal chance to be played, it is very intuitive to have the instantaneous capital, denoted by $C(t)$, at any number of games, equals to the half the sum of that of game A and game B. Therefore, the generated phenomenon cannot be simply classified as the paradoxical effect.

The reverse scheme of Scheme #3, Scheme #6, is also a trivial case. The only difference between these two schemes is that the results for these games, game A, game B, and compound game, are completely reversed. On the other hand, the same situation applies to Scheme #6. As shown in Figure 3 – 12, the trend of compound game is mainly influenced by the trend of game B, and the instantaneous capital equals to the half the sum of that of game A and game B. Therefore, the produced expectation is hardly to be classified as a mathematical paradox.

The similar situation applies to Scheme #4 and its reverse scheme, Scheme #5. Both cases can be classified as trivial cases. In Scheme #4, as shown in Figure 3 – 10, the winning probabilities of these two games, game A and game B, are less than the corresponding losing probabilities.

Therefore, there is no doubt that the compound game is a losing game in the perspective of the gambler. The similar situation applies to its reverse scheme, Scheme #5. As shown in Figure 3 – 11, both game A and game B are winning games to the gambler. It is very intuitive to have the compound game is also a winning game.

In summary, from the simulation results, Scheme #1 and Scheme #8, are able to produce very strong paradoxical effect. Scheme #2 and Scheme #7 are capable of producing relatively strong paradoxical effects. For the remaining schemes, they are belonging to the trivial cases as they are failed to produce any paradoxical effects. All these schemes and corresponding results are summarized in Table 3 – 2. These results are totally identical to the predicted results.

3.5 Effects of Other Parameters

In addition to the study of other eight combinations of two winning and/or losing games, it is also interested to investigate whether the generated phenomenon is sensitive to other parameters that had been employed in the paradox. In this section, the effect of three different parameters are studied – the effect of predefined integer M (section 3.5.1), different mixing sequences of two games (section 3.5.2), and the non-linear combination of two games (section 3.5.3).

3.5.1 Effect of predefined integer M

From previous design of Parrondo's paradox, it is able to observe the fact that capital-dependent Parrondo's paradox can be reproduced as long as there exists a connecting line of two selected points of probabilities across the curve boundary with two probability points located in the losing region and the middle point lying inside the winning region. Therefore, it is possible to modify the probability curve based on this observation. The modification can be done by simply changing the value of predefined integer from $M = 3$ to $M = 5$. After such modification, the equation (3.2) is transformed to equation (3.7) as stated below.

$$p_2 = \frac{p_3^4 - 4p_3^3 + 6p_3^2 - 4p_3 + 1}{2p_3^4 - 4p_3^3 + 6p_3^2 - 4p_3 + 1} \quad (3.7)$$

Based on equation (3.7), it is possible to plot the modified probability curve (in solid black line) together with the original curve (in dash grey line) in one probability space, as shown in Figure 3 – 15.

Figure 3 – 15: *Probability space of original and modified Parrondo's paradox*

Similarly, the new probability curve (the solid black line in Figure 3 – 15) divides the entire probability space into two regions – winning region (yellow-line shaded area) and losing region (grey-line shaded area). The original probability curve ($M = 3$) and modified probability curve ($M = 5$) have one property in common, that is, both versions are symmetric about the intersection point of curve boundary and straight line. Due to this specific property, probability of game A remain unchanged in this case, that is, $p_1 = 0.495$.

On the other hand, the original probabilities of game B, $(p_2, p_3) = (0.095, 0.745)$, is no longer feasible as it is no longer galling inside the losing region. By adjusting the points to a new location in the losing region of probability space, let's say, $(p_2, p_3) = (0.095, 0.625)$, the paradoxical effect can be reproduced. The original and the adjusted points are presented in Figure 3 – 15. However, there is certain region of the new connecting line located in the winning region. By manually controlling the location of resultant compound game, *i.e.*, modify the value of mixing parameter, the resultant compound game can also be a winning game. Based on these values, and

by retaining parameters unchanged, the simulation of this new case can be generated as shown in Figure 3 – 16.

Figure 3 – 16: *Simulation of modified capital-dependent Parrondo's paradox*

3.5.2 Effect of mixing sequence

In previous sections, the compound game is mixed in a stochastic method, *i.e.*, deciding which game to be played is merely depending on the mixing parameter γ . In the original Parrondo's paradox [5], the compound game can also be generated in a deterministic way, *i.e.*, the mixing sequence is fixed on a sequence of AABBAABB.... The simulation result indicates the deterministically mixed compound game is slightly outperformed the stochastically mixed compound game. In this section, the emphasis is lying on whether the result is likely to be affected in other alternative mixing sequences.

In order to simplify the representation of the sequence of mixing, it is possible to express the sequence in terms of $[A_a B_b]$, where a is the number of game A is continuously played, b is the number of game B is continuously played and $a, b \in \mathbb{Z}_+$. Such that, the fixed mixing sequence of AABBAABB... in original Parrondo's paradox can also be represented by $[A_2 B_2]$.

The effect of the mixing sequences can be studied based on the same number of game A and game B is played. The only difference between these sequences is lying on the switching frequencies. Under this test scenario, the mixing sequences is selected to be $[A_3 B_3]$ and $[A_5 B_5]$. The simulation is generated based on these mixing sequences as shown in Figure 3 – 17.

Figure 3 – 17: *Simulation of Parrondo's paradox of different mixing sequences*

Figure 3 – 17 reveals the truth that different mixing sequences are likely to affect the outcome of the compound game. The instantaneous capital, denoted by $C(t)$, is proportional to the frequency of switching between these two games, game A and game B. The higher the frequency of switching between these two games, the higher the reward of the gambler obtained from playing the compound game. On the other hand, the lower the frequency of switching between these two games, in this case, the lower the reward of the gambler obtained from playing the compound game.

3.5.3 Effect of non-linear combination of two games

In previous cases, the compound game is formed in terms of a convex linear combination of two individual games. In reality, the formation of compound game can also be a non-linear combination. In the following case, the concept of non-linear combination of two individual games is demonstrated to outperform the original linear combination of two games. The same case, which has been used in section 3.5.1 (predefined integer $M = 5$) is used in this demonstration. There are totally three nonlinear combinations are selected in this in this demonstration. The functions of these three non-linear combinations are equation (3.8), (3.9), and (3.10). These functions are determined based on two probabilities points.

$$p_3 = -0.2p_2^2 - 0.207p_2 + 0.64647 \quad (3.8)$$

$$p_3 = -0.5p_2^2 - 0.03p_2 + 0.63236 \quad (3.9)$$

$$p_3 = -p_2^2 + 0.265p_2 + 0.60885 \quad (3.10)$$

All these functions can be visually expressed in Figure 3 – 18. The original linear combination of two games is expressed in terms of a green solid line. The equation (3.8) is presented by a pink dashed curve, which is labelled as “new compound game #1”. The equation (3.9) is presented by a blue dashed curve, which is labelled as “new compound game #2”. The equation (3.10) is presented by a red dashed curve, which is labelled as “new compound game #3”. Only the middle points of these dashed curves are selected as the probabilities of the non-linear compound game.

The simulation is produced based upon the same parameters that have been employed in previous case. Only in this case, deciding which game to be played is longer depending on the mixing parameter γ . In addition, the executional procedures are slightly different: two probabilities of compound game, p_{c_1} and p_{c_2} , are firstly determined. As both compound game and game B of capital-dependent Parrondo’s paradox are sharing the same feature, *i.e.*, condition-based games, it is possible to employ the probabilities of the compound game under the paradigm of game B.

Figure 3 – 18: *Non-linear and linear combination of two games in probability space*

Figure 3 – 19: *Simulation of non-linear combinations of two games*

As shown in Figure 3 – 19, the similar paradoxical effects can be reproduced even the compound game is formed in the way of non-linear combination. In reality, one additional intriguing feature can be observed from the simulation result, that is, the capital of the gambler is proportional to the distance between selected probabilities of compound games to the curve boundary, *i.e.*, the gambler earns the greatest amount of capital by playing the new compound game #3 as its middle point of the curve has the largest distance from the curve boundary.

3.6 Conclusion

The chapter begins by summarizing two versions of Parrondo's paradox, known as capital-dependent and history-dependent Parrondo's paradox. Since the first appearance of the paradox, there are several issues surrounding Parrondo's paradox. Some of the issues have been responded by its initiator. The issue, which specifies the similar paradoxical effect can be reproduced by changing the original framework to a simple capital-dependent game, is verified in section 3.2. It is proved that the simple capital-dependent game is failed to reproduce the paradoxical effect. In reality, the phenomenon generated by Parrondo's paradox can be simply explained by placing all the probabilities, which have been used in game A and game B, into a probability space, as mentioned in section 3.3. In that sense, game A is a losing game as its probability is located inside the losing region of probability space. Game B is a losing game as its probabilities are also located in the losing region of probability space. On the other hand, the stochastically mixed compound game is a winning game as its probabilities are convex linear combination of probabilities of two individual games, which is falling inside the winning region of probability space. In short, it can be understood as employing the compound probabilities, $p_{c_1} = 0.295$ and $p_{c_2} = 0.56$, under the paradigm of game B.

Based upon the established theoretical foundation, it is possible to develop a series of alternative Parrondo's paradox, namely, reverse Parrondo's paradox and other six combinations of two winning and/or losing games, in section 3.4. In these combinations, only four out of eight schemes, including scheme #1, scheme #2, scheme #7, and scheme #8, are managed to produce the similar paradoxical effects.

In addition to that, other parameters, including the value of predefined integer M , mixing sequences of game A and game B of deterministically mixed compound game, and non-linear combination of two games, may likely to affect the paradox. These concerns are verified in section 3.5. The value of predefined integer M is likely to affect the probability boundary in the probability space. It is possible to adjust the value of probabilities according to the modified probability boundary, and the identical paradoxical effect can be reproduced. The mixing sequences of game A and game B is likely to affect the outcome of deterministically mixed compound game. In reality, the instantaneous capital, denoted by $C(t)$, is proportional to the frequency of switching between these two games, game A and game B. Finally, the non-linear combinations of two games, game A and game B, are likely to produce the same paradoxical

effects. And the capital of the gambler is proportional to the distance between selected probabilities of compound games to the curve boundary.

In fact, the actual application of Parrondo's paradox can also be regarded as an appealing demonstration, which reveals the importance of developing the fuzzy logic gate. In contrast with traditional logic, or two-valued logic, fuzzy logic is a multi-valued logic which has a truth value varying between 0 and 1. Linguistic-wise speaking, fuzzy logic is believed to be much closer to the concept used in the natural language, which has vague borders and fuzzy meaning. The Parrondo's paradox could be served as a mathematical tool in demonstrating how random noise can alter the outcome of the entire system.

In the scheme of Parrondo's paradox, both game A and game B are losing games in the perspective of the gambler. If the term "losing" is defined under the traditional logic, both game A and game B equal to 0. No matter what additional conditions applied to game A and game B, the mixed game is always a losing game (0). However, in the framework of fuzzy logic, the term "losing" means if the individual games are played long enough, the overall trend for gambler is losing. This does not count the result of each individual trial, whether it is a gain or loss of the capital. It only specifies under a large amount of trials, the number of losing trials is greater than the winning trials. Therefore, with the addition of mixing parameter, the entirely reversed outcome (or paradoxical effect) can be generated.

Chapter 4

Financial Option Insurance

As provoked by both versions of Parrondo's paradox, both capital-dependent and history-dependent Parrondo's paradox, any losing strategies may lead to positive outcomes if all these strategies are appropriately arranged [4 – 8]. It is very intuitive to correlate its potential applications in contemporary financial market leading to the construction of well-diversified portfolio by combining several financial instruments. The constructed portfolio based upon the framework of Parrondo's paradox may offer higher potential returns, which outperforms the most frequently used delta-neutral portfolio. However, after a throughout analysis of both versions of Parrondo's paradox, as specified in Chapter 3, it is possible to make the conclusion that Parrondo's paradox rarely have any potential applications in the contemporary financial market due to three reasons as discussed in section 3.3.

Nowadays, any market participants are unavoidably exposed their investments to the increasing volatility and uncertainty of financial market. The adverse market movements can spell catastrophic consequences to market participants, from individual investors to large financial institutes. Unfortunately, the entire financial market is highly concentrated on several large financial institutes. Any problem occurs to these institutes, the impact on the entire financial market is tremendous, which may ultimately lead to the financial crisis. Therefore, in the perspective of market participants, it is always ideal to seek out an avenue to reduce the potential risk to their investment. Even in the perspective of the government, it is always optimal to introduce one additional entity to contemporary financial market, which helps to redistribute and regulate the risks among entities. Based on that, the concept of financial option insurance is proposed in this chapter. Unlike the traditional way of hedging, which is achieved by introducing negatively correlated financial derivatives into an existing financial portfolio, financial option insurance can be issued by an independent institution, which is very alike to an insurance company dedicated in option market.

This chapter begins by providing a conclusive introduction section, section 4.1, to those who are not very familiar with the financial fundamentals. The fundamental business model of option

insurance company is drafted in section 4.2, which includes two types of insurance policies targeting two groups of potential customers. To enhance the functionality of financial option, a business strategy of matching is provided in section 4.3. In section 4.4, the feasibility of financial option insurance is discussed in terms of three aspects, including market acceptance, option insurer profitability, and simulated positive market effect with the involvement of option insurer.

4.1 Introduction

Contemporary financial market is always associated with the increasing volatility and uncertainty. The market participants, no matter individual investors or institutional traders, are unavoidably exposed to the risk³ infused in the market. Therefore, there is always a saying among traders: “trading may have princes, but nobody stays a king.”[93]. In order to beat the market, agile and accurate anticipations are the fundamental skills for investors, fund managers, and corporate financial managers. Any inappropriate strategies can simply spell catastrophic consequences to their representing clients as well as organizations due to large amount of money involved.

One of the most fundamental roles performed by financial professionals is to measure and quantify the risk which leads to the construction of a well-diversified portfolio⁴, such that, balances out or limits the exposure of underlying asset to any possible market fluctuations [94].

The most actively traded financial derivative⁵ is option, which represents a contract sold by an option writer to a holder in exchange for an option premium⁶. The contract offers holder a right, but not an obligation, to either call or put an underlying financial instrument at strike price⁷ on a specified exercise date⁸. If the option is in-the-money (ITM)⁹ on its exercise date, option holder chooses to exercise the option, therefore, the option writer is obliged to deliver (for call option) or take delivery (for put option) of underlying financial asset to or from option holder, respectively. However, if the execution condition is unfavorable to option holder on the exercise date, that is, the option is either at-the-money (ATM)¹⁰ or out-of-the-money (OTM)¹¹, he has the right to give up the right and let it be expired worthlessly, which directly incurs a loss of entire option premium.

³ **Risk:** in financial market, three types of risks are generally encountered by market participants, including market risk, opportunity risk, and inflationary risk.

⁴ **Portfolio:** a collection of negatively correlated financial instruments.

⁵ **Derivative:** a financial instrument whose value is merely depending on its underlying asset.

⁶ **Option premium:** the market price of option during the transaction between option writer and holder.

⁷ **Strike price:** the price at which the option holder buy or sell underlying security as specified in the contract.

⁸ **Exercise date:** a date specified on option contract at which a European option holder must decide whether to exercise the option or let it expires worthlessly.

⁹ **In-the-money (ITM):** the strike price of call option is lower than spot price of its underlying security or the strike price of put option is higher than spot price of its underlying security.

¹⁰ **At-the-money (ATM):** the strike price of call/put option is close or equal to the spot price of its underlying security.

¹¹ **Out-of-the-money (OTM):** the strike price of call option is higher than spot price of its underlying security or the strike price of put option is lower than spot price of its underlying security.

With the introduction of options into the prevailing financial market, the variety of trading strategies is significantly improved. In general, there are two primary reasons for a market participant to purchase an option: 1) speculating on the market movement and employ options as means of leverage to improve its gains. The market participants of this category is generally described as *speculators*; 2) utilizing options to hedge a position of protecting underlying security or portfolio against potential adverse market movement. The market participants of this category are generally regarded as *portfolio managers*. In the perspective of portfolio manager, financial options can be deemed as a special “insurance” in their trading strategies [95].

However, investing in option is still risky. As is known to all, the economic value of financial option consists of two parts, known as intrinsic value and time value (or time premium). The intrinsic value is directly related to the spot price and volatility of its underlying financial asset, while its time value vanishes once it approaches the specified exercise date. Therefore, in certain perspective, option can be deemed as one of the most volatile trading instruments as its risk is not only limited to the market uncertainties but also related to the time remaining to its exercise date. Such risk has been technically reported in [96 – 100]. The risk becomes much more severe for option speculators with a totally unprotected position. In simple words, option trader has to predict the trend of underlying security within a given period. Even for portfolio managers, it is ideal to have their money, which is used in hedging, to be compensated, if the option is failed to be exercised.

Based upon the motivations as specified above, the concept of option insurance is firstly introduced in the chapter. It integrates two starkly-dissimilar concepts of insurance and financial options, which allows any option holders to claim partial compensation to mitigate their cost of the option if they have purchased the option insurance by paying an additional insurance premium during the transaction of their option. The option insurance can be either sold by an existing financial institution as an extra service to interested investors or a dedicated company. For the sake of convenience, the organization, which provides the service of option insurance, is referred to as the *third entity*.

The proposed concept of option insurance constitute two independent policies which enable it to be fitted into the trading strategies of speculators, which are known as *speculating insurance policies* and portfolio managers, which are known as *hedging insurance policies*. In order to guarantee the business holds a no-loss position and independent of the financial market trend, it is always ideal to seek a pair of option investors with totally reversed market expectations, that is, a call and put of identical strike price and expiration date. However, the trend of an actively-traded

financial instrument is biased on most occasions, *i.e.*, every single price fluctuations of a financial instrument reflects the different future expectations and applied strategies of market participants. In order to enhance the functionality of option insurance in a realistic financial market, business strategy of matching is proposed, which allows optimal combination of calls and puts with different moneyness, at the same time, minimizes the insurer's exposure to any potential losses.

4.2 Business Model

The fundamental business model for an option insurance company, namely the third entity, consists of two major aspects, speculating insurance policies and hedging insurance policies, which aims to target two different groups of market participants – speculators and portfolio managers.

4.2.1 Speculating Insurance Policies

The purpose of speculating insurance is to allow third entity to sell insurance to option speculators, mainly for those who opened a naked option position, in protecting their option against adverse market movements. The insured speculators are entitled to claim reimbursement from the third entity when the corresponding bet goes wrong.

The speculating insurance policies can be briefly summarized in terms of five (5) major statements as indicated below:

1. Option insurance is available for speculators from both parties, call option and put option buyers, at any time during the transaction of option before its exercise date (or expiration date).
2. Insurances are sold in pairs, *i.e.*, when there is a match in the moneyness and expiration date of call and put option buyers.
3. The total payable option insurance premiums are divided and shared by paired option speculators at a predetermined percentage. It is determined by a pricing structure.
4. Third entity only compensates/reimburses the party whose option is out-of-the-money (OTM) at maturity, with the amount of reimbursement based on the percentage that the investors paid for their options.
5. If the option is in-the-money (ITM) or at-the-money (ATM) at maturity, due to any reasons, the option holder failed to exercise the option; third entity does not reimburse the insurance buyer, profiting collected insurance premium.

In order to demonstrate the concept of speculating insurance policies, assuming there are two speculators, let's say, speculator **A** and speculator **B**. Suppose both speculators are interested in a

specific European stock *IKEA*¹², and employing option as means of leverage to improve their potential return on investment. However, these two speculators have entirely reversed future expectations on *IKEA* – speculator **A** expects it to go bullish, while speculator **B** anticipates it to go bearish. Under current spot price of *IKEA*, which is $S_0 = \$505$, speculator **A** decides to purchase 100 shares of *IKEA Feb 2013 500 Call* @ premium price of $C = \$24$ ¹³; on the opposite, speculator **B** decides to purchase 100 shares of *IKEA Feb 2013 500 Put* @ premium price of $P = \$15$. Without purchasing the option insurance, speculator **A** and **B** have potential to lose their entire option premium if the trend of market is unfavorable to both of them, which is described in Figure 4 – 1.

The situation is significantly improved if both speculators decide to purchase option insurance from third entity, which is employed to protect the naked positions. As their options have reversed expectations, one call and one put, with identical expiration date and moneyness, they are allocated by third entity to be paired investors. The allocation is determined by a business strategy of matching, which is explained in section 4.3. Third entity charges a premium (for option insurance) to each of paired speculators in order to insure them against their potential losses. The critical component in option insurance is to calculate on how much the third entity should charge for the premium in order to be profitable. It operates in a manner that it charges the premium based on a specific pricing structure. At current stage, it charges 50% (this percentage can be subjected to changes depending on the actual market conditions, 50% means the chance of specific financial instrument go bullish or bearish is identical) of the maximum value between market prices of corresponding call and put options.

In this case, option premium of call (\$24) is more than that of put (\$15), *i.e.*, $C > P$. Based on the given pricing structure, third entity charges an insurance premium of \$12¹⁴ for undertaking the risk of insured option speculators, call option and put option speculators. This amount is split evenly (due to the 50% yardstick) between speculator **A** and speculator **B** such that each of them pays only \$6 as insurance premium. The entire transaction process of three-entity structure is described in Figure 4 – 2.

¹² *IKEA*: trading quote.

¹³ *IKEA Feb 2013 500 Call @ 24*: each option quote consists of five components – option symbol (*IKEA*), exercise date (Feb 2013), strike price (\$500), option type (call option), and option premium (\$24).

¹⁴ *Insurance premium*: $\frac{1}{2} \times \max(\text{option premium of call}, \text{option premium of put})$

Figure 4 – 1: Risk and reward graph for call option and put option speculators

Under the first scenario, assuming speculator **A**'s prediction is right, that is, the price of underlying security *IKEA* is increased to \$555 (*i.e.* $S_{t1} = \$555$) at option's exercise date. Speculator **A**, who brought a call option, expected the price of underlying security to rise, choose to exercise his option and gains a net profit of \$2,500 [$= 100 \times (\$555 - \$500 - \$24 - \$6)$]. At the same time, he is not valid to claim the reimbursement from the third entity as "misfortunate" does not befall on him.

On the contrary, under the identical scenario, speculator **B**'s predication goes wrong as he expected the price declination of underlying security. Since his put option is out-of-the-money (OTM) at maturity, it expires worthlessly, which incurs a net loss of \$1,500, the entire amount for him to purchase the put option. However, if speculator **B** had purchased the option insurance, he is entitled to receive a reimbursement of \$750 ($= 100 \times 50\% \times \15) from third entity, whereas he only paid \$600 for the insurance. As a consequence, he stands to gain a net profit

of \$150 purely from the transaction between him and third entity alone. In such a case, third entity makes a net profit of \$450.

Figure 4 – 2: *Transaction among three-entity structure with financial options and option insurance*

Conversely, if the situation is reversed, whereas the spot price of its underlying security plunges to \$455 (*i.e.* $S_{t2} = \$455$) at maturity, the “misfortunate” has befallen on speculator **A** – his call option expires worthless. Under this scenario, he is valid to claim a reimbursement of \$1,200 ($= 100 \times 50\% \times \24) from the third entity, whereas he only paid \$600 for his option. As a result, he stands to profit \$600 purely from the transaction between him and the third entity.

Under this scenario, speculator **B** is not capable of claiming any reimbursement. The third entity does not make any profit or suffer any loss as entire collected premium from both speculators is paid to speculator **A** as compensation.

In both scenarios, both speculators stand to gain, even in the case of wrong bets. Similarly, third entity earns either \$0 or \$450, that is, an expected value (EV) of \$2.25 for each share of option

insurance if both scenarios have equal chance of occurring. Hence, it can be clearly observed that the proposed option insurance creates a win-win situation among all entities involved, which is described in Figure 4 – 3.

Figure 4 – 3: *A win-win situation for both speculators and third entity under different scenarios*

The main benefit of introducing third entity, option insurer, into original two-entity framework is that it can act as an insurance company, which quantifies the risk and helps to redistribute it over more market participants, creating a win-win situation for all entities involved. The risk and reward for speculator A and B with and without the involvement of option insurance are compared in Figure 4 – 4. The option insurance is acting as a tool to spread initial risk among more entities, resulting in a lower risk for each market participants.

Figure 4 – 4: Risk and reward graph for speculators with or without option insurance

4.2.2 Hedging Insurance Policies

Hedging is a specific financial strategy, which is frequently employed by portfolio managers to reduce the exposure of their position to any potential risk, mainly the market risk, which they may encounter, by constructing portfolio with a combination of negatively-correlated financial instruments. Such strategy generally involves one or several financial derivatives, for instance, option.

The hedging strategies usually involve two possible trading positions of a particular financial instrument, namely long position and short position. Entering into a long position implies the investor buys and owns the corresponding financial asset and profits from the position when the price of corresponding financial instrument increases. Entering into a short position implies the investor borrows the asset from another entity (usually a brokerage firm) and sells it immediately in exchange of a credit in his trading accounting. As the price of financial asset decreases, the investor can purchase it back at a lower price and return it back to the lender, profiting from the

difference between selling and buying price of the corresponding financial instrument. In order to open a short position, a margin account¹⁵ is compulsory.

The hedging insurance policies are similar to that of speculating insurance, except the reimbursement clause is reversed:

1. Option insurance is available for portfolio managers from both parties, call option and put option buyers, at any time during the transaction of option before its exercise date (or expiration date).
2. Insurances are sold in pairs, *i.e.*, when there is a match in the moneyness and expiration date of call and put option buyers.
3. The total payable option insurance premiums are divided and shared by paired portfolio managers at a predetermined percentage. It is determined by a pricing structure.
4. The third entity only compensates/reimburses the party whose option is in-the-money (as a consequence, the entire portfolio incurs a loss of capital) at maturity, with the amount of reimbursement subjected to the percentage that the portfolio managers paid for their options.
5. If the option is at-the-money (ATM) or out-of-the-money (OTM) at maturity, third entity does not reimburse the insurance buyer and profits entire collected insurance premium.

Similarly, in order to demonstrate the concepts of hedging insurance policies, assuming there are two portfolio managers, let's say, **C** and **D**, are interested in the same European stock *IKEA* with entirely reversed future expectations – portfolio manager **C** expects it to go bullish, while portfolio manager **D** anticipate it to go bearish. Therefore, in the trading strategy of portfolio manager **C**, it combines a long position on the stock (purchases 100 shares of *IKEA* at spot price of \$505) and a long position on put option (purchase 100 shares of *IKEA Feb 2013 500 Put* at option premium of \$15 each) in his portfolio. Portfolio manager **D**, on the other hands, integrates a short position on stock (short 100 shares of *IKEA* at spot price of \$505) and a long position on call option (buy 100 shares of *IKEA Feb 2013 500 Call*) in his portfolio. In both cases, by employing hedging strategy in their trading, the risk has been greatly limited. With the introduction of hedging insurance in their portfolio, their outcome is even better off, as shown in Figure 4 – 5.

¹⁵ **Margin account:** a margin account is a typical customer account. For each margin trades, the customer is allowed to put up a certain percentage of total cost of the trade amount in cash and the remaining in account as a form of deposit.

Figure 4 – 5: Risk and reward graph for portfolio managers with and without option insurance

In summary, the clauses proposed in the insurance contract targeting two groups of market participants, the speculators and portfolio managers, are compared in Table 4 – 1.

Speculators <i>Speculating on future price movement using option as a means of leverage</i>	Portfolio managers <i>Employing option as a negatively correlated derivative such that reduce the exposure of underlying security to market uncertainties</i>
1. Reimburse the investors who purchase call option when spot price at maturity is below spot price when insurance contract was signed.	1. Reimburse the investors who purchase call option when spot price at maturity is above spot price when insurance contract was signed.
2. Reimburse the investors who purchase put option when spot price at maturity is above spot price when insurance contract was signed.	2. Reimburse the investors who purchase put option when spot price at maturity is below spot price when insurance contract was signed.

Table 4 – 1: Reimbursement clauses for speculators and portfolio managers

4.3 Business Strategy of Matching

Business strategy plays a crucial role for third entity to minimize its potential losses, and at the same time, maximize its profitability. There are two major business strategies adopted by option insurance company, the third entity. One of them is to determine appropriate pricing structure, which has been mentioned in section 4.2. The yardstick of 50% suggests that both insurance premium and reimbursement should be tweaked according to the prevailing financial market condition such that the third entity can immediately respond to rapid changes.

Under the perfect market assumption, there is always a balance between two equal-sized parties of calls and puts. Such that, a plenty number of financial derivative investors can be protected by option insurance as demanded by in the insurance policies, that is, option insurance is sold in pairs. By following such assumption, no matter the price of underlying security ascended or plunged at maturity, under current version of reimbursement clauses, it always allow the third entity to stand in a no-loss position – third entity only reimburses one of the paired speculators, whose option is out-of-the-money (OTM), or one of two portfolio managers, whose option is in-the-money (ITM), and stands a chance to profit entire insurance premium collected from both paired investors once their options are at-the-money (ATM) at maturity. Such strategy is depicted in Figure 4 – 6 (a).

In reality, such balance between two equal-sized parties of calls and puts rarely exists in real financial market. There are always unequal-sized parties of calls and puts with different moneyness, due to different future expectations of financial market and applied strategies. In order to enhance the functionality of option insurance in such an imperfect financial market, a business strategy of matching is firstly proposed in this section, which allows the optimal matching between calls and puts of different moneyness in the category of both speculators (speculating insurance) and portfolio managers (hedging insurance). As the only difference between two insurance policies, speculating insurance policies and hedging insurance policies, is depending on their corresponding reimbursement clauses, the business strategy of matching is demonstrated in the perspective of option speculators.

4.3.1 Three possible scenarios of matching

In summary, the matching between calls and puts of different moneyness, or strike price, are likely to fall into one of the following three possible scenarios: 1) strike price of call option equals to that of put option, *i.e.*, $K_C = K_P$; 2) strike price of call option is lower than that of put option, *i.e.*, $K_C < K_P$; 3) strike price of call option is higher than that of put option, *i.e.*, $K_C > K_P$.

Figure 4 – 6: *Three possible scenarios of matching between different moneyness of calls and puts*

The first scenario, where $K_C = K_P$, is the ideal case for third entity as it is under the perfect market assumption. As shown in Figure 4 – 6 (a), there are two overlapping horizontal lines – green one (above) stands for the strike price of put option, and the red one (below) stands for the strike price of call option. In fact, these two lines are overlapping each other. The line breaks the entire quadrant into two regions – yellow region (above the horizontal line) and pink region (below the horizontal line). The concept is that if the separate price of underlying security of financial option with various exercise date (along horizontal axis) retains either in yellow region or pink region, the third entity reimburses insured put option buyer or call option buyer, respectively. However, if the price of underlying security at any exercise date of option is place right on the overlapping lines, third entity is going to profit the entire insurance premium collected from both insured call and put option speculators, who have purchased the option insurance.

The second scenario, where $K_C < K_P$, is the most preferable one of three possible scenarios in the standing of third entity. As compared to the diagram of first scenario, Figure 4 – 6 (a), these two horizontal lines are no longer overlapping – the green line is right on top of the red line in this new scenario. In addition to that, there is one additional region in the middle of two original regions, the blue region, as shown in Figure 4 – 6 (b-1). If the price of underlying security at any specified exercise date falls into this region, third entity profits all premiums collected from paired investors. Under the same scenario, if the difference between these two strike prices is increased, as depicted in Figure 4 – 6 (b-2), the blue region is enlarged accordingly, which implies third entity stands a higher probability to profit all collected premium without reimbursing any single party of paired investors.

The third scenario, where $K_C > K_P$, is a reversed version of second scenario. In Figure 4 – 6 (c-1), a new region (in dark purple color) is created between two horizontal lines, which indicates the strike prices of corresponding call and put options, by overlapping the original yellow and pink regions. On the contrary, if the price of underlying security at any exercise date, along the horizontal axis, falls into the dark purple region, it will incur a huge loss to third entity as it has to reimburse both speculators. The situation becomes severe if the difference between two strike prices is increased, as shown in Figure 4 – 6 (c-2).

Three possible scenarios of matching between options of different moneyness as mentioned above suggest that the most preferable scenario in the perspective of third entity is the second one. And the level of preference is proportional to the difference between two strike prices of corresponding options. On the other hand, the least desirable scenario to third entity is the third

scenario, which becomes more severe if the difference between these two strike prices is increased. The business strategy of matching is therefore becomes the determination of optimal pairs of call option and put option investors, which minimize the occurrence of third scenario and maximize the occurrence of second scenario.

4.3.2 Matching strategy

Based upon three possible scenarios of matching, it is able to draft a simple business strategy of matching among calls and puts of different moneyness. It begins by separating a group of investors, who demand for option insurance to protect their naked positions, into two subgroups of totally reversed reimbursement clause. For each call option of different strike prices, third entity ranks its preference of matching, in accordance with three possible scenarios of matching, to the put options of different strike prices. Similarly, for each put option of different strike prices, third entity also ranks its preference of matching to the call option options of different strike prices with respect to three possible scenarios.

In such a matter, the original problem is simplified into a problem of determining a stable assignment between calls and puts with specified rankings. Furthermore, the rankings of preference can be tabulated in the form of matrix, called *ranking matrix*. Each entry of ranking matrix consists of two numbers (i, j) , whereas the first number i is the ranking of put options given by third entity in the correspondence to each call option, and the second number j is the ranking of call options given by third entity in the correspondence to each put option.

Once the ranking matrix is established, the remaining procedures of matching become relatively simple. First of all, all of the call options are proposed to put options in terms of their first preference, *i.e.*, $i = 1$, in the ranking matrix. After that, based on the second number of each entry, that is j as specified in ranking matrix, the put options, which have been proposed by call option, retain the one with highest ranking and reject the remaining. Followed by, all unassigned call options propose to put options in terms of their second preference, *i.e.*, $i = 2$. The entire propose-and-reject procedures are likely to repeat until all call options have been successfully accepted by put options. The result of assignment is likely to be the stable matching in the view of third entity (option insurer) as the result completely satisfies the recommendation obtained from three possible

scenarios of matching, which minimize the insurer's chance of exploring to potential losses, and at the same time, boosting its profiting probability.

4.3.3 Example

Assuming there are a group of eight option speculators – half of them are call option speculators $\{A, B, C, D\}$, the remaining of them are put option speculators $\{\alpha, \beta, \gamma, \delta\}$. To protect their naked option position against potential losses, they seek to purchase option insurance from third entity. The options, calls and puts, purchased by corresponding speculators, are of different strike prices as specified in Figure 4 – 7 (a).

The very beginning task of third entity is to identify the stable matching among these investors before initializing appropriate insurance contract. It begins by ranking the preference of matching in the perspective of each call option speculators to the put option speculators based on the result obtained from three possible scenarios of matching. As indicated in Figure 4 – 7 (b), in the perspective of third entity, for call option speculator **A**, the most preferable combination is likely to be put option speculator δ as such combination satisfies the third scenario of matching, *i.e.*, $K_C > K_P$, followed by put option speculator α , put option speculator γ , and the least preferable combination is put option speculator β . Therefore, in the perspective of third entity, the final ranking for call option speculator *A* is likely to be $\{\alpha, \beta, \gamma, \delta\} = \{2, 4, 3, 1\}$, which is recorded in the first number of the entries within the first row of ranking matrix, as indicated in Figure 4 – 7 (c). By completing the remaining ranking assignment procedures, the final ranking matrix can be established as depicted in Figure 4 – 7 (d).

Figure 4 – 7: Demonstration of matching strategy among call option and put option speculators

After the completion of ranking matrix, by repeating the propose-and-reject procedures as mentioned, it is capable of determining the most optimal matching. Suppose the procedures start from call option speculators, who propose their first preference to put option speculators; the result is likely to be $\{A \rightarrow \delta, B \rightarrow \delta, C \rightarrow \delta, D \rightarrow \delta\}$. With respect to the preference of put option speculator δ , as he is the only one receive proposes from others, it is likely to retain call option speculator C, due to its own highest preference, and reject the rest as it has the highest preference.

After that, all unassigned call option speculators propose to all option speculators in terms of their second preference; the result becomes $\{A \rightarrow \alpha, B \rightarrow \alpha, D \rightarrow \alpha\}$. In this case, put option speculator α retains call option speculator B, as this is the highest preference of those proposed. The same process repeats until all call option speculators have been assigned to corresponding put option speculators, which results in the combination of $\{A\beta, B\alpha, C\delta, D\gamma\}$, as shown in Figure 4 – 7 (d) and (e). By representing the matching result in terms of the reimbursement scenario, as depicted in Figure 4 – 7 (f), it is able to have the observation that the third entity stands a much higher chance to gain all insurance premiums collected from both call and put option speculators than to reimburse both parties. The developed business strategy of matching is analogous to the solution of stable marriage problem, except in our model, the preference can be quantified in terms of the moneyiness of financial options.

4.4 Option Insurance in Financial Market

The feasibility of option insurance can be addressed in terms of three aspects: market acceptance, option insurer profitability, and positive market effect.

4.4.1 Market acceptance

Option insurance integrates two distinct concepts of financial options and insurance. The contemporary financial market is a typical two-entity structure composing option writer and option holder. In this case, one additional entity, the third entity, is firstly introduced in the establishment of a three-entity structure. The third entity mitigates the investors' cost used in hedging or speculating.

As mentioned above, option is generally employed for the purpose of hedging or speculating. However, investors have either little or no control over the probability from a particular set of portfolio even with the involvement of options, due to the concerns of strike price and expiration date, just analogous to an individual, who purchases accident insurances have little or no control over probability of accident occurring to him.

Therefore, it makes economic sense that the market participant is capable of seeking out an avenue to reduce the risk or the premium invested in the financial options. The primal idea of the business is to spread the total risks between paired investors, and hence, the third entity should be pleasing to option market participants. They will be more than willing to purchase option insurance to insure against their potential losses due to "misfortunate bet" if an option premium is affordable in the view of a possible refund or possible gain.

In the contemporary financial market, the annual trading volume of financial derivatives amounts to trillions of dollars, and therefore, it is certainly a huge market to tap. In addition to that, no noteworthy evidences have been discovered to suggest that the option insurance with any rudimentary intention of an "insurance policy" to derivative market participants have been developed so far.

4.4.2 Option insurer profitability

Any risk, which lasts sufficiently long, may be assessed and insured. Such a phenomenon explains why the insurer's strategy is employed to financial derivatives instead of shares, while the valuation of stocks is highly unpredictable. In essence, option insurer's ability for reimbursement is to shift risk from individual investors to a group of investors by redistributing losses among insured option market participants.

Under the perfect market assumption, the ideal version of option insurances, speculating insurance policies and hedging insurance policies, seeks to identify a pair of option investors with entirely reversed reimbursement condition (one call option and one put option), identical strike price and the same expiration date. Such that, the third entity always holds a no-loss position and its profitability is perfectly independent of the market trend.

In real financial market, the perfect market rarely exists as there are always unequal-sized "camps" of calls and puts with different moneyness. This is mainly caused by different expectations about one particular financial asset and adopted trading strategies. When encountering the imperfect reality, the proposed business strategy, as described in section 4.3, makes it possible in the combination of different calls and puts with different moneyness, which effectively mitigates the potential risks faced by third entity and maximizes its exposure to the potential profitability.

In addition to that, the pricing structure (or yardstick) always allows the third entity make in-time adjustment with respect to market condition in order to stand in a profiting position. And the third entity can always act as a normal insurance company to invest its collected option premium into the fixed-income financial products, or take certain amount of risk in providing insurance for unequal-sized "camps".

4.4.3 Positive market effect

With the introduction of option insurance, it reduces the volatility of option market and stimulates the willingness of market participation. The reimbursement mainly stems from the premium paid to option insurers, such that, the cash flow between option insurer (third entity) and insurance buyer is merely independent of the prevailing financial derivative market.

The business of option insurer is seemingly an arbitrage but its profitability is much lower than a successful bet by an option speculator or even a portfolio manager. Therefore, the position shall, in certain degree, foreclose the deterioration of option insurance business.

The proposed business idea seeks to supplement prevalent market practices of option and insurance business, where the contemporary financial market does not have avenues for market participants to seek insurances to protect their option premiums against “misfortune” or “wrong bet”, especially for option speculators with an entirely unprotected position.

The central idea of option insurance is unique and distinctive in which it brings two starkly dissimilar concepts together, merges them in the creation of a completely new situation, whereas opportunities arise for a third entity to exist, and creates an overall win-win situation for all entities involved. It allows the redistribution of risks among a larger pool of investors and mitigates the loss of individuals. With the potential for lower risks and having a payoff that outweighs its investment, an investor is positively enticed to purchase the option insurance to protect against his potential losses.

4.5 Conclusion

The third entity seeks to replicate insurance practices into the field of risk management in finance, whereas risks are distributed among a larger pool of entities of speculators or portfolio managers. This would be made possible without a complete understanding in the process flow of how speculators and portfolio managers utilize options to speculate or hedge against adverse market movements and how insurance industries generally operate. It leads to the successful integration of two starkly-dissimilar concepts of insurance policies and financial derivatives.

It is a worthwhile strategy if the third entity is sufficiently large to have simultaneously speculators or portfolio managers taking opposite position to each other in terms of option purchase and the level of option moneyness. Even under the imperfect market conditions, the proposed business strategy of matching allows more entities to be protected by the option insurance, while minimizing the exposure of third entity to the potential losses.

In addition to that, the third entity issues two types of option insurances, one for speculators and another for portfolio managers, has a tremendous impact on contemporary financial option market. Firstly, it reduces the volatility of prevailing financial derivative market via spreading the total risk among various entities. Secondly, it simulates the willingness of market participation with the introduction of additional protection which does not require sophisticated calculations. Last but not least, the disclosure of option buyers, who seek for option insurance, always reflect a better sense of market expectations, which can even be treated as an effective index to monitor the overall performance of special financial asset.

Chapter 5

Literature Review of Biomolecular Computing

Once concerning about the word “computer”, our intuition automatically associate it with the images of monitors, keyboards, or many professional terms such as central processing unit (CPU), random-access memory (RAM) and read-only memory (ROM). It is because people have grown accustomed to the concept of emulating computation by using a specific device referred to as digital computer, which assembles individual functional electronic components on a silicon-based substrate, therefore, in certain cases, the digital computer is also called as silicon-based computer. Since the first digital computer was firstly introduced at the beginning of 1970s, improving its computational ability – processing speed, parallelism, and minimization – has always been the top concerns among researchers all over the world.

As predicted by Moore’s law, the size of individual transistor elements can always be reduced with the advance of time due to the newly available technology, which allows additional processing units to be packed onto the same chip. The increasing packing density always leads to problems, involving power consumption and heat dissipation. Moreover, the employment of silicon-based substrate as base material in manufacturing of digital computer always has negative health and environmental influence [33]. Despite that, the entire semiconductor industry is quickly approaching the physical constraints of Moore’s law.

In reality, any devices endowed with three fundamental functions – processing, storing, and displaying information – can be regarded as a computer. Therefore, researchers from various disciplines are engaged in exploring alternatives to electronic-based digital computer. Among various intriguing approaches, biomolecular computing seems to be the most feasible method to become the computer of next generation.

Biomolecular computing is a specific computing paradigm which utilizes the artificially synthesized nucleic acids or any other analogous structures as information storing and processing substrates to tackle computational problems.

The field of biomolecular computing was arguably initiated by Leonard M. Adleman [37], who had successfully demonstrated the solution of a non-deterministic polynomial (NP) problem – seven-vertex directed Hamiltonian Path Problem (HPP). The introduction of surface-based technique¹⁶ to biomolecular computing provides additional flexibility to the design of algorithms [53]. These subsequently developed biomolecular computing algorithms utilize the advantage of biomolecular computing, that is, its massive parallel computational ability. However, all these algorithms have one critical limitation as they are restricted in solving one specific class of combinatorial optimization problems.

In order to develop a general purpose biomolecular computer, analogous to contemporary digital computer, designing appropriate biomolecular logic circuits is believed to be a feasible starting point. The logic circuit is assembled by a series of individual functional units called logic gates, which is capable of performing Boolean logic operations. Most recently, the researches in the field of biomolecular computing has been further extended to the information storage by means of pure biomolecular method.

Prior to the research field of biomolecular computing, it is essential for researchers to overcome several technical barriers such as getting familiar with the concepts of fundamental molecular biology and theoretical computer science. After that, the ability of researchers to discover inter-connections between these two starkly-dissimilar concepts plays an important role. In addition to a complete literature review of biomolecular computing, this chapter is aiming to provide researchers from unrelated disciplines the essential knowledge prior to the field. This chapter can be broadly separated into three individual sections as listed below:

1. *Introductory molecular biology* (section 5.1)

The section is designated for researchers from non-biochemical background. It selectively includes the essential concepts of molecular biology, such as molecular structures, various definitions, and the most frequently used laboratory techniques.

¹⁶ **Surface-based technique:** one end of single-stranded DNA containing specifically design sequence is able to bind to the surface with the addition of appropriate attachment chemical.

2. *Theoretical computer science* (section 5.2)

In early developments of biomolecular computing, the method was applied to solve one specific class of problems, namely combinatorial optimization problems, which are generally associated with the graphical representations. Few researchers from a non-mathematical background can understand the concepts of graph theory. Therefore, these important terms and definition are selectively included in this section. In addition to that, the key concepts of Boolean logic are included in this section, which is likely to be used as the foundation of designing biomolecular logic circuit.

3. *Developments of biomolecular mediated computing* (section 5.3)

Unlike previous two sections, which serve as the background information of biomolecular computing for researchers from unrelated disciplines, this section is dedicated in summarizing the major breakthroughs of biomolecular computing. It involves the detailed analysis and discussion of several crucial papers throughout the years.

5.1 Introductory Molecular Biology

Understanding the essential concepts of biomolecular biology plays a crucial role prior to this cross-disciplinary research, biomolecular computing. The aiming of this section is to selectively include the fundamental concepts, which are very likely to be used in the field of biomolecular computing. It starts with a brief history of molecular biology (subsection 5.1.1), followed by the introduction of two types of chemical bonds (subsection 5.1.2), and the structure of DNA and RNA (subsection 5.1.3). Converting from 2-dimensional DNA to 3-dimensional DNA always lead to additional concerns, these concerns are included in DNA topology (subsection 5.1.4). The most frequently used laboratory techniques are described in laboratory techniques (subsection 5.1.5).

5.1.1 A brief history of molecular biology

About 150 years ago, Charles Darwin and Alfred R. Wallace proposed the evolution theory, also known as Darwinian Theory, based upon the natural selection of most fit. The Darwinian Theory predicts all the lives existing in Earth today are coming from a simple form of bacteria in more than 4 million years ago.

In 1865, Gregor Mendel, an Austrian scientist, proposed the concept that gene is a discrete entity in his paper “experiments in plant hybridization”. Unfortunately, his work did not draw any attentions at that time. His theory was later confirmed by three plant breeders 16 years after his death.

The evidence, which indirectly indicate the deoxyribonucleic acid (DNA), may be genetic material, was firstly involved in the experiment conducted by English biologist Frederick Griffith in 1928, which is commonly known as the transforming principle.

In 1953, based upon the high quality X-ray diffraction photograph provided by Rosalind Franklin, also known as photo 51¹⁷, the actual structure of DNA, a double helical structure consists of two anti-parallel, complementary polynucleotide chains, was discovered by Francis Crick and James D. Watson. Understanding the actual structure of DNA is believed to be the foundation of

¹⁷ **Photo 51:** a series of photo taken by Rosalind Franklin. The No. 51 is the photo with the best resolution.

biomolecular computing. In addition to that, the self-complementary nature of DNA immediately suggests its hypothetical replication mechanism, that is, each polynucleotide chain served as the template in the formation of its complementary polynucleotide chain.

The hypothetical replication mechanism was later proved by two major breakthroughs: (1) U.S. biochemist A. Kornberg's discovery of DNA polymerase I, which is capable of catalyzing the synthesis of new DNA strand based on the existing template and energy-rich precursors (dATP, dGTP, dCTP, and dTTP); (2) the replication of DNA is a semiconservative process, which was proved by M. Meselson and F. W. Stahl in 1958.

The genetic information flow from deoxyribonucleic acid (DNA) to amino acids (protein) was postulated by Francis Crick in 1956, which is now known as central dogma. . In 1961, Brenner and Francis Crick discovered that a group of three nucleotides (three codons) are exploited to specify individual functional amino acids.

In summary, molecular biology is the study of properties of genetic material and various molecular machines that govern the information flow from nucleic acids to proteins. In biomolecular computing, it is necessary to understand the properties of genetic material, various molecular machines, as well as frequently used laboratory techniques. However, the concept on how genetic information from nucleic acids to functional proteins are not of the top concerns.

5.1.2 Two types of chemical bonds

Before moving into the detailed explanation of structures of nucleic acids, deoxyribonucleic acid (DNA) and ribonucleic acid (RNA), it is necessary to understand two types of chemical bonds in molecular biology as any complicated molecular structures are started with these simple chemical bonds.

Chemical bonds, according to the corresponding strength and geometrical proximity, can be broadly classified into two categories, known as weak bonds and strong bonds. These two types of chemical bonds are of equivalent importance in molecular biology as well as biomolecular computing. The building block of nucleic acid is nucleotide. The individual nucleotides are linked together by covalent bonds, which is also referred to as strong bonds, leading to the formation of polynucleotide chains. Weak bonds are also playing a crucial role in molecular biology. It mediates interactions among different parts of the macromolecules.

Characteristics of chemical bonds

By definition, chemical bond is an attractive force that holds atoms all together. It is characterized by four key parameters – bond strength, valence, bond angle, and the freedom of rotation.

The bond strength of chemical bonds between any two atoms is directly reflected by the physical distance between these two atoms. The strength of the bonds is inversely proportional to the corresponding distance between two atoms. In addition to that, as the distance between any two atoms is usually extremely small. Therefore, the unit of length, which is frequently used in molecular biology, is angstrom, denoted by Å, in which $1 \text{ Å} = 1 \times 10^{-10} \text{ m}$. Valence is another important characteristic, which is defined as the maximal number of covalent bonds, can be formed for one specific atom.

The angle between two bonds originated from a single atom is defined as bond angle. By simply referring to the bond angle alone, it is possible to differentiate whether the bond between two atoms is weak or strong. The bond angles of any two covalent bonds, or strong bonds, are nearly identical. On the contrary, the bond angles of any two weak bonds, for instance, hydrogen bonds, may be completely different. Bonds are also different in the corresponding freedom of rotation, that is, the weak bonds permits free rotation of bonded atoms, while the strong bonds rarely allows bonded atoms to rotate, *i.e.*, less freedom of rotation.

The formation of all chemical bonds, no matter weak bonds or strong bonds, follows the first law of thermodynamics. The process of two initially separate atoms, let's say, atom A and B, in the formation of the bonded aggregate AB can be described by equation 5.1.

Reversely, the chemical bonds can also be broken by introducing energy into the aggregate AB, which is described by equation 5.2.

In molecular biology, the resulting solution is a direct result of a series combined actions of bond-making and bond-breaking. Once the solution is reached in equilibrium state in a closed system, the proportion of bonded atoms can be described by equation 5.3.

$$K_{eq} = \frac{conc^{AB}}{conc^A \times conc^B} \quad (5.3)$$

Where the term K_{eq} is the constant of equilibrium, $conc^A$, $conc^B$, and $conc^{AB}$ are the concentrations of atom A, B, and AB, respectively.

The unit of concentration is in mole/liter. If the value of equilibrium constant K_{eq} is large, it simply indicates the formation of AB is effectively irreversible, *i.e.*, the solution favors bond-making instead of bond-breaking, which is reflected by comparably high concentration of AB. On the contrary, if the value of equilibrium constant K_{eq} is relatively small, it promotes bond-breaking instead of bond-making.

Gibbs free energy

In molecular biology, the Gibbs free energy, denoted by symbol G, is the chemical potential, where proportion of bonded atoms move toward the equilibrium concentration. In the perspective of biomolecular computing, Gibbs free energy is the energy that has ability to do work.

The equation of Gibbs free energy in an isolated system can be derived as follows. Firstly, according to the second law of thermodynamic, the change of total entropy in a completely isolated system is always greater than zero, as expressed in equation 5.4.

$$\Delta S_{total} \geq 0 \quad (5.4)$$

The total entropy of the system consists of two parts – internal entropy, denoted by ΔS_{ini} and external entropy, denoted by ΔS_{ext} , that is, $\Delta S_{total} = \Delta S_{ini} + \Delta S_{ext}$. The external entropy is used to describe the closed system thermally connected to its surrounding. By convection, if +Q is the heat transferred to system from surrounding to the system, then -Q is subjected to the heat dissipated from system to its surrounding. Based on that, equation 5.5 can be established.

$$\Delta S_{ini} - \frac{Q}{T} \geq 0 \quad (5.5)$$

It is possible to rearrange equation 5.5 by multiplying both sides of equation 5.5 by T . After that, assuming the process is acrobatic, *i.e.*, the process is under constant pressure, the heat transfer can be expressed in terms of the difference in enthalpy, *i.e.*, $Q = \Delta H$. After that, the resulting equation is converted to equation 5.6.

$$\Delta H - T\Delta S_{ini} \leq 0 \quad (5.6)$$

In molecular biology, entropy is a measure of the amount of disorder. In other words, the amount of disorder is directly proportional to the amount of entropy in a closed system. The change in Gibbs free energy can be defined as the left-hand-side of equation 5.6, *i.e.*, $\Delta G = \Delta H - T\Delta S_{ini}$. Therefore, according to the second law of thermodynamics, it is possible to obtain the following relationship: (1) if $\Delta S_{total} > 0$, or $\Delta G < 0$, the chemical reaction is favored; (2) if $\Delta S_{total} = 0$, or $\Delta G = 0$, the chemical reaction reaches the equilibrium state; (3) if $\Delta S_{total} < 0$, or $\Delta G > 0$, the chemical reaction is disfavored as it disobey the second law of thermodynamics.

There is a relationship between equilibrium constant K_{eq} and the change of Gibb's free energy ΔG , which can be expressed in terms of equation 5.7.

$$\Delta G = -RT\ln K_{eq} \text{ or } K_{eq} = e^{-\frac{\Delta G}{RT}} \quad (5.7)$$

Recall that the equilibrium constant K_{eq} is proportional to the concentration of bonded aggregate as described in equation 5.3. Therefore, the large value of K_{eq} implies the speed of bond-making is much faster than that of bond-breaking. On the other hand, it reveals the method of determining whether a chemical bond is a strong bond or weak bond can simply referring to the value of ΔG . For covalent bond, the chemically strong and therefore stable bond, it is always associated with a large energy reduction during the formation of the chemical bond, *i.e.*, a large negative value of ΔG . On the contrary, the formation of weak bonds atoms is usually associated with a small negative value of ΔG (only several times larger than the average thermal energy of molecules). Therefore, weak bonds play an important role in the living cells due to the high frequency of made and broken.

Weak bonds

In molecular biology, there are four types of most frequently associated weak bonds: hydrogen bond, van der Waals bond, hydrophobic bond, and ionic bond. Among these weak bonds, hydrogen bond and van der Waals bond are playing a crucial rule in biomolecular computing.

Based on the chargeability of molecules, the molecules can be classified as polar molecules and non-polar molecules. The polar molecules are capable of forming hydrogen bonds. Water, or H_2O , is the most important polar molecule, which allows other polar molecules to be soluble in water, *i.e.*, polar molecules are able to form hydrogen bonds with water molecules. The non-polar molecules, on the other hand, can only form van der Waals bonds, and therefore, they are insoluble in water. As compared to van der Waals bond, the hydrogen bond is comparably stronger.

Although two atoms bounded together by single or double weak bonds are easily to fall apart, *i.e.*, highly unstable, a collection of weak bonds is still able to result in a stable aggregate. This explains the reason why double helical structure of DNA is stable under general physiological condition, and therefore, nucleic acids can be utilized as information storage medium in biomolecular computing.

Strong bonds

As specified by second law of thermodynamics, the spontaneous biosynthetic reactions always involve a decrease in Gibbs free energy, *i.e.*, $\Delta G < 0$. Such that, many biosynthetic processes are mostly achieved as a result of so-called “coupled” reactions, *i.e.*, the precursors are firstly converted to high-energy activated molecules, which is acting as the energy suppliers to promote subsequent reactions.

In most biosynthetic processes, the primary energy source in living cells is adenosine triphosphates, denoted by ATP. It is obtained as a result of interaction among adenosine diphosphates, denoted by ADP, inorganic phosphate, and additional energy supplied during the degradative reactions or photosynthesis. Such interaction can be described by using equation 5.8. The symbol $\sim$ is used to represent the high energy bond.

The high-energy molecule contains one or more high-energy bonds are likely to breakdown in water. The entire process is called hydrolysis process, which is accompanied by a large decrease in free energy. ATP is the most frequently encountered high-energy molecule in molecular biology, which contains several high energy bonds. Its hydrolysis creates a large negative ΔG , as specified by equation 5.9.

The chemical groups linked by high-energy bonds are called high-energy groups, for instance, Adenosine – P~P~P in equation 5.9 a. High-energy groups can be transferred to other molecules by a process called group transfer reactions leading to the creation of new high-energy compounds. These derivatives are acting as the intermediate precursor for many subsequent biosynthetic procedures. In the following equation, equation 5.10, demonstrates guanosine monophosphate, which is denoted by GMP, is activated by high energy group ATP in the formation of new high-energy group guanosine triphosphate GTP.

Both types of nucleic acid, DNA and RNA, are built up from mononucleotide monomers, also called nucleoside phosphate. The immediate precursors for both DNA and RNA are the nucleoside-5'-triphosphates. For DNA, these precursors are dATP, dGTP, dCTP, and dTTP (“d” stand for deoxy-). For RNA, these corresponding precursors are ATP, GTP, CTP, and TTP.

5.1.3 Chemical structure of DNA and RNA

Nucleotide

The best way to study a complicated chemical structure is decompose its overall structure into simplest building blocks. The fundamental building block of deoxyribonucleic acid (DNA) is nucleotide.

Each nucleotide consists of two compounds – nucleoside and a phosphate group. Furthermore, each nucleoside comprises a sugar group (formally known as 2'-deoxyribose) and a nitrogenous base (base).

The core of a nucleotide is the 2'-deoxyribose, as shown in Figure 5 – 1, which connects base and phosphate group.

Figure 5 – 1: Chemical structure of 2'-deoxyribose

The central chemical component of a 2'-deoxyribose is five-carbon structure. These carbons can be differentiated by simply numbering these carbon elements with respect to the attached chemical

components, as depicted in Figure 5 – 1. The nitrogenous base is attached to 1' carbon. The phosphate group is attached to 3' carbon of one nucleotide and 5' carbon of another nucleotide in the formation of phosphodiester bond. The phosphodiester bond, a typical covalent bond, is the key to assemble individual nucleotides into a complete polynucleotide chain. In addition to that, the absence of hydroxyl group (OH) in 2' carbon (this is the reason why it is called 2'-deoxyribose), is one of the features that distinguish DNA from RNA.

The formation of nucleotide, as shown in Figure 5 – 2, by joining nitrogenous base and phosphate group to 2'-deoxyribose can be hypothetically treated as a double-dehydration process, although it is not realistically achieved in nature.

Figure 5 – 2: Formation of nucleotide by a hypothetical double dehydration process

The only difference among nucleotides is the nitrogenous base attached to 1' carbon. There are four variations of nitrogenous bases. These bases can be classified in two classes, purines and pyrimidine, based on their corresponding chemical structure. The purines are double-ringed

structure, a five-membered together with a six-membered ring. The pyrimidines are single-ringed structure, a six-membered ring. The purines are adenine (A) and guanine (G); the pyrimidines are cytosine (C) and thymine (T). The detailed structures of purines and pyrimidines are described in Figure 5 – 3.

Figure 5 – 3: *Purine and pyrimidine*

Single-stranded DNA (ssDNA)

Individual nucleotides are joined to each other in the formation of polynucleotide chains by combining 3' – hydroxyl (OH) of one nucleotide and phosphate (P) of another nucleotide. In biomolecular computing, a single polynucleotide chain in the solution is frequently referring to as single-stranded DNA (ssDNA), as described in Figure 5 – 4.

Two ends of a single polynucleotide chain, or single-stranded DNA, are manually defined as 5' end and 3' end, respectively. The end with exposed 5'-phosphate group, as indicated in left-hand side of Figure 5 – 4, is defined as 5' end. Analogously, the end with exposed 3'-hydroxyl group, as indicated in right-hand side of Figure 5 – 4, is defined as 3' end. The size of ssDNA is measured by counting the number of nucleotides, and the corresponding unit for single-stranded DNA is “mer”. The molecule, as indicated in Figure 5 – 4, is a typical 4 mer ssDNA.

In biomolecular computing, for the sake of convenience, the single-stranded DNA (ssDNA) is frequently expressed in terms of an arrow, whereas, by convention the arrow head is the 3' end; the other end is 5' end. In addition to that, the same ssDNA can also be expressed in terms of a string representation: 5' – ACGT – 3'.

Figure 5 – 4: *Chemical structure of a single polynucleotide chain*

Watson-Crick complementary principle

Two complementary polynucleotide chains, or single-stranded DNA (ssDNA) in biomolecular computing, are able to form double-stranded DNA (dsDNA) through hydrogen bonding. The word “complementary” is referring to the Watson-Crick complementary principle. As specified by complementary principle, purine adenine (A) is always paired with pyrimidine thymine (T) in the formation of two hydrogen bonds; purine guanine (G) is always paired with pyrimidine cytosine (C) in the formation of three hydrogen bonds, as shown in Figure 5 – 5. In addition to that, each of the bases exists in two alternative tautomeric states. The Watson-Crick base pairing principle is valid only if both bases are in their corresponding predominant states.

Another important feature of base pairs is that A : T and G : C base pairs have exactly the same geometry. There is always one purine (double-ringed structure), which pairs with one pyrimidine (single-ringed structure). Based on this specific characteristic, all four bases can be

accommodated within the same arrangement in the double helical structure of DNA without introducing any possible geometrical distortions.

Figure 5 – 5: *The Watson-Crick base pairing*

The specific geometrical arrangement of base pairs introduces one additional feature to the double helical structure, which is known as minor groove and major groove. The angle between two sugars protrudes from the base pairs, *i.e.*, the angle between two glycosidic bonds is either approximately 120° (narrow angle) or 240° (wide angle). Imagine the individual bases pairs stack on top each other, the narrow angle between glycosidic bonds on one edge of base pairs generates a minor groove and the large angle on the other edge generates a major groove. The major groove and minor groove in B-form DNA are illustrated in Figure 5 – 6.

The major groove of B-form DNA plays a crucial role in various processes of molecular biology. The most dominant feature is that the major groove is rich in chemical information. This reveals the reason why the chemical information in major groove is frequently referred to as the “second genetic code”. The second genetic code consists of four letters – A, D, H, and M. The letter A stands for the hydrogen-bond acceptor, D is a hydrogen-bond donor, H is the non-polar hydrogen, and M is the methyl group (CH_3), which can be observed in Figure 5 – 7.

Based on the chemical groups exposed in the major and minor grooves from the edges of the base pairs, also termed as “second genetic code” (the first genetic code is adenine, thymine, guanine, and cytosine), it is capable of distinguishing all possible base pairs. By listing the second genetic code in Table 3 – 1, several features can be observed. Firstly, by simply referring to the

exposed chemical groups in the major groove of base pairs, it is possible to differentiate various base pairs – A:T, T:A, G:C, C:G. Secondly, the second genetic codes are symmetric in the minor groove of various base pairs.

Figure 5 – 6: *Minor and major groove in B-form DNA*

	Major groove							Minor groove				
G	A	A	D	H	C		G	A	D	A	C	
A	A	D	A	M	T		A	A	H	A	T	
C	H	D	A	A	G		C	A	D	A	G	
T	M	A	D	A	A		T	A	H	A	A	

Table 5 – 1: *Second genetic code in major and minor groove of base pairs*

In addition to the ample chemical information in the major groove, it is also comparably large geometrical space, which allows protein, *i.e.*, functional polymerases are protein, to stick its amino side chains into the major grooves of DNA to detect and subsequently modify the chemical information.

Figure 5 – 7: *Exposed chemical groups in major and minor grooves of A:T and G:C pairs*

Double-stranded DNA (dsDNA)

Due to the geometry of base pairs, that is, the reverse orientations of the sugar group on the opposite strands, two strands of double helical structure of DNA have an anti-parallel orientation, as shown in Figure 5 – 8. In other words, the polarity of the sugar phosphate backbones of double helical structure of DNA is antiparallel, with one strand running from 3' end to 5' end and the complementary strand running from 5' end to 3' end. Analogous to single-stranded DNA (ssDNA), the size of double-stranded DNA (dsDNA) is measured by counting the number of base pairs, and therefore the corresponding unit for dsDNA is “bp”. Figure 3 – 8 shows a typical 12 bp dsDNA.

Figure 5 – 8: *Anti-parallel orientation of double-stranded DNA*

Under physiological condition, two polynucleotide strands are wrapped around each other to form a right-handed, double helical structure. Such conformation of double helical structure is B form, which was initially discovered by Watson and Crick in 1953. In addition to B form DNA, there are two additional conformation of double helical structure of DNA, known as A form DNA and Z form DNA.

The A form DNA can be observed under condition of low humidity. The physiological significance of Z form DNA still remains uncertain in molecular biology. However, in biomolecular computing, it is always dealing with B form DNA. For B form DNA, each base pair in the double helix is displaced from the base pair beneath it by approximately 34° , therefore, there are approximately 10.5 base pairs per turn of the helix.

The stability of double helical structure of DNA is proportional to the number of hydrogen bonding between two complementary strands, base stacking, and the increment of entropy raised from the displacement of water molecules. In the perspective of biological structure, G:C pairs contribute more stability in the formation of double helical structure as compared to A:T pairs due to one additional hydrogen bond.

DNA denaturation, renaturation, and hybridization

The most commonly used process to convert from double-stranded DNA (dsDNA) to single-stranded DNA (ssDNA) is referred to as DNA denaturation. The DNA denaturation process can be achieved by either heating the solution containing dsDNA above its physiological temperatures or increasing the pH value of solution. Reversely, once the heated solution of denatured DNA is

slowly cooled, individual DNA strands are likely to search for their complementary strands, according to Watson Crick complementary principle, reform the original double helical structure. Such reverse process is termed as renaturation.

DNA denaturation can be monitored by simply measuring the absorbance of ultraviolet light pass through the DNA solution. Under physiological condition, DNA in solution absorbs ultraviolet light at a wavelength of approximately 260 nm. However, if the solution is heated above its physiological temperatures, the corresponding ability of absorbing the ultraviolet light is increased by roughly 40%. This phenomenon is known as hyperchromicity.

One specified parameter of temperature is associated with the denaturation process: melting temperature, denoted by T_m . Melting temperature is defined as the temperature when half of double-stranded DNA has dissociated. Melting temperature is mainly affected by the percentage of G:C pairs of double helical structure and the ionic strength of solution, which is also known as salt concentration.

The widely accepted method, which is used to estimate the melting temperature of double-stranded DNA (dsDNA), is the nearest neighbor method proposed by John S. L. Jr. [83 – 85]. The proposed formula in estimating melting temperature T_m is equation 5.11.

$$T_m = \frac{\Delta H}{\Delta S + R \ln \left(\frac{C_T}{N} \right)} \quad (5.11)$$

Where T_m is the melting temperature in Kelvin (K). ΔH and ΔS are representing standard enthalpy and standard entropy, respectively.

The value of ΔH and ΔS of different nearest neighbor sequence can be obtained from Table 5 – 2. C_T is the total concentration of DNA strand. R is the gas constant which equals to 1.987 cal/(mol · K). N is the arbitrary constant.

The process of binding of two single-stranded DNA (ssDNA) is called hybridization, or annealing. In molecular biology, hybridization and annealing have totally distinct meanings: for hybridization, two single-stranded nucleic acids are coming from two different sources, for instance, complementary strands of artificially synthesized DNA and RNA; for annealing, two single-stranded nucleic acid should come from the same source. In biomolecular computing, these two words, annealing and hybridization, are interchangeable.

Sequence	ΔH (Kcal/mol)	ΔS (Kcal/mol · K)
AA/TT	-8.3 ± 1.1	-23.2 ± 3.1
AT/TA	-8.5 ± 0.8	-24.6 ± 2.4
AC/TG	-9.5 ± 1.1	-26.0 ± 2.7
AG/TC	-8.2 ± 0.9	-22.3 ± 2.7
TA/AT	-6.5 ± 0.6	-19.0 ± 1.7
TC/AG	-8.7 ± 0.8	-23.6 ± 2.4
TG/AC	-8.9 ± 0.4	-24.3 ± 1.3
CC/GG	-8.8 ± 0.6	-21.7 ± 1.7
CG/GC	-11.4 ± 0.9	-29.7 ± 2.5
GC/CG	-10.1 ± 1.0	-25.4 ± 2.6
Terminating A/T	4.4 ± 1.4	10.7 ± 3.9
Terminating G/C	1.7 ± 0.5	2.4 ± 1.3
Symmetry	0	-1.4

Table 5 – 2: *Nearest neighbor thermodynamic parameters for Watson-Crick base pairs*

Structure of RNA

The chemical structure of ribonucleic acid (RNA) has three distinct features as compared to deoxyribonucleic acid (DNA):

1. Backbone of RNA contains ribose instead of 2'-deoxyribose, *i.e.*, there is a hydroxyl group (OH) at 2' carbon, as shown in Figure 5 – 9.
2. RNA contains uracil (U) instead of thymine (T). Uracil (U) also has similar single-ringed structure except there is a 5' methyl group ($5' - CH_3$) is transcribed into a hydrogen group ($5' - H$), as indicated in Figure 5 – 9.
3. Under physiological condition, RNA is generally discovered as single polynucleotide chain in solution, *i.e.*, the complementary strand of single-stranded RNA does not generally exist.

Figure 5 – 9: *Structural differences between DNA and RNA*

Besides chemical structural differences between DNA and RNA, the function of RNA is also different from DNA. RNA is not capable of acting as genetic material as it is not able to replicate itself. In summary, RNA is acting as:

1. The intermediate between DNA and protein-synthesizing machinery. This part of RNA is generally referring to as messenger RNA (mRNA).
2. The adaptor between three-letter codons in mRNA and amino acids. This part of RNA is called transfer RNA (tRNA).

Under physiological condition, most RNA molecules, unlike DNA, which is able to form stable double helical structure, are likely to form stem loop structure, *i.e.*, if two stretches of complementary sequences of single-stranded RNA are close to each other, they are likely to form regional base pairing structures, such as hairpin, bulge, and loop, as shown in Figure 5 – 10.

Figure 5 – 10: *Various stem loop structure of RNA*

One possible reason RNA has property of forming regional stem loop structures is that besides the Watson-Crick base pairs, G:U base pair is also allowed in RNA in the formation of two hydrogen bonds, as depicted in Figure 5 – 11. This also explains the reason why RNA is mostly in single-stranded form in nature.

Figure 5 – 11: *The G:U base pair in RNA structure*

Under physiological condition, the three-dimensional structure of RNA is also different from that of DNA. Recall that, under physiological condition, DNA resembles B-form double helical structure. For RNA, due to the presence of 2'-hydroxyl ($2' - OH$) in the sugar phosphate backbone, the only possible structure that RNA is likely to assemble is A-form helical structure.

Ribozyme

Enzymes are frequently used in biomolecular computing. For a typical enzyme, it must be able to perform all of the following four functions: (1) bind to the substance through either hydrogen bonding, electrostatic, or van der Waals interactions; (2) carry out one specific chemical reaction; (3) release a typical product; and (4) repeat this series of events for many times.

Prior to 1980s, protein was believed to be the only possible substrate for creating enzyme. In 1980s, RNAs were proved to be the feasible biological catalysts, as they are able to (1) adopt complex tertiary structures which allow them to bind to and subsequently release from the target substance; (2) contain active site for the catalysis; (3) associate with small molecule cofactor.

Such RNAs are always referring to as the ribozymes, and they exhibit the characteristics of a classic enzyme. One of the discovered ribozyme is RNase P, which is made of two parts – protein part and RNA part. The RNA portion is acting as the catalyst. The protein portion facilitates the reaction by shielding the negative charges on RNA such that it can bind to target substrate.

5.1.4 DNA topology

In solution, not all the DNA molecules are in linear form under physiological condition. In certain circumstances, DNA may also in circular form, whereas two ends of DNA are covalently linked. Such DNA is frequently referred to as covalently-closed circular DNA, abbreviated cccDNA.

If there are no chemical interruptions to sugar-phosphate backbones of two strands of DNA, the absolute number of times the chains can twist around each other is a constant number, *i.e.*, linking number is an invariant property for cccDNA. In other words, a cccDNA is said to be topologically constrained. In reality, the topologically constrained property of cccDNA is not only applied to

cccDNA but also to the extremely long linear DNA in chromosomes. In addition to that, the topologically constrained DNA molecules play a crucial role in numerous dynamic processes including DNA replication and transcription. Therefore, an understanding of DNA topology and feasible methods to utilize topological constraints is of fundamental importance in biomolecular computing.

Linking number

Linking number, denoted by Lk , is defined as the number of times one strand has to pass through the other stand in order for two strands to be separated from each other. The linking number is an integer number, which is an invariant topological property of cccDNA without any chemical interruptions in its sugar-phosphate backbones.

The linking number is the sum of two geometrical components called the twist, denoted by Tw and writhe, denoted by Wr , which can be described by equation 5.12.

$$Lk = Tw + Wr \quad (5.12)$$

Twist, denoted by Tw , is the number of helical turns of one strand surrounds the other strand, *i.e.*, the number of times one strand completely wraps around its complementary strand. The twist can be determined by simply counting the number of times two strands cross each other. Writhe, denoted by Wr , is the number of long axis of the double helical structure crosses itself repeatedly in a three dimensional space. The relationship of these definitions, including linking number, twist and writhe are demonstrated by using a cccDNA of length of 378 base pairs in Figure 5 – 12. In Figure 5 – 12 (a), it is a relaxed cccDNA of link number of 36, twist of 36, and writhe of 0. By adding gyrase¹⁸ to relaxed cccDNA, it is possible to turn relaxed cccDNA to negatively supercoiled DNA, as shown in Figure 5 – 12 (b). Reversely, it is possible to add topoisomerase¹⁹ to relax cccDNA by increasing its linking number.

¹⁸ **Gyrase:** certain bacteria have a special type II enzyme that introduces negative supercoils.

¹⁹ **Topoisomerase:** a frequently used enzyme to relax supercoiled cccDNA by increasing the linking number (Lk).

Figure 5 – 12: *Topological states of covalently closed, circular DNA (cccDNA)*

Writhe, denoted by Wr , has two different forms: interwound (or plectonemic) writhe and toroid (or spirals) writhe. The interwound writhe is referring to the long axis of double-stranded DNA is twisted around itself. The toroid writhe is referring to the long axis is wound in a cylindrical manner, which is how DNA is packed into the chromosome.

Interwound writhe and spiral writhe are topologically equivalent to each other and are readily interconvertible as described in Figure 5 – 12 (b) and (c). In summary, the linking number is an invariant property of cccDNA, if any change is made to the twist number, writhe number is changed accordingly, and vice versa.

Linking number of fully relaxed cccDNA

Consider a typical covalently close, circular DNA, abbreviated by cccDNA, that is free of supercoiling (over or under helical twist) and its twist is identical to that of B form of DNA in

solution under physiological conditions (approximately 10.5 bp per turn of double helical structure).

Linking number (Lk) of fully relaxed cccDNA under physiological condition is assigned with the symbol Lk^0 . The linking number of fully relaxed cccDNA can be calculated by using equation 5.13.

$$Lk^0 = \frac{\text{total number of bps}}{10.5} \quad (5.13)$$

One additional thing worth to mention is that the linking number (Lk) of one specific cccDNA is always an integer. However, the fully relaxed linking number for cccDNA (Lk^0) need not to be an integer number.

Supercoiling of DNA

The extension of supercoiling is measured by the difference between Lk and Lk^0 , which is termed as linking difference, denoted by symbol ΔLk . The relationship among linking difference, linking number and linking number of fully relaxed cccDNA can be described by using equation 5.14.

$$\Delta Lk = Lk - Lk^0 \quad (5.14)$$

If linking difference equals to zero, *i.e.*, $Lk = Lk^0$, cccDNA is fully relaxed. If linking difference is not equal to zero, *i.e.*, $Lk \neq Lk^0$, cccDNA is said to be supercoiled. In this case, if linking difference $\Delta Lk > 0$, cccDNA is positively supercoiled; if linking difference $\Delta Lk < 0$, cccDNA is negatively supercoiled.

As all linking number parameters, Lk, Lk^0 , and ΔLk , are dependent on the length of DNA molecules, it is convenient to refer to a normalized measurement of supercoiling, which is known as the super-helical density, denoted by symbol σ , which can be defined by equation 5.15.

$$\sigma = \frac{\Delta Lk}{Lk^0} = \frac{Lk - Lk^0}{Lk^0} \quad (5.15)$$

Both negative and positive supercoiling can be regarded as a store of free energy. The only difference between these two types of free energy is reflected by their significance in the subsequent procedures. The negative supercoiling, *i.e.*, $Lk < Lk^0$, can be treated as the store of free energy that facilitates DNA strands separation in replication, transcription, and certain kinds of recombination process. The negative supercoiling of DNA promotes strand separation, which can be observed by Figure 5 – 12. Suppose the initial covalently closed, circular DNA (cccDNA) is relaxed as beginning, as shown in Figure 5 – 12 (a). The negative supercoiling of cccDNA is achieved by adding specific enzyme into the solution leads to the introduction of negative writhe of $Wr = -4$, as indicated in Figure 5 – 12 (b). Recall that the twist and writhe are interconvertible. Therefore, the double helical structure is converted to that as described in Figure 5 – 12 (c). Recall that the linking number (Lk), by definition, is the number of times one strand have to pass through the other strand in order for two strands to be entirely separated from each other. The resulting strand as shown in Figure 5 – 12 (c), one strand has to pass through the other complementary strand for another 32 times prior to the complete separation. Therefore, it explains the reason why DNA in most organisms is negatively supercoiled.

5.1.5 Laboratory technique

In biomolecular computing, one or a series of laboratory techniques are applied to achieve specific purpose as designated by biomolecular computing algorithm. The most frequently adopted laboratory techniques are summarized in this subsection including gel electrophoresis, polymerase chain reaction, restriction endonuclease process, ligation, and sequencing.

Gel electrophoresis

Gel electrophoresis is probably the most frequently used laboratory techniques in biomolecular computing to separate, sieve, and purify DNA fragments with respect to its size. In addition to that, it is relatively simple, rapid to perform, and capable of resolving fragments of DNA that cannot be easily separated by other methods.

A standard laboratory setup of gel electrophoresis generally consists of three essential components: gel slab, power source, and electrophoresis buffer. It is schematically described in Figure 5 – 13.

As indicated in Figure 5 – 13, a series of sample slots, or wells, are created near the cathode end during the formation of gel slab. The wells are used to retain various DNA solutions. The entire gel slab is immersed in specifically designed electrophoresis buffer. Once the solution is loaded into the wells, the electrical field is activated. In nature, DNA fragments are negatively charged. Therefore, once the electric field is activated, the DNA solution in sample slots migrates towards the anode end through the porous material, the gel slab. Due to the different sizes of DNA fragments, the experienced frictional forces from gel matrix are entirely different. Shorter linear DNA molecules suffer less frictional force, and hence have better migration ability when passes through the pores of gel matrix. On the other hand, longer linear DNA molecules experience more frictional force, its mobility is therefore limited. As a consequence of electrophoresis, the shorter linear DNA fragments are closer to the anode end as compared with longer linear DNA fragments. Therefore, it is able to conclude that gel electrophoresis is capable of distributing the linear DNA molecules according to its length. The migration distance of corresponding DNA fragments is inversely proportional to the length of linear DNA molecules.

Two types of materials are frequently selected as the sieving material, known as polyacrylamide gel and agarose gel. Choosing which type of gel is used is primarily depending on the size of fragments and the resulting resolution.

Polyacrylamide gel is effective for separating small DNA fragments of length ranging from 5 bp to 500 bp. The resulting resolution is extremely high: DNA fragments of 1 bp difference can be clearly observed in polyacrylamide gel. The laboratory configuration for polyacrylamide gel is slightly different as compared with that for agarose gel, that is, it is subjected to a vertical configuration²⁰. Polyacrylamide gel also has disadvantage over agarose gel. The preparation and handling of polyacrylamide is much more difficult as compared to that of agarose gel.

²⁰ **Vertical configuration:** The gel slab is placed vertically with sample slots on top of the gel slab.

Figure 5 – 13: *A schematic representation of standard gel electrophoresis configuration*

Agarose gel has lower resolution than polyacrylamide gel. However, linear DNA fragments of length of 50 bp to 20,000 bp can probably resolved in agarose gels in a horizontal configuration. The limit of resolution is reached when the radius of gyration²¹ of linear duplex exceeds the pore size of gel matrix. The pore size of gel matrix can be controlled by adjusting the concentration of agarose, *i.e.*, the pore size of gel matrix is inversely proportional to the concentration of agarose. The casted agarose gel with low concentration of agarose (0.5%) is capable of resolving extremely long linear DNA fragments of length up to 25,000 bp. However, agarose gel at this concentration is extremely fragile and the entire electrophoresis may take several days. Agarose gel also has its physical limitation. It is not able to resolve linear DNA molecules of more than 750,000 bp in length once it is subjected to a constant electric field. The linear DNA molecules of more than 750,000 bp can be resolved by a method called pulsed-field gel electrophoresis, abbreviated PFGE. Nowadays, the resolution of linear DNA molecules larger than 6,000 kbp can be achieved.

²¹ **Gyration:** a single complete turn either in axial or orbital.

In addition to the sieving material, the electrophoretic mobility of DNA is also affected by composition and ionic strength of electrophoresis buffer. Without the involvement of ion, *i.e.*, using water instead of electrophoresis buffer, as a consequence, the electrical conductivity is extremely small; the DNA fragments migrate slowly through gel matrix. Select appropriate buffer at right concentration plays an important role. There are several buffers available for electrophoresis of native, double-stranded DNA, including TAE buffer, TBE buffer, and TPE buffer. TAE buffer is a buffer solution containing a mixture of Tris base, acetic acid, and EDTA. It is frequently employed in agarose gel electrophoresis typically for the separation of nucleic acids, including DNA and RNA, which sequesters divalent cations. TBE buffer is a buffer solution containing a mixture of Tris base, boric acid, and EDTA. The application of TBE buffer is similar to TAE buffer. The difference between TAE buffer and TBE buffer is:

- TAE buffer is ideal for the electrophoresis of large DNA fragments of length greater than 20,000 bp. TBE buffer is ideal for the electrophoresis of relatively small DNA fragments of length of range from 10 bp to 1,000 bp.
- TBE buffer has slightly higher manufacturing cost than TAE buffer. However, it has high ionic strength and high buffering capacity. Therefore, no circulation is necessary for extended running time.

The rate of migration of DNA through agarose gel matrix is affected by following parameters: molecular size of the DNA, concentration of agarose, the conformation of DNA, the presence of ethidium bromide in the gel and electrophoresis buffer, the applied voltage, the type of agarose, and electrophoresis buffer.

At the end of gel electrophoresis, the nucleic acids sieved by gel matrix can be detected and visualized by subjected gel slab to the illumination of ultraviolet (UV) light. Prior to the illumination of UV light, the gel slab is undergoing the staining process.

The most frequently employed staining chemical is ethidium bromide. It has wide applications including the detection of both single and double-stranded nucleic acids. Another advantage of ethidium bromide is that it binds to DNA with very limited sequence preference. The ethidium bromide is prepared as a stock solution of 5 – 10 mg/ml in decontaminated water. After immersing entire gel slab into the solution containing ethidium bromide, the ethidium bromide intercalates into the duplex. The fixed position of planar group and its close proximity to the base pairs allows the fluorescent dye bound to duplex. As a result, the fluorescent signal is improved

significantly as compared to the solution containing fluorescent dye alone (without the involvement of ethidium bromide).

After the staining process, the gel slab is illuminated by ultraviolet (UV) light of approximate 254 nm, the excitation wavelength. The radiation of UV light is absorbed by DNA and subsequently transmitted to the bounded fluorescent dye. The radiation at 302 nm and 366 nm is absorbed by bounded dye itself. Finally, the energy is emitted at wavelength of 590 nm in the red orange region of visible spectrum. Finally, the signal can be captured by dedicated machine.

Although ethidium bromide is cost effective and the resulting image associated with highest resolution, the use of ethidium bromide is limited as it is a mutagen and a suspected carcinogen. Fortunately, there are several alternative staining chemicals, which are capable of minimizing the potential risks, can be used as a perfect replacement of ethidium bromide. These staining chemicals are known as SYBR Gold and Gel Red.

Polymerase chain reaction (PCR)

In biomolecular computing, many laboratory processes are dealing with various enzymes as most of them are effectively dedicated in specific processes. In certain cases, enzymes can also be employed as catalysts to promote the reaction speed. Generally, the most frequently used enzymes in biomolecular computing can be broadly classified into three categories, known as polymerase, restriction endonuclease, and ligase. Polymerase is employed in a process called polymerase chain reaction, abbreviated PCR. PCR plays two key roles in biomolecular computing.

One feature of PCR is to create a single-stranded DNA of specified sequence. Suppose it is crucial to create a single-stranded DNA of sequence, let's say, sequence α . All we need is its complementary sequence (template α'), and part of sequence α (primer β). The primer is complementary to known sequence immediately flanking the region of interest.

In general, the PCR protocol of creating a single-stranded DNA of specified sequence consists of two steps. The entire process can be visually described in Figure 5 – 14. For the sake of simplicity, single-stranded DNA (ssDNA) and double-stranded DNA (dsDNA) are expressed in terms of an arrow and a pair of anti-parallel arrows, respectively. In first step, DNA primer β is

annealed to the complementary region β' of DNA template α , this process is known as primer annealing. The second step is that the DNA polymerase adds individual dNTPs onto the template gradually; such process is known as primer extension. One of the important features of PCR process is that the direction of primer extension is restricted. In other words, polymerase can only extend the primer in the $5' \rightarrow 3'$ direction. Finally, a partial dsDNA is created, which contains desired ssDNA of sequence α . In order to extract demanded ssDNA α , DNA denaturation process is applied at the end of extension process.

Figure 5 – 14: *A schematic representation of creating ssDNA of specified sequence*

Another feature of polymerase chain reaction (PCR) is to multiply the total number of selected double-stranded DNA in the solution, *i.e.*, amplify the “output signal” in biomolecular computing. In this case, the entire double-stranded DNA (dsDNA) is acting as the template. A complete cycle of polymerase chain reaction (PCR) consists of three stages, known as template denaturation, primer annealing, and primer extension, are visually described in Figure 5 – 15. In the template denaturation phase, the double-stranded DNA template is denatured into two complementary single-stranded DNA (ssDNA) by simply increasing the solution temperature far above melting temperature T_m of corresponding dsDNA. In the primer annealing phase, the solution temperature

is decreased to allow the hybridization of two primers to the corresponding two template ssDNAs. In the primer extension phase, the DNA polymerase adds individual dNTPs onto two DNA templates.

By applying a complete cycle of polymerase chain reaction (PCR), the process doubles the total number of targeting double-stranded DNA molecules. By simply repeating the entire process for N number of cycles, it is possible to multiply the initial number of dsDNA for 2^N times, where $N \in \mathbb{Z}_+$ (positive integer number).

Figure 5 – 15: *A schematic representation of one cycle of PCR process to multiply dsDNA*

Restriction endonuclease

In biomolecular computing, another most frequently used enzyme is known as restriction endonuclease (or simply restriction enzyme). By adding restriction enzyme into the solution, it is able to detect and cleave specific sequence of target single-stranded or double-stranded DNA molecule.

For each restriction enzyme, it has its unique restriction recognition site (or simply restriction site) and digestion site. The restriction site is referring to the location of a DNA containing specific base sequences of nucleotides, which can be detected by specific restriction endonuclease. The digestion site is the location where the restriction endonuclease cleaves the phosphodiester bonds of DNA strands.

According to the different cleaving mechanism, restriction enzymes can be broadly classified into three categories: type I, type II, and type III, depending on the distance between restriction site and digestion site. The cleaving mechanisms for different classes of restriction enzymes in cleaving double-stranded DNA (dsDNA) are explained in Figure 5 – 16. The partial sequences of dsDNA, which are surrounded by rectangular box, are the recognition site of specific restriction endonuclease; the strand of dsDNA, which are pointed by arrow head, representing the digestion site of restriction endonuclease.

As indicated in Figure 5 – 16, the difference between three classes of restriction endonuclease can be explained as follows:

- *Type I restriction endonuclease*: the digestion site of restriction endonuclease is far away from the restriction recognition site.
- *Type II restriction endonuclease*: the digestion site of restriction endonuclease is retained within the restriction recognition site.
- *Type III restriction endonuclease*: the digestion site of restriction endonuclease has a short distance from the restriction recognition site.

Figure 5 – 16: *A schematic description of cleaving mechanisms of restriction endonucleases*

Furthermore, as a consequence of digestion, the original double-stranded DNA (dsDNA) results into one of two possible ends, known as blunt end and sticky end (or overhanging end). For a blunt-ended dsDNA, both strands terminate in a base pair. For a sticky-ended dsDNA, a stretch of unpaired nucleotides exist for both resulting strands. These two resulting ends are visually described by Figure 5 – 17.

Figure 5 – 17: *A schematic representation of two types of ends as a result of cleaving*

Ligase

A reverse process of cleaving DNA through phosphodiester bonds is done by adding DNA ligase into the solution. The working principle of DNA ligase is to repair the phosphodiester bonds between adjacent DNA strands. The DNA ligase works for both blunt ends and sticky ends. However, the efficiency of DNA ligase to sticky ends is much higher than the blunt ends. In biomolecular computing, DNA ligase is frequently added into the solution at the end of hybridization process. As visually described in Figure 5 – 18, two DNA solutions contain identical DNA molecules. However, one solution is added with DNA ligase and the other is not. The difference can be observed by simply increasing the solution temperature (denaturation).

Figure 5 – 18: *A schematic comparison between two identical DNA solutions with and without the addition of DNA ligase after hybridization process*

Sequencing

In biomolecular computing, it is always ideal to detect the actual content of DNA in terms of DNA sequences of A, T, G, and C. Such process is referred to as DNA sequencing. Majority of sequencing method available nowadays are the variations of conventional sequencing method developed by F. Sanger in 1977, which is also termed as Sanger's method or chain-terminating method [86-87].

The chain-terminating method relies on dideoxynucleotides, abbreviated as ddNTPs, polymerase chain reaction (PCR), and gel electrophoresis. The working principle of chain-terminating method is simple. The solution containing target DNA molecules is firstly separated into four test tubes. In each test tube, one of four available fluorescently labeled dideoxynucleotides, ddATP, ddTTP, ddGTP, and ddCTP, are added into the corresponding test tubes. These test tubes are subjected to complete with cycles of polymerase chain reaction, *i.e.*,

template denaturation, primer annealing, and primer extension. During each round of PCR, primer extension process is stochastically terminated by incorporating fluorescently labeled ddNTPs instead of dNTPs.

For a successful primer extension stage of PCR, three elements must be fulfilled including 3' end hydroxyl ($3' - \text{OH}$), 5' end phosphate ($5' - \text{P}$), and polymerase. With the addition of ddNTPs, there is certain change that the original $3' - \text{OH}$ is replaced by a 3' end hydroxyl group supplied by ddNTPs, which terminates the primer extension process. After PCR process, by inserting the solution in four test tubes into solution slots of gel slab and subjected to gel electrophoresis, it is to determine the sequence of DNA in solution. Nowadays, DNA strands of length of up to 1,000 bp can be sequenced with extremely high accuracy (99.999%) at reasonable cost (approximately \$0.5 per kbp) [88-89].

5.2 Theoretical Computer Science

The ultimate target of biomolecular computing is to develop a functional biomolecular computer to compete with contemporary silicon-based digital computer. However, as compared to the prolonged history of silicon-based computing, biomolecular computing is still remaining at the proof-of-concept stage of development.

The field of biomolecular computing was initiated by solving one specific class of mathematical problems, namely, combinatorial optimization problems. This specific class of mathematical problems are generally modeled by graphs and solved by algorithms provided by graph theory. It is believed that solving this specific class of mathematical problems merely by biomolecular means can utilize the advantage of biomolecular computing, which is its massive parallel computational ability.

Although the proposed algorithm is capable of maximizing the parallel computational ability, their applications are restricted to one specific class of mathematical problem, that is, one biomolecular-based algorithm is only feasible to one specific problem. Hence, in order to develop a general purpose biomolecular computing, analogous to contemporary digital computer, designing appropriate biomolecular logic circuits is believed to be a feasible starting point.

In order to summarize and explain the developments of biomolecular computing, it is inevitable dealing with the technical terms and definitions of graph theory as well as Boolean logic. However, for those researchers from unrelated disciplines may find difficult to understand these concepts. Therefore, prior to the literature review of developments of biomolecular computing, this section is provided for these researchers to clear the potential barriers. The entire subchapter can be broadly separated into two sections – graph theory and combinatorial optimization problem (section 5.2.1) and Boolean logic (section 5.2.2).

5.2.1 Graph theory and combinatorial optimization problems

Graph

A graph, denoted by G , consists of two finite elements: a set of vertices, denoted by V , and a set of edges, denoted by E . Such relationship can be expressed as $G = (V, E)$. Vertices are a set of points, which can be expressed in terms of a series of strings, $v_1, v_2, \dots, v_n$ or simply a series of integer numbers $1, 2, \dots, n$. Such relationship can be described by $V = (v_1, v_2, \dots, v_n)$ or $V = (1, 2, \dots, n)$. The edges are a set of connecting lines, which is used to link any two vertices i and j , where $i, j \in \mathbb{Z}_+$. There are two types of edges, directed edges and non-directed edges. For non-directed edges, it can be simply denoted by e_{ij} or (i, j) . The directed edges can only be represented by $i \rightarrow j$. The graph consists of directed edges termed as directed graph, or simply, digraph. In a graph, the vertex is represented by a node. The non-directed and directed edge is represented by a line and arrow, respectively. The graphical representation of a typical non-directed graph and a directed graph of vertices i and j are given in Figure 5 – 19.

Figure 5 – 19: *Non-directed graph and directed graph*

The graph theory was initiated by Leonard Euler in 1735. It is used to describe a problem in terms of graph and solve the problem by using specifically developed algorithmic models. It is especially efficient to a class of problem, namely combinatorial optimization problem, which are of pronounced discrete or combinatorial structure. There are several famous combinatorial

optimization problems including shortest path problem, minimal spanning tree problem, assignment problem, and maximal flow problem.

Shortest path problem

One of the most important classes of optimization problem is the shortest path problem. It has wide applications in daily life such as designing the fastest (or least expensive) routes for delivery man, cargo carrier, and so on.

Prior to the definition of shortest path problem, it is important to differentiate the following four terms: walk, trail, path, and cycle. A walk is a route from any vertex v_i to any vertex v_j without any restrictions, *i.e.*, it may pass some of the vertices or edges for more than once. A trail is a walk from any vertex v_i to any vertex v_j , whereas the walk may pass through each edge at most once. A path is a trail from any vertex v_i to any vertex v_j , whereas the trail may pass through each vertex at most once. A cycle is a path from any vertex v_i and back to the same vertex v_i , *i.e.*, the starting vertex and terminating vertex is the same vertex.

In order to define the concept of a shortest path, supposing the graph G is a weighted graph, *i.e.*, each edge in G has a given length. Therefore, the shortest path problem is the problem of determining the path between two assigned vertices which minimize the sum of lengths of its edges. The famous mathematical algorithm, which is used to solve shortest path problem is known as Dijkstra's algorithm [90].

A sample of shortest path problem, as indicated in Figure 5 – 20, is provided to demonstrate the concept. As shown in Figure 5 – 20, the sample problem consists of six vertices and eleven edges. The number (in gray color) at the middle of each edge specifies the length of corresponding edge. The solution paths, shortest paths, from vertex 1 to any other vertices, are provided at right hand side of Figure 5 – 20.

Figure 5 – 20: *Shortest path problem and shortest paths from vertex 1 to any other vertices in G*

Minimal spanning tree problem

Another crucial optimization problem is minimal spanning tree problem. It is frequently used in estimating the optimal routes for railway lines that connects a number of physical places while minimizing the total construction cost.

A spanning tree, denoted by T , is a tree containing all the vertices in graph $G = (V, E)$ but has no cycles. Therefore, the minimal spanning tree in a fully connected and weighted graph G is defined as a spanning tree for which the sum of lengths of edges is minimal. Two mathematical algorithms are generally used to solve the minimal spanning tree problem, which are known as Greedy algorithm and Prim's algorithm [91].

Similarly, in order to demonstrate the concept, an example of minimal spanning tree problem is provided in Figure 5 – 21. It consists of five vertices and eight edges. Analogous, the number (in

gray color) at the center of each edge is the length of that specific edge. The solution, minimum spanning tree in graph G, is given at the bottom of Figure 5 – 21.

Figure 5 – 21: *Minimal spanning tree problem and its corresponding solution*

Assignment problem

In addition to shortest path and minimal spanning tree problem, another important class of combinatorial problem is assignment problem. The assignment problem is always related to allocating task to workers, goods to storage, students to exams, and so on. In addition to that, the assignment problem is always related to strategic applications.

The graph $G = (V, E)$ associated with assignment problem is termed as bipartite graph. In a bipartite graph, the entire set of vertices is partitioned into two sets S and T , which results in every edge in G having one end in S and the other end in T . The objective of assignment problem is to determine the matching set M of greatest number of edges and no two of them have a vertex in common.

Similar to previous two problems, an example of assignment problem and its corresponding solution is provided to illustrate the concept. As shown in Figure 5 – 22, there are totally eight

vertices, four of them in set S, and the remaining in set T, and eight edges. The objective is to determine the maximal matching set M within graph G .

Figure 5 – 22: *Assignment problem and its corresponding solution*

5.2.2 Boolean algebra

A numeral is a symbol that represents a quantity. A number system is a method of assigning combinations of numerals to different quantities. The base 10 or decimal number system is the most frequently used number system in majority of cultures.

The simplest number system is base 2 or binary number system, which is mainly dealing with two digits, that is, 0 and 1. Binary number system is critical as the contemporary digital

computers are counting in base 2. They fundamentally operate on the principle of Boolean algebra, a typical system based upon base 2 number system. Within the circuits of a digital computer, two different voltage levels are corresponding to 0 and 1, respectively. The section is dedicated in the introduction of Venn diagrams and several most frequently used Boolean functions.

Venn diagrams

In sets algebra, there are two fundamental things: the set and element. Set is a group of objects, or elements. An object inside a set is called an element of that corresponding set. The concept of sets algebra is really abstract. Therefore, it can be visually expressed in terms of the Venn diagrams, which were initially developed by Reverend J. Venn.

A Venn diagram begins with a rectangular box that represents the Universe, denoted by U . A set is generally represented by a closed circle inside the rectangular box. Objects inside the closed circle are treated as the elements of the set. Objects outside the closed circle are not regarded as the elements of the set. In addition to that, a particular set in a Venn diagram is generally indicated by shading it. The set of all objects that are not in a particular set, let's say, x is termed as the complement of x , or x' . In Figure 5 – 23 (a), it is a Venn diagram of a set x . The complement of x , x' , is indicated in Figure 5 – 23 (b).

Figure 5 – 23: *Venn diagram of set x and its complement*

In a Universe, which contains more than a single set, there are several possible relationships among these sets. These relationships are called subsets, supersets, intersection, and union. When all the elements of one set are also the elements of the other set, the first set is termed as the subset

of the second set, and the second set is the superset of the first set. The symbol $\subset$ means “is a subset of”.

The intersection of two (or more) sets is the set of objects in both (or all) of those sets. The symbol of intersection is $\cap$. The shaded area in Figure 5 – 24 (a) is the intersection of set a and set b. The union implies the merging of separate things. The symbol of union is $\cup$. A set is always a subset of union of itself and any other sets. The shaded area in Venn diagram of Figure 5 – 24 (b) is the union of set a and set b.

(a) Intersection of sets a and b

(b) Union of sets a and b

Figure 5 – 24: Intersection and union of sets a and b

Venn diagram has a close relationship with the truth table for a Boolean expression. The intersection function in sets algebra is analogous to the Boolean function AND. The union function in sets algebra is analogous to the Boolean algebra OR.

Boolean functions

In the binary system, it is mainly dealing with two digits, 0 and 1. These binary digits are generally termed as bits. The manipulation of bits is called Boolean functions. Most of the Boolean functions are binary functions, that is, they take two input bits and produce only one output bit. The only exception is the NOT function, which is a unary function, *i.e.*, one input bit produces one output bit. The truth table and circuit symbol of NOT function are indicated in Figure 5 – 25.

Figure 5 – 25: *Circuit symbol and truth table of NOT function*

Truth table for all possible two-input Boolean functions are concluded in Figure 5 – 26. However, not all of these Boolean functions are of interest. Column 0 and column 15 are not of interest as no matter what input bits are, the output bits are always the same constant. These Boolean functions are called the constant 0 function and the constant 1 function, respectively.

X	Y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Figure 5 – 26: *Truth table for all possible two input bits Boolean functions*

The truth table for column 8 is AND function, where the output bit is 1 if and only if both input bits are 1. The algebraic symbol for AND function is $\wedge$. The circuit symbol and truth table of AND function are specified in Figure 5 – 27.

Figure 5 – 27: *Circuit symbol and truth table of AND function*

The truth table for column 14 is OR function, where the output bit is 1 if either input bit is 1. The algebraic symbol for OR function is $\vee$. The circuit symbol and truth table of OR function are specified in Figure 5 – 28.

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Figure 5 – 28: *Circuit symbol and truth table of OR function*

The truth table for column 7 and column 1 are NAND and NOR function, respectively. NAND is the short expression of NOT AND function, which simply means a combination of AND together with NOT function. Identically, NOR is the short form of NOT OR function, which means a combination of OR together with NOT function. The algebraic symbol for NAND function is called Sheffer stroke, denoted by $\uparrow$. The algebraic symbol for NOR function is termed as Peirce arrow, denoted by $\downarrow$. The circuit symbols and truth tables of NAND and NOR function are indicated in Figure 5 – 29. In the circuit diagram, the NOT gate right next to other gates can simply be expressed in terms of a small circle.

x	y	$x \uparrow y$
0	0	1
0	1	1
1	0	1
1	1	0

x	y	$x \downarrow y$
0	0	1
0	1	0
1	0	0
1	1	0

Figure 5 – 29: *Circuit symbol and truth table of NAND and NOR function*

The truth table for column 6 is XOR function. The OR function, as specified in column 14, is also known as inclusive OR function. In this case, the XOR is referred to as exclusive OR function. The only difference between inclusive OR and exclusive OR function is when both input bits are 1, the out bit is zero for XOR. The algebraic symbol for XOR function is $\oplus$. The corresponding circuit symbol and truth table for XOR function is specified in Figure 5 – 30.

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Figure 5 – 30: *Circuit symbol and truth table of XOR function*

The truth table for column 9 is a COIN function. The COIN is coincidence function, which returns output bit of 1 if and only if both input bits are the same. The algebraic symbol for COIN is $\odot$. The respective circuit symbol and truth table for COIN function is specified in Figure 5 – 31. By comparing Figure 5 – 30 and Figure 5 – 31, it is possible to observe that $x \oplus y$ and $x \odot y$ are complementary to each other.

x	y	$x \odot y$
0	0	1
0	1	0
1	0	0
1	1	1

Figure 5 – 31: *Circuit symbol and truth table of COIN function*

The last important Boolean function is implication function. Unlike previous two-input bit Boolean functions, the implication function is asymmetric, *i.e.*, $x \rightarrow y$ is not equivalent to $y \rightarrow x$. There is no specific circuit symbol for implication function. The truth tables of implication functions are listed in Figure 5 – 32.

x	y	$x \rightarrow y$
0	0	1
0	1	1
1	0	0
1	1	1

x	y	$y \rightarrow x$
0	0	1
0	1	0
1	0	1
1	1	1

Figure 5 – 32: *Truth table of implication functions*

5.3 Developments of Biomolecular Mediated Computing

Since the first silicon-based microcomputer was introduced at the beginning of 1970s, improving the computational ability has always been the top-ranked concerns among researchers all over the world. For an electronic based digital computer, in order to confront the growing demands for computational ability, processing speed and parallelism, the size of individual transistor elements has to be reduced, such that allows additional processing elements to be packed onto the same silicon chip. The increasing packing density always led to problems, including increasing power consumption and problematic heat dissipation. Despite that, the entire semiconductor industry is quickly approaching the physical constraints of Moore' Law.

In reality, any devices endowed with three fundamental functions – processing, storing, and displaying information – can be regarded as a computer. Therefore, the researchers from various disciplines are engaged in exploring potential alternatives to electronic-based digital computer. Among various intriguing approaches, biomolecular mediated computing seems to be the most feasible method to establish the computer of next generation. The biomolecular mediated computing was arguably initiated by L. M. Adleman, who had demonstrated the employment of artificially synthesized DNA as a simple computational device to solve a seven-vertex directed Hamiltonian Path Problem in 1994 [37]. Shortly after the contribution of L. M. Adleman, it was theoretically modified and extended into the Boolean logic operations, which is known as the satisfactory problem [38]. Since that, the subsequent researches of biomolecular mediated computing can be broadly classified into two categories, known as limited-function biomolecular mediated devices and biomolecular mediated logic circuits.

The major focus of this section is to selectively review various works has been done in these two categories. The first subsection, subsection 5.3.1, selectively reviews the past development of limited-function biomolecular mediated devices. The second subsection, subsection 5.3.2, selectively revisits several crucial developments of biomolecular mediated logic circuits.

5.3.1 Limited-function biomolecular mediated devices

The first category of the researchers focus on the development of limited-function biomolecular mediated devices. The proposed conceptual models maximize the advantage of biomolecular mediated computing, that is, its massive parallel computational ability. There are many mathematical problems have been solved by developed conceptual models [37 – 62]. In this subsection, several crucial biomolecular mediated models are selectively revisited as follows:

Reference: L. M. Adleman, “Molecular computation of solutions to combinatorial problems” [37]

The computational ability of DNA molecules is first demonstrated by using in-vitro manipulations. The design algorithm is used to solve a non-deterministic polynomial time problem – directed Hamiltonian path problem (abbreviated HPP).

The proposed algorithm is a typical 5 stage approach as shown below:

1. Generate random paths through the graph.
Each vertex O_i is encoded with a random 20-mer sequence of DNA. The edge $O_{i \rightarrow j}$ is designed to join 20-mer $\overline{O}_i$ to 20-mer $\overline{O}_j$ by using 10-mer for each sequence.
50 pmol of each individual DNA sequences are mixed during the hybridization. For each edge in graph, there are approximately 3×10^{13} copies of the associated oligonucleotide were added to the ligation reaction.
2. Keep only those paths that begin with v_{in} and end with v_{out}
Product of step 1 is amplified by polymerase chain reaction (PCR) using appropriate designed oligonucleotides.
3. If the graph has n vertices, then keep only those paths that enter exactly n vertices
Product of step 2 is subjected to agarose gel electrophoresis, the 140-base pair (bp) band is retained and extracted by soaking in doubly distilled H_2O (ddH_2O).
4. Keep only those paths that enter all of the vertices of the graph at least once.
Product of step 3 is affinity-purified with a biotin-avidin magnetic beads system.
5. If any paths remain, say “Yes”; otherwise, say “No”.
Product of step 4 is amplified by PCR and subjected to agarose gel electrophoresis.

The proposed method is relatively time-consuming, which requires approximately 7 days of laboratory work. The biotin-avidin magnetic bead separation method is the most labor-intensive.

The molecular-based computing has several advantages over the electronic computer. Firstly, if the ligation of two molecules is considered as a single operation, 10^{20} or more operations can be executed per second. As compared to that, the fastest supercomputers can execute approximately 10^{12} operations per second. Secondly, 1 joule is sufficient for approximately 2×10^{19} ligation operation based on Gibbs free energy $\Delta G = -8 \text{ kcal/mol}$. For existing supercomputer executes 10^9 operations per joule. Lastly, the information density for DNA is 1 bit per cubic nanometer. For videotapes, the density of information is approximately 1 bit per 10^{12} nm^3 . However, the advantage of electronic computers is the variety of operations and flexibility with which these operations can be applied.

Reference: Qi Ouyang et al., “DNA solution of the maximal clique problem” [39]

A new DNA search algorithm is developed to solve the maximal clique problem. The problem is that given a network containing N vertices and M edges, determine the number of vertices in the largest clique.

The algorithm consists of four stages of approach:

1. In graph G , which consists of N vertices, each possible clique is converted in terms of N – digit binary number representation. Bit set equals to 1 if a vertex present in clique, otherwise, the bit set equals to 0

Example (“543210”):

Clique	Binary representation
(4, 1, 0)	010011
(5, 4, 3, 2)	111100

2. Locate pairs of vertices in the graph that are not connected by an edge based upon complementary graph (example in Figure 3 – 33), which is the graph contains all edges missing in primal graph

Figure 5 – 33: *Example of primal graph and its complementary graph*

3. Eliminate from the complete data pool all numbers containing connections in complementary graph

Example:

Binary representation	xxx1x1	1xxxx1	xx1x1x	1xxx1x
Edge	(0,2)	(0, 5)	(1, 3)	(1, 5)

4. Sort remaining data pool to find out data containing the largest number of 1's

The designated algorithm is executed in terms of a combination of DNA-based computing techniques including problem encoding, parallel overlap assembly (abbreviated POA), polymerase chain reaction (abbreviated PCR), and digestion of restriction endonuclease.

1. *Problem encoding*: each bit in binary number is represented by two DNA sections to specify its value (V_i) and position (P_i). Each bit's position (P_i) is expressed in terms of 20 mer random ssDNA; Each bits value (V_i) is expressed in terms of either 0 mer ssDNA (if $V_i = 1$) or 10 mer ssDNA (if $V_i = 0$)
2. *Parallel overlap assembly (POA)*: the synthesized ssDNA (or oligonucleotide) is subjected to thermal cycling. For each thermal cycle, position string in one ssDNA annealed to the complementary strings of next oligonucleotide, and 3' ends extended in the presence of polymerase to form a longer dsDNA.
3. *Polymerase chain reaction (PCR)*: exponential amplification of molecules start with specified DNA sequence
4. *Digestion of restriction endonuclease*: eliminate the edges specified in complementary graph from complete data pool. The broken DNA sequences are not amplified by PCR with identical primer.

Example (cut connection of $(0, 2)$):

- a. Divide complete data pool into two test tubes, namely t_1 and t_2
- b. Add corresponding restriction enzyme into the test tube t_1 and t_2 (one link at a time)
- c. Combine test tube t_1 and t_2 into final test tube t , which do not contain connection of $(0, 2)$

5. *Polymerase chain reaction*

6. *Gel electrophoresis*

Reference: Frank Guarnieri et al., “Making DNA add” [40]

A prerequisite for designing a computer useful in a wide range of applications is the ability to perform mathematical calculations. In the article, a DNA-based algorithm for addition of two nonnegative binary numbers is presented.

The mathematical calculation is a different problem from the search problem as proposed by Adleman and Lipton in 1994 and 1995, respectively. The search problem is solved by generating all possible combinations and searching for the correct output, whereas binary operations require that only the correct output is produced in response to specific inputs.

Recall that the decimal counting using the symbols 0 through 9, while binary only uses symbols 0 and 1. Since the binary is base-2 system, each digit represents an increasing power of 2. The conversion between binary number and decimal number from decimal 0 to decimal 16 is summarized in Table 5 – 3.

A general encoding theme and complete process for DNA-based addition of two nonnegative one-digit binary numbers are presented in Figure 5 – 34. All these DNA sequences are single-stranded, unique, and non-complementary, except the negation symbol indicates a complementary relationship.

Similarly, the general encoding theme for DNA-based addition of two nonnegative two-digit binary numbers is presented in Figure 5 – 35.

Decimal number	Binary number
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000

Table 5 – 3: *Conversion table of decimal and binary number*

The selected study case for addition of two-digit binary number $11 + 01$ is illustrated in Figure 5 – 36. The method proposed in this article, however, has an obvious limitation: the output is encoded in a different form than the input. Therefore, it is not presently possible to perform parallel addition.

Encoding theme: 2^0 position

1st digit: 0

2nd digit: 0

Placeholder for 2^1 position

DNA-based algorithm for adding two binary number

Figure 5 – 34: DNA representation of all possible non-negative one-digit binary integers

Figure 5 – 35: DNA representation of all possible nonnegative two-digit binary integers

Figure 5 – 36: DNA operation of binary addition of 11 and 01

Reference: Masahito Yamamoto et al., “DNA solution of the shortest path problem by concentration control” [41]

One of the major disadvantages of Adleman and Lipton is the explosive increase of candidate solutions due to the size of problem increases. The concentration control method is used to demonstrate a shortest path problem.

The highlight of the article is in encoding the cost of edge in graph by using the concentration approach and the specific purification method called single-strand conformation polymorphism (abbreviated SSCP).

The relative concentration D_{ij} of each oligonucleotide encoding the edge with cost C_{ij} is calculated by using the equation 5.16:

$$D_{ij} = \left(\frac{\text{Min}}{C_{ij}} \right)^\alpha \quad (5.16)$$

Where Min is the minimum value among costs of all edges in the graph, and α is a parameter value.

The single-strand conformation polymorphism (SSCP) is a simple technique for separation and purification of DNA paths according to their size, base sequence and base composition.

Reference: Ji Youn Lee et al., “Solving traveling salesman problems with DNA molecules encoding numerical values” [44]

Travelling salesman problem is to find a minimum length (or cost) path for a given set of vertices and edges. In addition, the path must begin and terminate at a specified vertex and pass through all the vertices.

Two previous methods, which either use length or concentration to represent the cost of the edges, are not ideal as pointed by the researcher. Representing the cost of edges by using length is not experimentally implementable, as very long DNA strands must be used to encode the problem, resulting in extremely high cost and unreliable result. The method involving concentration control is able to facilitate the local search among candidate solutions by reducing the size of the problem space. However, the solution is not ensured to be optimal and failed to retract the DNA solution from gel matrix.

In this paper, DNA strands are designed to encode the real values by variation of their corresponding melting temperature – DNA strands for high-cost values are assigned with high melting temperatures than those for low-cost values of edge. Subsequently, two biochemical techniques, known as denaturation temperature gradient polymerase chain reaction (PCR) and temperature gradient gel electrophoresis, are exploited as the computational operators. The denaturation temperature gradient polymerase chain reaction (PCR) is aiming to start the thermal cycle of PCR from relatively low denaturation temperature and increase the temperature gradually, cycle by cycle. The low melting temperature DNA sequences, representing the cost of edge, is designed to generate more number of paths during the temperature gradient polymerase chain reaction (PCR). After that, temperature gradient gel electrophoresis is used to separate the most economical path among the candidate paths.

5.3.2 Biomolecular mediated logic circuits

The second category of the researchers emphasize on the development of biomolecular mediated logic circuits. Although the biomolecular mediated limited-function device, as specified in the first category, maximize the advantage of biomolecular mediated computing, that is, its massive parallel computational ability. However, their applications are only restricted to one specific class of mathematical problems. As claimed by the researchers in this category, in order to develop a general-purpose biomolecular computer, analogous to contemporary electronic mediated computer, designing appropriate biomolecular logic circuits is believed to be a feasible starting point. The logic circuits are assembled by a series of logic gates which is capable of performing Boolean logic operations.

The concept of building up a biomolecular mediated logic gate was formally proposed by C. P. Collier [48]. However, the proposed device can hardly be accepted as a potential biomolecular logic gate. The first real meaning universal biomolecular logic gate was proposed and experimentally demonstrated by N. C. Seeman [49]. The design, on the other hand, is extremely complicated as each signal is defined as a DNA triple-crossover molecule with two sticky ends, which makes assembling a logic circuit totally impractical. In addition to that, displaying the output signal in terms of gel image as a result of gel electrophoresis is an inefficient way of information representation. The only information revealed by gel electrophoresis is the size of DNA molecule rather than the actual content. Furthermore, there are many other early stage designs of biomolecular logic gate which are perfectly summarized in [50 – 52].

The information accessibility of biomolecular computing is significantly improved with the introduction of surface-based techniques and fluorescent signal representation. Such method was firstly employed by [53], which allows one end of DNA molecule to be attached onto the solid surface during the entire process. Such strategy greatly improved the traceability of DNA molecules. In addition to that, representing the output signal in terms of fluorescent signal can be deemed as a significant advance for biomolecular mediated computing. The method of attaching fluorescent dye to DNA strand representing output in terms of fluorescent signal are further reinforced by A. Saghatelian [54] in the DNA-based photonic logic gates.

Since that, there are many conceptual models of biomolecular logic gates have been proposed. Of these models, the modification of nucleobase to improve the overall stability of base-pairing [55] seems intriguing. However, no matter how delightful the design of various logic gates is, the

information as the output from previous logic gate, which encoded in terms of DNA sequences, has to be passed to its subsequent logic gate. This is the fundamental criterion in assembling individual logic gates to the formation of biomolecular logic circuits. However, it seems that none of these approaches are capable of satisfying this crucial criterion.

Fortunately, the problem has been notified and corrected in the subsequent approaches. These conceptual models, which satisfy the fundamental criterion of logic circuit design, can be broadly classified in one of the two categories, namely, biochemically controlled logic gates [56 – 62] and strand displacement based (enzyme-free) logic gates [63 – 80]. In this subsection, several important biomolecular mediated logic gates as well as logic circuits are selectively revisited.

Reference: R. J. Lipton, “DNA solution of hard computational problems” [38]

The speed of computer is determined by two major factors: (1) how many parallel processes it has; (2) how many steps each can perform per unit time. The Adleman’s original method has been extended to solve another famous NP-complete problem – the “satisfaction” problem (SAT). The SAT problem is to find Boolean value for x and y that makes the following formula true.

$$F = (x \vee y) \wedge (\bar{x} \vee \bar{y}) \quad (5.17)$$

The truth table of formula F can be expressed as shown in Table 5 – 4.

x	y	$\bar{x}$	$\bar{y}$	$x \vee y$	$\bar{x} \vee \bar{y}$	$(x \vee y) \wedge (\bar{x} \vee \bar{y})$
0	0	1	1	0	1	0
0	1	1	0	1	1	1
1	0	0	1	1	1	1
1	1	0	0	1	0	0

Table 5 – 4: Truth table of formula F

In general, a SAT problem consists of Boolean formula of form $C_1 \wedge \cdots \wedge C_m$, where each C_i is clause. Therefore, the formula F consists of two clauses, where $C_1 = (x \vee y)$ and $C_2 = (\bar{x} \vee \bar{y})$. Making all of the clauses true is often called “satisfying” the clauses.

The graph G_n is constructed so that all paths that starts at a_1 and end at a_{n+1} encode an n -bit binary number, as shown in Figure 5 – 37. The paths of length $n + 1$ that start at a_1 and end at a_{n+1} are assumed in the initial test tube. At each stage, a path has exactly two choices: if it takes the vertex with primed label, it will encode a 0; otherwise, it will encode a 1.

Figure 5 – 37: The graph G_n which encodes two-bit numbers

Each vertex has been encoded with 20-mer single-stranded DNA sequence with two parts: the first half is denoted by p_l and the second half is denoted by q_l . The initial test tube consists of following kinds of DNA sequences:

- Vertex DNA sequence: $5' \rightarrow 3'$ DNA sequence of the form $p_l q_l$
- Edge DNA sequence: $3' \rightarrow 5'$ DNA sequence of the form $\hat{q}_l \hat{p}_l$ ($\hat{x}$ denotes the complementary sequence to x)
- Sealing DNA sequences: $3' \rightarrow 5'$ sequence of length $l/2$ complementary to the first half of initial vertex to test tube; $3' \rightarrow 5'$ sequence of length $l/2$ complementary to last half of the final vertex to test tube.

Operations are performed only on the DNA sequences from the graph G_n . In all of the following, the strands of DNA can be assumed to be single DNA strands. The $E(t, i, a)$ expression is used to denote all of the sequences in test tube t for which the i th bit is equal to a , where $a \in \{0, 1\}$.

The experiment of the case study can be implemented as follows. The initial test tube t_0 is the test tube which contains all specified DNA solutions as stated above.

1. Let t_1 be the test tube that corresponds to $E(t_0, 1, 1)$. The remainder be t_1' and let test tube t_2 be $E(t_1', 2, 1)$. Pour t_1 and t_2 together to form t_3 .
2. Let t_4 be the test tube that corresponds to $E(t_3, 1, 0)$. Let the remainder be t_4' . Let t_5 be $E(t_4', 2, 0)$. Pour t_4 and t_5 together to form t_6 .
3. Check to see if there is any DNA in the last test tube.

The entire process can be summarized in Table 5 – 5.

Test tube	Values present
t_0	00, 01, 10, 11
t_1	10, 11
t_1'	00, 01
t_2	01
t_3	01, 10, 11
t_4	01
t_4'	10, 11
t_5	10
t_6	01, 10

Table 5 – 5: *DNA in test tube and corresponding DNA sequences*

Reference: Q. H. Liu et al., “DNA computing on surfaces” [53]

The surface-based DNA computing technique was initially developed to solve the three-variable satisfactory (SAT) problem. The initial consideration of surface-based technique consists of three stages: prepare the single-stranded DNA with appropriate base sequences and bind them to the surface with appropriate attachment chemistry; a search technique is subsequently used to eliminate the strands that are not the answers to the problem; the remaining solutions on the surface after several cycles of elimination process are deciphered by using DNA sequencing method.

Currently, surface-based DNA computing technique is widely used in constructing the DNA-based logic gate. There is mainly due to four reasons as summarized by Susan D. Gillmor et al.: simple computational operations, minimize the loss of DNA between each step, no interaction

between single-stranded DNA leads to unexpected hybridization, and effective sequence purification.

Reference: A. Saghatelian et al., “DNA-based photonic logic gates: AND, NAND, and INHIBIT” [54]

The first successful demonstration of DNA-based logic gate was produced by A. Saghatelian et al. One 16 mer single-stranded DNA with carboxyl fluorescein moiety modified 3' end is used to represent an AND gate or an NAND gate. The presence of the input signal is termed as true (1), otherwise, the input signal is termed as false (0). One of the two input signals is fixed, which is the complementary strand of 16 mer single-stranded DNA that represents the logic gate. The selected second input signal is different for AND gate and NAND gate: for the AND gate, the input signal is a typical fluorescent strain called Hoechst 33342; for the NAND gate, the input signal is another type of DNA detection agent called Ethidium Bromide, abbreviated EtBr.

For DNA-based photonic AND gate, if and only if both input signal presents (*i.e.* the presence of 16 mer complementary single-stranded DNA and Hoechst 33342), the mixed result has a strong fluorescent signal (*i.e.* the output signal is 1). Otherwise, there is no emission of mix under 520 nm ultraviolet light (*i.e.* the output signal is 0). The working principle of NAND gate is very much similar to that of AND, except change the chemical from Hoechst 33342 to Ethidium Bromide.

The significance of the article is the first proof-of-concept that DNA molecule can be used in logic design. In addition, it proved that the truth table can be created in terms of the fluorescence outputs.

Reference: K. S. Park et al., “Illusionary polymerase activity triggered by metal ions: use for molecular logic gate operations” [58]

As mentioned by K. S. Park et al, the existing logic gates are typically rely on relatively complicated design for gate switching mechanism, which leads to high construction cost. Therefore, a new and simple alternative strategy of constructing the molecular scale logic gates

was proposed, which is based on polymerase chain reaction at the mismatched site triggered by metal ion. The strategy is used to construct four DNA-based logic gates: YES gate, PASS1 gate, AND gate, and OR gate. The working principle of these four created logic gates is somehow with very limited variations. Therefore, in the following paragraph, only the working principles that used to construct the DNA-based AND gate is illustrated.

The AND gate produces an output of 1 only if both input signal are held at 1. The DNA-based AND gate requires one double-stranded DNA, Taq polymerase, and two artificially designated primers, which causes T – T pair and C – C pair mismatches. The two input signal is the presence (1) or absent (0) of selected metal ions Hg^{2+} and Ag^+ , respectively. The detailed process is described in Figure 5 – 38.

The complete DNA solution, which contains the template (double-stranded DNA) and two primers, is separated into four test tubes. The metal ion is added (1) or not added (0) into the test tube as described in Figure 5 – 38.

If the both metal ions are absent in the test tube during the polymerase chain reaction, the primer is not extended due to the mismatch of T – T and C – C pairs. As a result, there is no fluorescent signal present in the gel slab as a result of subsequent gel electrophoresis process. In order to produce an output of 1, both input signal must present in the solution. The T – T mismatch pairs is triggered by metal ion Hg^{2+} to the formation of T – Hg – T, which allows the success of primer extension. Similarly, the C – C mismatch pairs is triggered by metal ion Ag^+ to the formation of C – Ag – C.

The significant of the proposed method is that the DNA-based logic gate can be simply constructed by using PCR amplification as an output and metal ion as triggers. However, it is impossible to assemble the developed DNA-based logic gates into a logic circuit as the output signal from the previous gate cannot be used as the input signal to the subsequent logic gate.

Figure 5 – 38: A schematic representation of illusionary polymerase chain reaction in DNA-based logic AND gate construction

Reference: B. M. Frezza et al., “Modular multi-level circuits from immobilized DNA-based logic gates” [63]

In order to wire individual logic gates in the formation of molecular multi-level circuits, the output signal of previous molecular logic gate must be used as an input signal to the subsequent logic gate. However, most of the previously developed logic gates are failed to do so. The proposed method first successfully wiring the developed logic gates (AND, OR, AND-NOT and REPORT logic gates) into a three-level molecular circuit.

Figure 5 – 39: A schematic representation of toe-hold strand displacement method

The “toe-hold” strand replacement method is first used in constructing the DNA-based logic gate. In such method, a full-length complementary strand binds to a single-stranded DNA with sticky ends known as “toe-hold” and proceeds to invade and replace the shorter annealed sequences via a three way branch mechanism. Such method is described in Figure 5 – 39.

However, there is one notable drawback for the proposed method. The imperfect strand displacement results in the minor loss of input or output signal at each level of circuit operation, which will limit the total number of logic gates that can be constructed.

Chapter 6

Biomolecular Mediated Solution of Strategic Assignment Problem²²

The proposed idea of financial option insurance, as specified in Chapter 4, successfully integrates two seemingly starkly dissimilar concepts of insurance and financial option by introducing one additional entity, namely, the third entity into contemporary two-entity structure. In order to ensure the profitability of option insurer (third entity) is independent of the financial market trend, it is always ideal to seek a pair of option investors with entirely reversed future market trend, that is, a call and a put of identical strike price and expiration date (or exercise date).

In reality, the equal-sized “camps” of calls and puts with identical moneyness rarely exist in the financial market. This is mainly caused by different future expectations of various market participants and adopted trading strategies. Therefore, a business strategy of matching, as described in section 4.3, is proposed to enhance the functionality of option insurance. It allows the optimal matching between calls and puts with different moneyness. Such business strategy of matching can be mathematically classified in terms of a specific class of combinatorial optimization problems, called assignment problem (or strategic assignment problem as its applications are always related to the situations involving strategic decision making). In fact, the strategic assignment problem has a prolonged history, which was firstly employed in solving a famous Tian Ji’s horse racing problem in Warring States of ancient China. In addition to that, the applications of strategic assignment problems also involve assigning workers to jobs [101], goods to storage [102], students to universities (also known as admission problem) [103], and males to females (also known as stable marriage problem) [104].

In this chapter, instead of using conventional method, which is generally adopted by conventional digital computer to solve selected strategic assignment problem, a novel technique of biomolecular mediated computing is proposed to tackle the problem. Biomolecular mediated

²² Jian-Jun Shu, Qi-Wen Wang, and Kian-Yan Yong, “DNA-based Computing of Strategic Assignment Problems”, *Physical Review Letters*, Vol. 106, Issue. 18, Article Number: 188702, 2011.

computing is a specific computing paradigm which utilizes the artificially synthesized nucleic acids or any other analogous structures as information storing and processing substrate to tackle computational problems. Biomolecular mediated computing is sometimes termed as DNA-based computing if DNA is used as the substrate of information encoding and processing. The method has already demonstrated in solving a directed Hamiltonian path problem [37], satisfactory (SAT) problem [38], maximal clique problem [39], Chinese postman problem [42], and travel salesman problem [44].

In this chapter, a biomolecular mediated device is designed and demonstrated to solve a special strategic assignment problem. This chapter is organized as follows: it begins with a brief introduction on strategic assignment problem (section 6.1). The selected strategic assignment problem is solved by using the conventional method, which is employed by conventional silicon-based digital computer (section 6.2). The obtained result from section 6.2 can be used to compare with that is solved by using biomolecular computing approach. A specifically designed biomolecular mediated algorithm and the corresponding information encoding methodology are briefly depicted in section 6.3. Finally, the detailed experimental protocols are described in section 6.4.

6.1 Strategic Assignment Problem

In graph theory, the strategic assignment problem is defined as the determination of a maximum cardinality matching in a bipartite graph, in which each vertex in set X is matched with a non-repeatable vertex in set Y [105].

The strategic assignment problem has a prolonged history, which is closely related to the strategic applications. In the Warring States of ancient China, Sun Bin, an alleged descendant of Sun Tzu, author of the world famous military strategy book, which is known as “The Art of War”, successfully assisted Tian Ji, general of Qi, in reversing the losing trend in a horse race through manipulating the sequence of horses selected in each round of the race. This famous story of strategy application is disseminated as Tian Ji’s horse racing [106]. The story concerns about a horse race took place between two parties: Tian Ji and King Wei of Qi. There was a total of three rounds in one race; one of three available horses for each party was selected in each round, the party who won two out of three rounds became the final winner. According to the speed of the horses, each party’s horses could be further divided into three classes, namely, super, plus, and regular. Under the same classes, those horses belonging to King Wei of Qi were slightly faster than those of Tian Ji. Tian Ji’s horse racing problem is a special case of finding the maximum cardinality matching in a bipartite graph as illustrated in Figure 6 – 1.

Figure 6 – 1: *Bipartite graph of Tian Ji's horse racing*

A bipartite graph is a graph G in which the complete set of vertices is partitioned into two subsets, namely, X and Y . Every edge (i, j) in graph G has one end in vertex set X and another end in vertex set Y . A matching in graph G is a set M (stands for matching set) of edges in graph G such that no two of them have a vertex in common. The strategic assignment problem is specified as follows: given a graph G containing two sets of vertices, namely, X and Y , and edge set E , what is the maximum cardinality matching M consisting of the greatest number of edges?

For the sake of simplicity, instead of using the example of matching between different calls and puts as specified in Figure 4 – 7, a study case, as shown in Figure 6 – 2(a), is selected to demonstrate the concept. It is because the selected case is a special case, which has unequal size of vertices in these two vertex sets. The graph G totally consists of seven vertices, three of them in vertex set X , i.e., $X = \{a, b, c\}$; the remaining vertices are allocated to vertex set Y , i.e., $Y = \{1, 2, 3, 4\}$. Figure 6 – 2 (b) is the complementary graph, denoted by G' to its original bipartite graph – those edges absent in Figure 6 – 2 (a) are presented in its complementary graph G' , that is, Figure 6 – 2 (b).

Figure 6 – 2: *The strategic assignment problem and its complementary graph*

6.2 Classical Solution

The classical method of solving strategic assignment problem is to verify whether there exists an augmenting path P^{23} within a bipartite graph $G = (X, Y; E)$ with a given matching set M . If the answer is “yes”, then the matching set M is not a maximum cardinality matching, for instance, $M = \{(b, 4), (c, 2)\}$. Further modification of matching set M is compulsory, and in such case, the complete verification process has to repeat until the augmenting path does not exist in graph G . If the answer is “no”, then the matching set M is the maximum cardinality matching. The resulting matching set M is the answer to the strategic assignment problem.

The classic algorithm employed by digital computer was formally reported by Erwin Kreyszig [105]. Suppose the matching procedure initiates with a matching set of $M_1\{(b, 4), (c, 2)\}$. By employing the algorithm, the problem, as described in Figure 6 – 2 (a), can be solved by following the procedures as stated below:

1. Label exposed vertex a^{24} in vertex set X with symbol $\emptyset$.
2. Label vertex 1 with b ; Label 3 with c ; Label 4 with a .
3. Label vertex b with 4; Label vertex c with 2. The labeling procedures from step 1 to step 3 are specified in Figure 6 – 3 (a).
4. Backtrack the alternating paths P ending on an exposed vertex in entire vertex set by using the labels on the vertices. It is possible to obtain two paths as specified below:

$$P_1: 2 \rightarrow c \rightarrow 3 \text{ (} P_1 \text{ is alternating, but not augmenting)}$$

$$P_2: a \rightarrow 4 \rightarrow b \rightarrow 1 \text{ (} P_2 \text{ is alternating and augmenting)}$$
5. Augment M_1 by using the argument path P_2 , dropping vertex $(b, 4)$ from matching set M_1 and including another two edges $(a, 4), (b, 1)$ into the new matching set M_1 . The modified graph is depicted in Figure 6 – 3 (b).
6. As there is no exposed vertex in vertex set X , then the new matching set $M_1\{(a, 4), (b, 1), (c, 2)\}$ is the maximum cardinality in graph G .

By following the same procedure, the solution of selected strategic assignment problem is either $M_1\{(a, 4), (b, 1), (c, 2)\}$ or $M_2\{(a, 4), (b, 1), (c, 3)\}$. This classical idea theoretically demonstrates the procedures by which a conventional silicon-based computer solves a typical

²³ **Augmenting path:** it is a path that consists alternately of edges in matching set M and not in M . In addition to that, such path begins and ends with exposed vertices.

²⁴ **Exposed vertex:** a vertex is exposed if it is not an endpoint of any edges in a matching set M .

strategic assignment problem. Once the size of problem scaled-up, the classical algorithm becomes inefficient to handle as the required computational times increases exponentially corresponding to the problematic size. Therefore, biomolecular mediated computing (or DNA-based computing if DNA is employed as information storage and processing substrate) is introduced to solve this specific class of problem.

Figure 6 – 3: *Solution of strategic assignment problem by employing classical algorithm*

6.3 Methods

The classic algorithm, which was originally designated for a digital computer, is no longer available for biomolecular computing as the philosophy of these two computing methodology is totally different. Such that, the following algorithm for DNA-based computing is proposed to solve the strategic assignment problem:

Step 1: Generate all possible solution paths in graph G .

Step 2: Retain solution paths that begin with any vertex in set X and ended with one specific designed vertex, called *stop* vertex.

Step 3: Retain these solution paths with a predetermined length.

Step 4: Retain solution paths containing each vertex at most once.

Step 5: Determine the maximum cardinality matching set M to the problem.

Before implementing the algorithm, the very beginning procedure is to represent information in terms of DNA sequences. Such process is always referring to as *problem encoding*. In the selected case, as depicted in Figure 6 – 2 (a), there are three vertices in set X : vertex a , b , and c . Each vertex in set X is represented by 20 mer single-stranded DNA (ssDNA) with 50% CG content as described in Table 6 – 1, whereas the underlined DNA sequences represent the recognition site of corresponding restriction enzymes: *Streptomyces caespitosus* (ScaI), *Streptomyces tubercidicus* (StuI), and *Serratia marcescens* (SmaI), respectively. These restriction enzymes have different recognition sites and only effective to double-stranded DNA (dsDNA) duplex. Suppose the original double-stranded DNA (dsDNA) is in 20-base pair (bp) length. Once it is subjected to the designated restriction enzyme, the original 20 base pair (bp) dsDNA is cleaved into two equal-length dsDNA of blunt ends.

Vertex	Single-stranded DNA (ssDNA) sequence
a	5' – ATGCCGTAGTACTAAGCAGC – 3'
b	5' – TATCGACAGGCCTATCGATC – 3'
c	5' – TATTGTCCCCGGGATCTAT – 3'

Table 6 – 1: DNA sequence design of vertices in set X

Similarly, each vertex in set Y , vertex 1, 2, 3, and 4, is represented by 20 mer ssDNA with 50% CG content as specified in Table 6 – 2. By following the similar design scheme, the underlined DNA sequences are the recognition site corresponding to four different types of restriction enzymes *Proteus vulgaris* (PvuII), *Escherichia coli* (EcoRV), *Haemophilus parainfluenzae* (HpaI), and *Pseudomonas maltophilia* (PmlI), respectively.

Vertex	Single-stranded DNA (ssDNA) sequence
1	5' - AACGTAG <u>CAGCTG</u> TTACGCT - 3'
2	5' - TCTCTGAGATATCCCGGCTA - 3'
3	5' - CGCCTGTG <u>TTAACG</u> CGTAAT - 3'
4	5' - GTACTTGCACGTGTAACGTG - 3'

Table 6 – 2: *DNA sequence design of vertices in set Y*

There are totally two classes of edges, all these edges are directed edges²⁵. The first class of directed edges is utilized to connect vertices in set X to vertices in set Y , which is the edges appear in graph G as specified in Figure 6 – 2 (a). These directed edges are denoted by (i, j) , where $i \in \{a, b, c\}$ and $j \in \{1, 2, 3, 4\}$. For each directed edge (i, j) , its corresponding DNA sequence contains two sections – the first section, 10 mer single-stranded DNA (ssDNA) is the complementary strand of rear 10 mer ssDNA of vertex i in set X ; the second section, 10 mer ssDNA is the complementary strand of former 10 mer ssDNA of vertex j in set Y . Such encoding scheme is demonstrated by connecting vertex a in set X to vertex 4 in set Y by using directed edge $(a, 4)$, as shown in Figure 6 – 4(a). All these corresponding DNA sequences for directed edges (i, j) are listed in Table 6 – 3.

Figure 6 – 4: *Demonstration of DNA encoding protocol*

²⁵ **Directed edge**: the edge assigned with a specified direction.

Edge	Single-stranded DNA (ssDNA) sequence
(a, 4)	3' - TGATTCGTCGCATGAACGTG - 5'
(b, 1)	3' - GGATAGCTAGTTGCATCGTC - 5'
(b, 4)	3' - GGATAGCTAGCATGAACGTG - 5'
(c, 2)	3' - CCCCTAGATAAGAGACTCTT - 5'
(c, 3)	3' - CCCCTAGATAGCGGACACAA - 5'

Table 6 – 3: *DNA sequence design of directed edges (i, j)*

The second class of the directed edges, denoted by (j, i) , is the edges appear in the complementary graph G' as described in Figure 6 – 2 (b). These directed edges are used to connect vertices in set Y to vertices in set X through a process called backtracking. Therefore, this special class of edges is named as *backtracking edges*. The objective of backtracking edges is to promote the creation of the complete solution path, which is required by biomolecular computing algorithm. Similarly, for each backtracking edge (j, i) , it contains two section – the first section, 10 mer ssDNA is the complementary strand of rear 10 mer ssDNA of vertex j in set Y ; the second section, 10 mer ssDNA is the complementary strand of former 10 mer ssDNA of vertex i in set X . The backtracking edge sequences are listed in Table 6 – 4.

Edge	Single-stranded DNA (ssDNA) sequence
(1, a)	3' - GACAATGCGATACGGCATCA - 5'
(1, c)	3' - GACAATGCGAATAACAGGGG - 5'
(2, a)	3' - TAGGGCCGAGTACGGCATCA - 5'
(2, b)	3' - TAGGGCCGAGATAGCTGTCC - 5'
(3, a)	3' - TTGCGCATTATACGGCATCA - 5'
(3, b)	3' - TTGCGCATTAATAGCTGTCC - 5'
(4, c)	3' - CACATTGCACATAACAGGGG - 5'

Table 6 – 4: *DNA sequence design of directed backtracking edges (j, i)*

In addition to all the vertices and edges as described above, a special class of vertices, namely, *stop vertex*, is required by the algorithm. This class of vertices is employed to regulate the output solution path, *i.e.*, no matter the solution path begins with any vertex in set X , it always terminates at a specific class of vertices called *stop vertex*. Any unqualified solution paths, which do not end up with any one of stop vertices are likely to be rejected from the solution at the end of step 2. There are totally four different stop vertices corresponding to four vertices in set Y , namely, *stop 1*, *stop 2*, *stop 3*, and *stop 4*. Each stop vertex is a 10 bp dsDNA duplex with an overhanging end of

10 mer. The overhanging ends are the complementary strands of rear 10 mer ssDNA of vertex j in set Y . These entire stop vertices are summarized in Table 6 – 5.

Vertex	Single-stranded DNA (ssDNA) sequence
<i>Stop 1</i>	5' – ATGCGTACTG – 3' 3' – GACAATGCGATACGCATGAC – 5'
<i>Stop 2</i>	5' – ATGCGTACTG – 3' 3' – AAGGGCCGCGTACGCATGAC – 5'
<i>Stop 3</i>	5' – ATGCGTACTG – 3' 3' – TTGCGCATTATACGCATGAC – 5'
<i>Stop 4</i>	5' – ATGCGTACTG – 3' 3' – CACATTGCACTACGCATGAC – 5'

Table 6 – 5: *DNA sequence design of stop vertices*

6.4 Implementation

Step 1: After all DNA strands, single-stranded DNA (ssDNA) and double-stranded DNA (dsDNA), as stated above are artificially synthesized in separate test tubes according to their design protocols. For each test tube, it contains ssDNA and dsDNA duplex with one-sided sticky end of approximately $50 \mu M$. A small amount of the solution, usually $1 \mu l$, is taken from each individual test tube and mixed in a new test tube. The resultant $23 \mu l$ mixture is subjected to a process known as *hybridization*²⁶. This is the process where ssDNA and partial dsDNA duplex readily anneal among each other through hydrogen bonds in the formation of the solution path by following Watson-Crick complementary principle. After hybridization, $2.4 \mu l$ of $10 \times$ ligation buffer is added to the mixture and left at room temperature for another 15 minutes. After that, $2 \mu l$ of NEB kinase is added into the solution and incubated for 1 hour at $37^\circ C$. The phosphate contained within NEB kinase is the essential chemical component to “fix” 5’ end of adjacent DNA strands, which is necessary in the subsequent ligation process.

After phosphorylation, $2 \mu l$ of NEB T4 DNA ligase is added into mixed solution. The DNA ligase is going to establish the covalent bond for these annealed DNA strands. Finally, the mixture is incubated at $25^\circ C$ for approximately 20 minutes to complete the ligation process. It is believed that all possible paths are generated in the solution pool as for each edge in graph G , there are approximately 6×10^{13} copies of associated ssDNA. For the sake of convenience, all possible paths can be expressed in terms of “*vertex* $\rightarrow$ *vertex* $\rightarrow \dots \rightarrow$ *vertex*”. For instance, the solution path of “ $a \rightarrow 4 \rightarrow c \rightarrow 2 \rightarrow b \rightarrow 1 \rightarrow stop\ 1$ ” can be expressed in terms of DNA sequence as shown in Figure 6 – 5. In fact, all solution paths are ended up with a duplex form similar to that of Figure 6 – 5. However, only one side (or strand) of DNA duplex, which contains the information of related vertices, is of interest.

²⁶ **Hybridization:** The single-stranded DNA is likely to anneal to its complementary single-stranded DNA in the formation of double-stranded DNA duplex by following Watson-Crick complementary principle.

Figure 6 – 5: *A typical example of possible paths generated at the end of ligation process*

By employing such encoding scheme, all possible solution paths generated during the hybridization process are restricted to alternating paths²⁷, which consist alternatively of edges in the graph G of strategic assignment problem, as shown in Figure 6 – 2 (a), and its complementary graph G' , as shown in Figure 6 – 2 (b). It simply states that all possible paths generated through hybridization are unlikely to contain any two adjacent vertices in the same vertex set, for instance, $a \rightarrow 4 \rightarrow 2$ or $a \rightarrow b \rightarrow 1$. Additionally, in certain perspective, the encoding scheme minimizes the chance of any paths pass through any vertices in graph G for more than once.

Step 2: To implement step 2, the solution obtained from step 1 is subjected to a process called *polymerase chain reaction* (PCR)²⁸. Prior to the process of PCR, the resultant solution is subjected to the purification process. The objective of solution purification is aiming to remove the single-stranded DNA (ssDNA), which is not annealed to its complementary strand, short DNA duplex (less than 50 bp), enzymes and other possible impurities from step 1. The QIAquick® PCR purification kit²⁹ is employed in purification process.

A total 147 μl of PB buffer, 5 times of the solution volume (29.4 μl), is added into the solution tube from step 1. It takes three steps to purify the mixture, namely, bind, wash, and elute. After

²⁷ **Alternating path:** all vertices in graph G are connected by edges in original graph and backtracking edges in its complementary graph.

²⁸ **Polymerase chain reaction:** a frequently used laboratory technique in molecular biology to amplify target DNA strands by manipulating the primers.

²⁹ **QIAquick® PCR purification kit:** a commercialized PCR purification kit provided by QIAGEN®, it consists of spin columns, buffers, and collection tubes. The objective of the purification kit is to purify both single-stranded and double-stranded DNA fragments from PCR and any other enzymatic reactions.

purification, the resultant solution is amplified by PCR using 10 mer complementary strand of 10 mer oligonucleotide of any vertices in set X and 10 mer complementary strand of stop vertices, which is fixed in its sequence ($3' - \text{TACGCATGAC} - 5'$), as primers of PCR. In addition to the primers, the QIAGEN fast cycling PCR kit³⁰ is added into the solution prior to the process of PCR. A total $14 \mu\text{l}$ of reagents are prepared prior to PCR, including $10 \mu\text{l}$ of PCR master mix³¹, $2 \mu\text{l}$ of Q solution³². Besides the reagent, $1 \mu\text{l}$ of each vertex primers in set X , $3 \mu\text{l}$ stop vertex primer are prepared and added into the solution. The resultant mixture, together with $2 \mu\text{l}$ of purified DNA solution from previous step, is added into a new test tube, leading to a mixture of $20 \mu\text{l}$ prior to PCR. The following PCR protocol is followed during the experiment:

1. Increase solution temperature to 95°C and maintain at that temperature for 5 minutes.
2. Increase the temperature to 96°C and maintain at that temperature for 5 seconds to allow the completion of denaturation process.
3. Gradually reduce solution temperature to 45°C and retain at that temperature for 5 seconds to promote the annealing of DNA primers.
4. Increase solution temperature to 68°C and maintain at that temperature for another 5 seconds, which allows the extension of primers.
5. Repeat step 2 to step 4 of PCR protocol for another 29 cycles.
6. Increase the temperature to 72°C and maintain at that temperature for another 1 minute.
7. Gradually reduce the temperature to 25°C (room temperature).

As the consequence of polymerase chain reaction (PCR), only those solution paths that begin with any vertices in set X and ended with any stop vertices is amplified. Those incomplete paths³³, for instance, " $b \rightarrow 1 \rightarrow c \rightarrow 3 \rightarrow a$ ", will be eliminated in the subsequent procedure, step 3.

Step 3: Product of step 2 is sieved in the process of native polyacrylamide gel electrophoresis, abbreviated PAGE. According to the working principle of gel electrophoresis, the migration

³⁰ **QIAGEN fast cycling PCR kit:** a commercialized fast cycling PCR kit provided by QIAGEN®, it is designated for hot-start RCR amplification, *i.e.*, the polymerase is necessary to be activated at the beginning of PCR by increasing the solution temperature. The fast cycling PCR kit consists of ready-to-use master mix, cycling dye, Q solution and RNase-free water.

³¹ **Master Mix:** an optimized solution which is used for fast-cycling PCR.

³² **Q solution:** an optimized solution which is dedicated in facilitating the amplification of GC rich templates.

³³ **Incomplete paths:** the paths which either begin with vertices in set Y or end up without reaching any stop vertices.

distance of linear DNA towards the anode end of gel slab, as a result of electrophoresis, is inversely proportional to the length of corresponding DNA molecules.

There are totally three solution slots (or wells) are built on the cathode side of the gel slab – the first well labeled with “O”, which is used to contain the solution obtained from step 2; the second well labeled with “F” is used to contain the DNA molecules with known length in the generation of reference path. In this case, the desired DNA molecule should be exactly 130 bp in length (i.e., the desired path must contain exactly the six-vertex sequence, each of 20 bp, together with a stop vertex of 10bp); the last well labeled with “L” is used to contain the DNA ladder of 10 bp resolution³⁴. The entire process takes approximately one and half hours at constant power supply of 18 W, and with variable voltage and current at approximately 560 V and 32 mA, respectively. After the completion of gel electrophoresis process, the gel slab is submerged into the staining chemical. After that, the resultant DNA bands can be observed once the gel slab is subjected to the ultraviolet (UV) light. The solution in well “O”, which has exactly the same length of DNA molecule in well “F”, is collected from the gel slab. In other words, any path of DNA duplex having a length either longer or shorter than 130 bp is discarded. The entire collected solution from gel is subjecting to the identical purification process, which is used prior to PCR, to remove any possible impurities from the solution.

Step 4: The selected restriction enzymes are employed to eliminate those paths pass through any vertices in graph G for more than once. This can be achieved by cleaving these paths containing more than one restriction site via adding corresponding restriction enzymes, followed by using gel electrophoresis to sieve the linear DNA duplexes according to their length.

Before going into further procedures, it is worth to take note that the most important information obtained at the end of the experiment is the maximum cardinality matching set expressed in terms of DNA sequences, rather than the exact DNA sequence itself. For instance, two paths “ $a \rightarrow 4 \rightarrow c \rightarrow 2 \rightarrow b \rightarrow 1$ ” and “ $b \rightarrow 1 \rightarrow a \rightarrow 4 \rightarrow c \rightarrow 2$ ” have different orders. However, both of them deliver the identical information, that is, the resulting matching set is likely to be $M[(a, 4), (b, 1), (c, 2)]$. Suppose all the solution paths start from vertex a and ended up with any one of the stop vertices are isolated for the analysis. It is possible to express all possible paths

³⁴ **DNA Ladder:** it is always referred to as molecular weight size maker. DNA ladder is generally used in gel electrophoresis to identify the approximate size of DNA molecules in other solution slots of gel slab. It is based upon the principle that the molecular weight of DNA fragments is inversely proportional to the migration rate passing through the gel matrix, a typical porous medium.

that start from vertex a and terminates in any one of stop vertices in terms of a tree diagram, as shown in Figure 6 – 6.

Figure 6 – 6: *All paths starts from vertex a constitutes a tree diagram*

In this case, five restriction enzymes are required – ScaI (for vertex a), PvuII (for vertex 1), EcoRV (for vertex 2), HpaI (for vertex 3), and PmlI (for vertex 4). The objective is to eliminate those paths passing through any vertices in graph G for more than once.

First of all, the solution, 10 μ l NEBuffer, which contains approximately 0.2 μ g restriction enzyme ScaI is added into the purified solution obtained from step 3. The entire reaction takes an hour to complete at room temperature. After that, the solution is split into four separate test tubes. These four test tubes are added with approximately 0.25 μ g of restriction enzymes PvuII, EcoRV, HpaI, and PmlI, separately, in the 10 μ l corresponding buffer, NEBuffer 3.1 (for PvuII), NEBuffer 3.1 (for EcoRV), CutSmart Buffer (for HpaI), and CutSmart Buffer (for PmlI). And these tubes are labeled corresponding to the added restriction enzymes. Another 1 hour is allowed for completion of digestion.

After the completion of digestion, similar to step 3, a gel slab containing five solution slots are created – wells “1” to “4” are used to contain the solution obtained from the restriction enzyme digestion process – “1”: ScaI + PvuII; “2”: ScaI + EcoRV; “3”: ScaI + HpaI; “4” ScaI + PmlI, and well “5” contains DNA molecules with known length of 100bp and DNA ladder in 10bp resolution. At the end of the gel electrophoresis, as predicted by Figure 6 – 7, only the solution in any line of gel slab with the same migration length of 100 bp as in the calibration line (well “5”) is retained as the result to the strategic assignment problem.

Figure 6 – 7: *Estimated pattern of DNA migration as a result of electrophoresis*

It is necessary to understand how these procedures work in step 4. First of all, the restriction enzyme ScaI is added into the solution. The restriction enzyme ScaI starts to search for its recognition site of corresponding dsDNA sequences (5' -AGTACT-3') and cleave the duplex into two equal-length double-stranded DNA (dsDNA), which is schematically expressed in Figure 6 – 8.

Figure 6 – 8: *Restriction enzyme ScaI cleaves dsDNA into two duplexes*

By following the design scheme, only DNA sequences used to encode vertex a contains recognition site for ScaI. Therefore, the paths contain more than one vertex a , for instance, “ $a \rightarrow 4 \rightarrow c \rightarrow 2 \rightarrow a \rightarrow 4 \rightarrow \text{Stop } 4$ ” and “ $a \rightarrow 4 \rightarrow c \rightarrow 3 \rightarrow a \rightarrow 4 \rightarrow \text{Stop } 4$ ” are completely eliminated from the solution pool. The elimination process takes place when the restriction enzyme ScaI is added into the solution – as these two paths contain more than one restriction site of restriction endonuclease ScaI, they are cleaved into three DNA duplexes in the length of 10 bp, 40bp, and 80bp. None of these duplexes satisfy the length requirement as required by the design protocol, which is exactly 100bp. At the end of gel electrophoresis, all these unqualified paths expressed in terms of DNA sequences are completely removed.

After that, the solution is equally separated into four individual test tubes. The other four restriction enzymes, PvuII (vertex 1), EcoRV (vertex 2), HpaI (vertex 3), and PmlI (vertex 4), are added into these test tubes, each test tube corresponds to one specific restriction enzyme. At the end of digestion process, all these paths, such as “ $b \rightarrow 1 \rightarrow c \rightarrow 2 \rightarrow b \rightarrow 1$ ” (eliminated by PvuII), “ $c \rightarrow 2 \rightarrow b \rightarrow 4 \rightarrow c \rightarrow 2$ ” (eliminated by EcoRV), “ $c \rightarrow 3 \rightarrow a \rightarrow 4 \rightarrow c \rightarrow 3$ ” (eliminated by HpaI), and “ $a \rightarrow 4 \rightarrow c \rightarrow 3 \rightarrow a \rightarrow 4$ ” (eliminated by PmlI) are completely removed from the solution at the end of gel electrophoresis process. Only the paths “ $a \rightarrow 4 \rightarrow c \rightarrow 2 \rightarrow b \rightarrow 1 \rightarrow \text{Stop } 1$ ” and “ $a \rightarrow 4 \rightarrow c \rightarrow 3 \rightarrow b \rightarrow 1 \rightarrow \text{Stop } 1$ ” in well “1” satisfy the length requirement and retained as the solution to the strategic assignment problem. Although half of the sequences of vertices a and 1 are cleaved (and subsequently filtered at the end of gel electrophoresis) as a result of restriction enzyme ScaI and PmlI, it does not affect the decipherment of DNA sequences to the solution of strategic assignment problem.

Step 5: The target solution is collected from the gel slab after the process of electrophoresis. DNA sequencing method is employed to determine the actual sequence of collected sample. The sequencing can be done by either using the traditional Sanger’s method or more advanced approach like cyclic-array sequencing from single polymerase molecules. Consequently, the DNA sequence can be deciphered into the answer of the strategic assignment problem, which is matching sets $M_1\{(a, 4), (c, 2), (b, 1)\}$ and $M_2\{(a, 4), (c, 3), (b, 1)\}$.

6.5 Conclusion

The DNA-based computing algorithm, as proposed in section 6.3, is believed to be much more efficient in solving problems like strategic assignment problem, whereas the method demands much more searching ability, than any existing conventional digital computing methodology. The specifically designed algorithm is believed to maximize the advantage of hybridization, that is, single-stranded DNA (ssDNA) in solution is non-interruptedly searching for its complementary ssDNA and anneal to the searching result in the formation of much stable double-stranded DNA (dsDNA) duplex in the solution. There is roughly 10^{14} hybridization operations can be achieved within one second. Therefore, the biomolecular computing is extremely efficient in handling such problems and not easily penalized by the problematic size.

The proposed DNA-based computing algorithm, however, has one limit, that is, it is highly rely on the formation of the solution path. There must be sufficient number of edges in graph G , as shown in Figure 6 – 2 (a), and backtracking edges in its complementary graph G' , as specified in Figure 6 – 2 (b). Suppose the total number of vertices in the vertex set X is m , and the total number of vertices in vertex set Y is n . The maximum number of edges is therefore becomes mn . If the number of edges in the original graph is e_{ij} , then the number of edges in its complementary graph is e_{ji} , which equals to $mn - e_{ij}$. The proposed DNA-based algorithm is valid if and only if the number of backtracking edge satisfies the following equation, equation 6.1.

$$\min(m, n) \leq e_{ji} \leq mn - \min(m, n) \quad (6.1)$$

However, it can be a very rare case if the condition as specified above is not fulfilled. It is possible to explain the concept by using the same study case as specified in Figure 6 – 2 (a). The case consists of three vertices in set X (i.e. $m = 3$) and four vertices in set Y (i.e. $n = 4$). Therefore, in order to use the algorithm designated for biomolecular computing, the number of backtracking edges has to be in the range of 3 [= $\max(m, n)$] to 9 [= $mn - \max(m, n)$].

Suppose the number of backtracking edges is less than $\min(m, n)$, for instance, $e_{ij} = 10$, as indicated in Figure 6 – 9 (a). In the case, it has excessive edges in the graph G . Therefore, the backtracking edge is likely to be $e_{ji} = 2$, which is insufficient in the formation of solution path. However, as the excessive number of edges in the original path, each vertex in set X can be

assigned to any vertices in set Y . In such cases, it is not necessary to design a specific algorithm to solve the problem.

On the contrary, suppose the number of backtracking edges is more than $mn - \min(m, n)$, for instance, $e_{ji} = 10$, as indicated in Figure 6 – 9 (b). In this case, it has insufficient edges in the graph G , which is incapable to form the solution path. However, due to the insufficient edges in graph G , the maximum cardinality matching automatically becomes the existing edges in graph G , which also lose the significant meaning to establish any algorithm to such problems.

Figure 6 – 9: *Excessive edges versus insufficient edges in strategic assignment problem*

In the majority of the strategic assignment problems, there is always sufficient number of edges in graph G and backtracking edges in its corresponding complementary graph G' to form the solution path as required by proposed algorithm. In addition to that, the proposed DNA-based computing algorithm has many significant advantages over the existing methodology. First of all, it provides the instant result instead of manually verifying the augmenting path(s), which is required by the algorithm designed for digital computing. Secondly, it is capable of scaling up the problematic size as the entire process requires least amount of human involvement. Lastly, it is relatively simple and practical in the designated laboratory techniques. The proposed method opens up the possibility of extending the DNA-based computing technique in analyzing the situations, which requires the involvement of strategic decisions.

Chapter 7

Biomolecular Mediated Optimal Route Selection Apparatus

The massive parallelism inherent in biomolecular mediated computing has potential to yield a tremendous speed-up over the conventional silicon-based computer when monolithic parallel computing ability is demanded by specific computational algorithm [37]. This is mainly caused by nature that single-stranded DNA (ssDNA) is likely to continuous search and anneals to its complementary ssDNA, leading to the formation of a relatively stable duplex structure, called double-stranded DNA (dsDNA). In certain perspective, this specific biomolecular property is similar to the hypothetical computing device, called Turing machine [107 – 110], which was invented by Alan Turning in 1936.

In previous chapter, Chapter 6, a novel-based method of biomolecular mediated device is firstly proposed. It is theoretically demonstrated to tackle a specific class of combinatorial optimization problem, termed as strategic assignment problem [111]. In reality, the technique of biomolecular mediated device is capable of handling much more complicated tasks like constructing a fully functional *in-vitro* biomolecular mediated device. The developed biomolecular mediated device is able to store information in its unique “database”. Based upon the information manually assigned by user, the device is capable of tracing information from “database”, processing the information, and finally displaying the obtained result.

A biomolecular mediated optimal route selection apparatus is designed and experimentally demonstrated in this chapter, chapter 7. Optimal route selection apparatus is a functional unit embedded in the contemporary Global Positioning System (GPS) [112]. By manually assigning the current location and final destination to the information processing unit, it recalls information stored in its database, and returns with an optimal path between these two manually assigned physical locations.

This chapter begins with an introduction on optimal route selection apparatus in section 7.1. Similar to the arrangement of previous chapter, the selected case study is firstly solved by using

the conventional method, which is frequently employed by conventional silicon-based digital computer, as specified in section 7.2. An algorithm, which is designed specifically for biomolecular mediated computing, and the corresponding information encoding methodology is provided in section 7.3. The entire experiment procedures are described in section 7.4. In the end, the experimental result is presented in section 7.5.

7.1 Optimal Route Selection Apparatus

Optimal route selection apparatus behaves as a functional unit embedded in contemporary Global Positioning System (GPS) navigator. By manually assigning any two physical locations, present location and final destination, to the apparatus, it automatically routes an optimal path, or a shortest path, between selected locations, based on the information pre-stored within its database.

Figure 7 – 1: *Optimal route selection problem and its abstract graph G*

In order to demonstrate the concept, an arbitrary map in GPS navigation system is selected as the study case. In the map, as shown in Figure 7 – 1, there are totally six pre-stored physical locations, namely, home (vertex 1), gas station (vertex 2), shopping mall (vertex 3), hospital (vertex 4), restaurant (vertex 5), and company (vertex 6). All these locations are connected by roads, or bipaths³⁵. For the sake of simplicity, the key information of the map can be represented by using

³⁵ **Bipath:** bipath simply states there is no specific directional restriction on the corresponding edge.

an abstract graph G , as shown in the right hand side of Figure 7 – 1, which contains vertices and non-directed edges. Each vertex i and weighted edge (i, j) in graph G represents an associated physical location and its connecting bipath in the original map, respectively, where $i, j \in \{1, 2, 3, 4, 5, 6\}$. The numbers placed right on top of the edges are the distance (or weights) of the corresponding edges. Such that, the entire physical map can be converted into a mathematical problem: given a graph G , made up of vertices and weighted non-directed edges, what is the shortest path between any two user-specified vertices?

7.2 Classical Solution

For traditional electronic-based digital computer, two classic algorithms have been proposed to tackle the problem of optimal path, known as Moore's algorithm [113] and Dijkstra's algorithm [90].

Moore's algorithm is only capable of solving a special class of shortest path problem, as it demands all edges in graph G must be in equal lengths. On the contrary, Dijkstra's algorithm is much more suitable for general cases. In this section, section 7.2, Dijkstra's algorithm is theoretically demonstrated to solve a typical optimal path problem, as shown in Figure 7 – 1, by following the procedures of a conventional digital computing.

Dijkstra's algorithm is a specific combination of labeling and comparing procedure. In every cycle of computing, each vertex is assigned and controlled by using a label. There are totally two types of labels, called permanent label, denoted by PL , and temporary label, denoted by TL .

Suppose it is desired by the user of optimal route selection apparatus to determine the most optimal path(s) traveling from two distinct locations, home (vertex 1) and company (vertex 6) to the same location, which is the hospital (vertex 4).

The procedures on how the classic method, Dijkstra's algorithm, is employed to determine the most optimal path traveling from home (vertex 1) to hospital (vertex 4), as shown in Figure 7 – 1, are specified below:

1. $L_1 = 0; \widetilde{L}_2 = 46, \widetilde{L}_3 = \infty, \widetilde{L}_4 = \infty, \widetilde{L}_5 = 32, \widetilde{L}_6 = \infty$
 $PL = \{1\}, \quad TL = \{2, 3, 4, 5, 6\}$
2. $L_5 = \min\{\widetilde{L}_2, \widetilde{L}_3, \widetilde{L}_4, \widetilde{L}_5, \widetilde{L}_6\} = 32, k = 5$
 $PL = \{1, 5\}, \quad TL = \{2, 3, 4, 6\}$
3. $\widetilde{L}_2 = \min\{46, 32 + \infty\} = 46, \widetilde{L}_3 = \min\{\infty, 32 + 38\} = 70, \widetilde{L}_4 = \min\{\infty, 32 + 30\} = 62, \widetilde{L}_6 = \min\{\infty, 32 + \infty\} = \infty$
4. $L_2 = \min\{\widetilde{L}_2, \widetilde{L}_3, \widetilde{L}_4, \widetilde{L}_6\} = 46, k = 2$
 $PL = \{1, 2, 5\}, \quad TL = \{3, 4, 6\}$
5. $\widetilde{L}_3 = \min\{70, 46 + 42\} = 70, \widetilde{L}_4 = \min\{62, 46 + 32\} = 62, \widetilde{L}_6 = \min\{\infty, 46 + 38\} = 84$
6. $L_4 = \min\{\widetilde{L}_3, \widetilde{L}_4, \widetilde{L}_6\} = 62, k = 4$

- $$P\mathcal{L} = \{1, 2, 4, 5\}, \quad T\mathcal{L} = \{3, 6\}$$
7. $\widetilde{L}_3 = \min\{70, 62 + 36\} = 70, \quad \widetilde{L}_6 = \min\{84, 62 + \infty\} = 84$
 8. $L_3 = \min\{\widetilde{L}_3, \widetilde{L}_6\} = 70, \quad k = 3$
 $P\mathcal{L} = \{1, 2, 3, 4, 5\}, \quad T\mathcal{L} = \{6\}$
 9. $\widetilde{L}_6 = \min\{84, 74 + \infty\} = 84$
 10. $L_6 = 84, \quad k = 6$
 $P\mathcal{L} = \{1, 2, 3, 4, 5, 6\}, \quad T\mathcal{L} = \emptyset$

By simply referring to various values of L , as specified above, it is possible to obtain the length of shortest paths from starting vertex 1 to the other two terminating vertices in graph G . As indicated above, $L_4 = 62$ simply specifies the length of shortest path starting from vertex 1 to vertex 4 is 62 units of length. By referring to the information, it is possible to determine the shortest path from vertex 1 to vertex 4 is likely to be “*vertex 1* $\rightarrow$ *vertex 5* $\rightarrow$ *vertex 4*”. Moreover, it is possible to convert the original graph, as shown in Figure 7 – 1, into a new graph, known as modified graph, as specified in Figure 7 – 2 (a), with respect to the results obtained at different stages of computing of classic algorithm. The resulting graph is a typical spanning tree. Based upon the modified graph G , as specified in Figure 7 – 2 (a), it is simple to obtain all shortest paths from vertex 1 to any other vertices as depicted from Figure 7 – 2 (b) to Figure 7 – 2 (f).

Figure 7 – 2: *Modified graph G and shortest paths from vertex 1 to other vertices in G*

Similarly, by employing the same method, it is able to determine the most optimal path(s) traveling from company (vertex 6) to hospital (vertex 4), as shown in Figure 7 – 1. The procedures are detailed below:

1. $L_6 = 0; \widetilde{L}_1 = \infty, \widetilde{L}_2 = 38, \widetilde{L}_3 = \infty, \widetilde{L}_4 = \infty, \widetilde{L}_5 = \infty$
 $PL = \{6\}, \quad TL = \{1, 2, 3, 4, 5\}$
2. $L_5 = \min\{\widetilde{L}_1, \widetilde{L}_2, \widetilde{L}_3, \widetilde{L}_4, \widetilde{L}_5\} = 38, k = 2$
 $PL = \{2, 6\}, \quad TL = \{1, 3, 4, 5\}$
3. $\widetilde{L}_1 = \min\{\infty, 38 + 46\} = 84, \widetilde{L}_3 = \min\{\infty, 38 + 42\} = 80, \widetilde{L}_4 = \min\{\infty, 38 + 32\} = 70, \widetilde{L}_5 = \min\{\infty, 38 + \infty\} = \infty$
4. $L_4 = \min\{\widetilde{L}_1, \widetilde{L}_3, \widetilde{L}_4, \widetilde{L}_5\} = 70, k = 4$
 $PL = \{2, 4, 6\}, \quad TL = \{1, 3, 5\}$
5. $\widetilde{L}_1 = \min\{84, 70 + \infty\} = 84, \widetilde{L}_3 = \min\{80, 70 + 36\} = 80, \widetilde{L}_5 = \min\{\infty, 70 + 30\} = 100$
6. $L_3 = \min\{\widetilde{L}_1, \widetilde{L}_3, \widetilde{L}_5\} = 80, k = 3$
 $PL = \{2, 3, 4, 6\}, \quad TL = \{1, 5\}$
7. $\widetilde{L}_1 = \min\{84, 80 + \infty\} = 84, \widetilde{L}_5 = \min\{100, 80 + 38\} = 100$
8. $L_1 = \min\{\widetilde{L}_1, \widetilde{L}_5\} = 84, k = 1$
 $PL = \{1, 2, 3, 4, 6\}, \quad TL = \{5\}$
9. $\widetilde{L}_5 = \min\{100, 84 + 32\} = 100$
10. $L_5 = 100, k = 5$
 $PL = \{1, 2, 3, 4, 5, 6\}, \quad TL = \emptyset$

Identically, $L_4 = 70$, as indicated above, specifies the length of shortest path starting from vertex 6 to vertex 4 is 70 units in length. By referring to the information, it is possible to determine the shortest path from vertex 6 to vertex 4 is likely to be “*vertex 6* → *vertex 2* → *vertex 4*”. Similar to the previous case, it is possible to convert the original graph into a modified graph, as shown in Figure 7 – 3 (a), by simply retaining all edges appeared at different stages of computing by using classic algorithm. At the same time, the solution of optimal paths from vertex 6 to any other vertices are specified from Figure 7 – 3 (b) to Figure 7 – 3 (f).

In the following experiment, it is going to testify the hypothesis that the biomolecular mediated computing, or DNA mediated computing, is likely to deliver the identical result as that is

theoretically predicted by a classic algorithm, which is generally adopted by conventional digital computer.

Figure 7 – 3: *Modified graph G and shortest paths from vertex 6 to other vertices in G*

7.3 Methods

DNA-based computing is intrinsically different from conventional electronic-based digital computing. Rather than storing information in terms of binary numbers, 0 or 1, the data for DNA-based computing is represented by various nitrogenous base sequences within the “building blocks” of DNA molecules, known as nucleotide. There are totally four different variations of bases, known as adenine (A), thymine (T), guanine (G) and cytosine (C). The capacity of DNA molecules is not only restricted to act as memory, in reality, they can also be utilized as operand themselves. Two individual bases can be jointed in the formation of a base pair through hydrogen bonding, what has become known as Watson-Crick complementary principle: A always pairs with T in formation of two hydrogen bonds; G always pairs with C in formation of three hydrogen bonds.

Figure 7 – 4: *Process flow chart of DNA mediated optimal route selection apparatus*

DNA-based optimal route selection apparatus is composed of six stages, which can be visually expressed in Figure 7 – 4. The constituent elements, vertices and weighted non-directed edges, within the graph G are firstly converted into DNA sequences in terms of A, T, G and C inside the *problem encoder*. Based on the information obtained from the problem encoder, DNA solution required for subsequent processes is prepared in *DNA solution bay*. The *mixing controller* chooses appropriate DNA solution to be mixed in generating all possible DNA structures that represent the potential solution, the “database”, to the optimal path problem. In the perspective of mathematics, all non-restricted random walks are initiated within the graph G upon the completion of hybridization. The obtained solution is subsequently purified in the *solution purifier* to isolate DNA solution from oligonucleotides resulting from incomplete hybridization, T4 ligase and other possible impurities from previous stage. The purified DNA solution is subsequently amplified by using *polymerase chain reaction (PCR) amplifier*. Based upon the input provided by user of DNA-based optimal route selection apparatus, present location and final destination, the corresponding DNA primers are manually selected during the PCR process. At the end of solution amplification stage, the remaining solution contains only the DNA strands that represent the solution path begun with specified location and terminated at specified destination. The final mixture is subjected to native polyacrylamide *gel electrophoresis*. Native polyacrylamide gel is acting as a sieving material, which separates the linear DNA strands according to the length of these DNA fragments. Only the DNA solution that represents the optimal (shortest) path is retained at the end of native polyacrylamide gel electrophoresis. In the final stage, the length of optimal DNA duplex is deciphered into the solution of optimal path problem.

7.4 Implementation

Problem encoder – For electronic-based computer, the very beginning stage is to digitize information in terms of binary expressions. Analogous to digital computing, the first step for DNA mediated computing is to encode information in terms of a combination of DNA sequences: A, T, G, and C.

Each vertex in the graph G , as indicated in Figure 7 – 1, is encoded with 20-mer single-stranded DNA (ssDNA). To avoid undesired hybridization during mixing stage, the generated DNA strands must have 50% CG content and identical melting temperature, $T_m = 70^\circ\text{C}$. Based upon the specified rule and the sequence designing methodology as mentioned in [114], the encoded DNA sequences with respect to their corresponding vertices are listed in Table 7 – 1.

Vertex	Single-stranded DNA sequence (5' to 3')
1	CCGTGTCTACAACAGAAGGA
2	CCCTCATTTGACGAGGAATG
3	TGGTTAGACGCAGAGAGTTC
4	AGGCACACGATTATGGACAG
5	CGAATTTAGCACCGCATGTG
6	GGGGTCTTCAACTATTGTCC

Table 7 – 1: *Vertices and corresponding single-stranded DNA (ssDNA) sequences*

As described in graph theory, the edges within graph G are used to connect two adjacent vertices. The entire graph G is a typical bigraph, that is, there is no specific restriction on the direction of these edges. Therefore, it is likely to comprise sixteen assorted DNA structures to symbolize eight corresponding edges in both directions. The structures used to encode the edges are specifically designed double-stranded DNA (dsDNA) duplex with two overhangs, *i.e.*, exposed single-stranded DNA. Each structure, as depicted in Figure 7 – 5, contains exactly three segments, namely, Segment 1 (former 10-mer exposed single-stranded DNA), Segment 2 (double-stranded DNA duplex), and Segment 3 (rear 10-mer exposed single-stranded DNA). Segment 1 and Segment 3 are employed to connect from vertex i to vertex j , whereas Segment 1 is complementary to rear 10-mer single-stranded DNA (ssDNA) that represents vertex i and Segment 2 is complementary to former 10-mer single-stranded DNA (ssDNA) that represents vertex j , where $i, j \in \{1, 2, 3, 4, 5, 6\}$. Segment 2 is acting as the biasing parameter to count the length of

the edges in graph G. The length of Segment 2 is determined by subtracting 20 units from the length of edges in graph G. For instance, edge (1,2) has 46 units of length as depicted in graph G. Therefore, the corresponding length of Segment 2 of DNA structure is 26 base pairs (bp). In short, the length of edges stated in graph G is linearly proportional to the length of the assigned DNA structure. All the encoded dsDNA sequences with corresponding to corresponding edges are listed in Table 7 – 2.

Figure 7 – 5: *Schematic representation of edge encoding scheme*

Edges	Double-stranded DNA sequence
(1, 2)	5' – CCCAGTTAATGCAACCTTGTAGCGGT – 3' 3' – TTGTCTTCCTGGGTCAATTACGTTGGAACATCGCCAGGGAGTAAAC – 5'
(2, 1)	5' – CCCAGTTAATGCAACCTTGTAGCGGT – 3' 3' – TGCTCCTTACGGGTCAATTACGTTGGAACATCGCCAGGCACAGATG – 5'
(1, 5)	5' – TATAAGTGCCCG – 3' 3' – TTGTCTTCCTATATTCACGGGCGCTTAAATCG – 5'
(5, 1)	5' – TATAAGTGCCCG – 3' 3' – TGGCGTACACATATTCACGGGCGGCACAGATG – 5'
(2, 3)	5' – TATGCTGACCATCCTACCTTCC – 3' 3' – TGCTCCTTACATACGACTGGTAGGATGGAAGGACCAATCTGC – 5'
(3, 2)	5' – TATGCTGACCATCCTACCTTCC – 3' 3' – GTCTCTCAAGATACGACTGGTAGGATGGAAGGGGGAGTAAAC – 5'
(2, 4)	5' – AAAGCTCTAGGG – 3' 3' – TGCTCCTTACTTTCGAGATCCCTCCGTGTGCT – 5'
(4, 2)	5' – AAAGCTCTAGGG – 3' 3' – AATACCTGCTTTTCGAGATCCCGGGAGTAAAC – 5'
(2, 6)	5' – ACATGACATCTCCGCTGA – 3' 3' – TGCTCCTTACTGTACTGTAGAGGCGACTCCCCAGAAGT – 5'
(6, 2)	5' – ACATGACATCTCCGCTGA – 3'

	3' -TGATAACAGG TGTACTGTAGAGGCGACT GGGAGTAAAC-5'
(3, 4)	5' - ATCTGGTATTCGTCCC -3' 3' -GTCTCTCAAG TAGACCATAAGCAGGG TCCGTGTGCT-5'
(4, 3)	5' - ATCTGGTATTCGTCCC -3' 3' -AATACCTGTC TAGACCATAAGCAGGG ACCAATCTGC-5'
(3, 5)	5' - CCCCTAGTCATCGTTACT -3' 3' -GTCTCTCAAG GGGGATCAGTAGCAATG AGCTTAAATCG-5'
(5, 3)	5' - CCCCTAGTCATCGTTACT -3' 3' -TGGCGTACAC GGGGATCAGTAGCAATGA ACCAATCTGC-5'
(4, 5)	5' - GCAAGTTTGG -3' 3' -AATACCTGTCC CGTTCAAACC GCTTAAATCG-5'
(5, 4)	5' - GCAAGTTTGG -3' 3' -TGGCGTACAC CGTTCAAACC TCCGTGTGCT-5'

Table 7 – 2: *Edges and corresponding double-stranded DNA (dsDNA) sequences*

DNA solution bay – In order to prevent any chance of cross-contamination during the experiment, all DNA solutions, that is required for subsequent processes are prepared within an isolated and decontaminated area, namely, DNA solution bay. The primers are short single-stranded DNA, which is also referring to as oligonucleotide, employed as “starting signal” to begin synthesizing the complimentary copy by following the DNA template during the polymerase chain reaction (PCR) process. All these primers are required to be artificially synthesized as listed in Table 7 – 3. The synthesized single-stranded DNA (ssDNA) of approximately 50 μM , duplex DNA of approximately 100 μM , and primer of approximately 100 μM .

Vertex	Forward primer sequence (5' to 3')	Reverse primer sequence (5' to 3')
1	CCGTGTCTACAACAGAAGGA	TCCTTCTGTTGTAGACACGG
2	CCCTCATTTGACGAGGAATG	CATTCTCGTCAAATGAGGG
3	TGGTTAGACGCAGAGAGTTC	GAACTCTCTGCGTCTAACCA
4	AGGCACACGATTATGGACAG	CTGTCCATAATCGTGTGCCT
5	CGAATTTAGCACCGCATGTG	CACATGCGGTGCTAAATTCTG
6	GGGGTCTTCAACTATTGTCC	GGACAATAGTTGAAGACCCC

Table 7 – 3: *DNA primers sequence design*

Mixing controller – After all DNA strands, ssDNA and dsDNA, as stated in Table 7 – 1, Table 7 – 2, and Table 7 – 3, are artificially synthesized in individual test tubes on DNA solution bay. A small amount of solution, representing vertices and edges in the graph G, is extracted from individual test tubes.

For single-stranded DNA (ssDNA), which represents the vertices in graph G, 1 μ l of such solution is taken from each of these test tubes. The amount of double-stranded DNA (dsDNA), which represents the edges in graph G, which is added into final mixture, is inversely proportional to their corresponding weights. Adopting the strategy of adjusting the concentration of dsDNA promotes the formation of the optimal path(s). The individual volume of solutions of dsDNA is listed in Table 7 – 4.

Graph S	Test Tube	Solution Volume (μ l)
Vertex 1 to 6	1 to 6 (ssDNA, 20 mer)	1.00
Edge (1, 2) & (2, 1)	7 and 8 (dsDNA, 46 bp)	0.78
Edge (1, 5) & (5, 1)	9 and 10 (dsDNA, 32 bp)	1.13
Edge (2, 3) & (3, 2)	11 and 12 (dsDNA, 42 bp)	0.86
Edge (2, 4) & (4, 2)	13 and 14 (dsDNA, 32 bp)	1.13
Edge (2, 6) & (6, 2)	15 and 16 (dsDNA, 38 bp)	0.95
Edge (3, 4) & (4, 3)	17 and 18 (dsDNA, 36 bp)	1.00
Edge (3, 5) & (5, 3)	19 and 20 (dsDNA, 38 bp)	0.95
Edge (4, 5) & (5, 4)	21 and 22 (dsDNA, 30 bp)	1.2

Table 7 – 4: *Selected volume of individual DNA solution*

The resultant 22 μ l mixture is subjected to a process known as hybridization, whereas the complementary ssDNAs and dsDNA duplexes with two sticky ends will readily annealed to each other by following Watson-Crick complementary principle. To optimize the result of hybridization, 2.4 μ l of 10 $\times$ ligation buffer is added into final mixture. The resultant solution is kept at room temperature for another 15 minutes to allow for the completion of hybridization process.

After the completion of hybridization process, complementary DNA strands are held together through relative weak hydrogen bonds. In order to fix the covalent bond between two adjacent DNA strands by joining corresponding hydroxyl (OH) group (also known as 3' end) and phosphate (P) group (also known as 5' end), the solution is subjected to a process called DNA ligation.

Prior to the ligation process, 2 μ l of NEB kinase³⁶ is added into the solution and sufficient time is given for the completion of respectful reaction. The propose of adding NEB kinase is to supply phosphate group to the 5' end of DNA strands. The process of DNA ligation is done by

³⁶ **NEB kinase:** also known as T4 polynucleotide kinase, which is commercialized by New England Biolabs® Inc. It is used to catalyze the transfer and exchange of phosphate group from ATP to 5' hydroxyl terminus of ssDNA and dsDNA.

adding 2 μl of NEB T4 DNA ligase³⁷. The DNA ligase, a specific class of enzyme, is dedicated in the formation of strong covalent bonds among annealed DNA strands. The resultant solution is incubated at room temperature (25 °C) for another 20 minutes.

For every 50 *pmol* (for these ssDNAs that are used to encode the vertices in graph G), there are approximately 1.5×10^{12} copies of associated DNA. Therefore, it is believed that all possible solution paths can be generated at the end of ligation process.

Solution purifier – In order to obtain the optimal outcome from subsequent PCR process, DNA solution from previous stage is immediately subjected to the solution purifier. In DNA mediated computing, the solution purifier is acting like a “noise signal remover”, which isolates and removes the impurities including incompletely hybridized oligonucleotides (short DNA strands), various redundant enzymes added in previous processes, and other possible impurities from solution.

Contemporarily, there are several optimized purification kits available in the market, which are manufactured to satisfy various requirements in accord with range of DNA fragment binding-size. In this experiment, QIAquick® PCR purification kit is used for the purification process.

The resultant solution from previous stage is 28.4 μl . After taking 2 μl of the mixture for concentration analysis, the remaining 26.4 μl solution is pipetted into a new test tube. A total volume of 132 μl of PB buffer from QIAquick® PCR Purification Kit, or 5 times of the amount of remaining solution, is added into this new test tube. The perfectly mixed solution is then poured into a QIAquick column and attached to a 2ml collection tube. After that, the entire sample in the test tube is centrifuged at 13,000 rpm (rounds per minute) for another 60 seconds. The QIAquick column is then placed back into a new test tube.

The 0.75 ml PE buffer from QIAquick PCR purification kit is added into the QIAquick column and the well mixed solution is centrifuged for another 60 seconds. Again, the flow through is discarded and the QIAquick column is placed back in the same tube. After that, the QIAquick column is centrifuged in a 2ml collection tube for another 1 minute to remove the residual buffer. The QIAquick column is then placed in a new 1.5 ml micro-centrifuge tube.

³⁷ **NEB T4 DNA ligase:** a specific DNA ligase commercialized by New England Biolabs® Inc. It is used to catalyze the formation of phosphodiester bond between 5' end and 3' end of adjacent DNA strands.

Finally, 50 μl TEB buffer is added into the membrane at the center of the column, and retain it at room temperature (25°C) for another 1 minute followed by one additional centrifugal process to elute the solution of DNA. The final elute contains roughly 40 to 50 μl of purified dsDNA duplex.

PCR amplifier – Polymerase chain reaction (PCR) is perhaps the most frequently employed laboratory technique in DNA mediated computing to amplify, or simply making duplicate copies of target “signal” – DNA template, by varying the “input signal” – various artificially synthesized DNA primers, as listed in Table 7 – 3, which is specified by the user of optimal route selection apparatus.

As mentioned in section 7.2, it is desired to determine the optimal path(s) traveling from two distinct locations home (vertex 1) and company (vertex 6) to the identical destination, which is hospital (vertex 4).

In the first case, the resultant optimal path(s) must starts from vertex 1 and terminates at vertex 4, the selected primers are likely to be forward primer of vertex 1 and reverse primer of vertex 4, as specified in Table 7 – 3. In the second case, the optimal path(s) must start from vertex 6 and terminates at vertex 4. Therefore, the selected primers are likely to be forward primer of vertex 6 and reverse primer of vertex 4.

The related primer solution together with the reagents of QIAGEN fast cycling PCR kit and solutions from previous stage are mixed in a new test tube, and the resultant solution is separated into two test tubes. Based upon the different primers selected for these two test tubes, they are labeled with “vertex 1 $\rightarrow$ vertex 4” and “vertex 6 $\rightarrow$ vertex 4”, respectively. Each test tube contains 14 μl reagents (10 μl of PCR master mix and 4 μl of Q solution), 2 μl of corresponding forward primer, 2 μl of corresponding reverse primer, and 2 μl of purified DNA solution from previous stage, resulting in a mixture of 20 μl prior to process of PCR.

The employed PCR protocol is described as follows:

Step 1: Increase solution temperature to 95 °C and maintain at that temperature for 5 minutes. DNA polymerase is inactivated in the room temperature, increasing and maintaining the solution temperature at 95 °C is aiming to activate the corresponding polymerase.

- Step 2: Increase the solution temperature to 96 °C and maintain for 5 seconds. This double ensures all dsDNA duplex retained in solution is denatured in the formation of ssDNA.
- Step 3: Gradually reduce solution temperature to 48 °C and maintain at that temperature for another 5 seconds. This step allows annealing of primers to target DNA template(s).
- Step 4: Increase solution temperature to 68 °C and maintain for 5 seconds. This step promotes the extension of annealed primers.
- Step 5: Repeat step 2 to step 4 for another 29 rounds.
- Step 6: Increase solution temperature to 72 °C and maintain for another 1 minute.
- Step 7: Gradually reduce the temperature to room temperature (25 °C).

As a consequence of polymerase chain reaction (PCR), in test tube labeled “vertex 1 → vertex 4”, only the dsDNA duplex encoding paths, which begins with vertex 1 and terminates at vertex 4, are amplified; the identical condition applies to test tube labeled “vertex 6 → vertex 4”, where the dsDNA duplex encoding paths, which begins with vertex 6 and terminates at vertex 4, are amplified. After PCR process, it is recommended to subject the solution to solution purifier again, such that, any impurities, which may affect the result of experiment, will be rejected at the end of purification process. However, such process is not necessary.

Gel electrophoresis – Gel electrophoresis employs native polyacrylamide gel as sieving material to distribute linear DNA molecules with respect to its corresponding lengths. In the experiment, 12% native polyacrylamide gel is prepared according to the recipe as specified in Table 7 – 5.

Ingredient	Volume (μ l)
H ₂ O	15.700
30% acrylamide	16.000
5 ×TBE buffer	8.000
10% ammonium persulfate	0.280
TEMED	0.026

Table 7 – 5: *12% native polyacrylamide gel recipe*

As specified in Figure 7 – 6, there are totally four solution slots, or wells, are built in the gel slab prior to electrophoresis, which used to contain different DNA solutions – *well 1*: DNA ladders in 10 bp resolution; *well 2*: dsDNA template of known length of 82 bp; *well 3*: PCR product of solution in test tube “vertex 1 → vertex 4”; *well 4*: PCR product of solution in test tube “vertex 6 → vertex 4”.

According to its working principle, the migration distance of DNA solution towards anode electrode is inversely proportional to the length of linear dsDNA strands. Therefore, the most prominent DNA band along lane 3 and lane 4, as a result of polymerase chain reaction (PCR), with the maximum migration length towards anode electrode is likely to be DNA strands, which represents the solution to optimal path problem. *Well 1* is used to comprise DNA ladders, ranging from 50 bp to 150 bp, in 10 bp interval. *Well 2* is served for the same purpose, which contains DNA molecules of known length of 82 bp.

The gel slab immersed in TBE buffer is subjected to electrophoresis process by setting the power at 18 W, and with variable voltage and current at approximately 560 V and 32 mA, respectively. The entire electrophoresis process takes approximate 1 hour 45 minutes to complete. After gel electrophoresis, it is possible to determine the total length of optimal path by comparing the migration lengths of the most prominent band in *lane 3* and *lane 4* to the DNA bands in reference lanes, that is, *lane 1* and *lane 2*.

7.5 Results and Discussion

After the completion of electrophoresis, the gel is immersed in the TBE buffer containing $3 \times$ concentration of GelRed³⁸. The staining process takes approximately 30 minutes to complete. Finally, the results are observed in a 2-dimensional multi-florescence scanner Typhoon 9410. Under the green-excited mode, the DNA bands in gel slab have a strong emission once subjected to excitation at 532 nm wavelength ultraviolet (UV) light.

The 2D image of 12% native acrylamide gel of four lanes is displayed in Figure 7 – 6 (a). As mentioned in section 7.4, *lane 1* is the reference path which contains the DNA ladders ranging from 50 bp to 150 bp with interval of 10 bp. *Lane 2* contains the double-stranded DNA (dsDNA) duplex of specified length, that is, 82 bp, which has the exact length as theoretically predicted length of shortest path from home (vertex 1) to hospital (vertex 4), that is, the path “vertex 1 → vertex 5 → vertex 4”. The length is mathematically calculated in section 7.2 as well as referring to the encoding scheme as specified in section 7.4. The reference dsDNA duplex in *lane 2* is used to determine the migration length of the most prominent band in *lane 3*. In simple words, it is going to testify the following hypothesis: the resultant DNA in *lane 3* as a result of polymerase chain reaction (PCR) process, which represents the shortest path from home (vertex 1) to hospital (vertex 4), has identical migration distance, and therefore identical length, with respect to *lane 2*. By observing Figure 7 – 6 (a), a uniquely visible band in *lane 3* has equivalent migration distance as compared to the content in *lane 2*. Therefore, it is believed the DNA-based optimal route selection apparatus has selected the predicted path, as shown in Figure 7 – 6 (b).

Lane 4 is used to contain the product, as a result of PCR by selecting the primers representing the starting and terminating points of company (vertex 6) and hospital (vertex 4), respectively. Analogously, a unique solution band, as shown in *lane 4* of Figure 7 – 6 (a), can be observed. As compared to various DNA bands in *lane 1* (DNA ladder), it is possible to determine the migration distance of the unique band in *lane 4* is equivalent to that of dsDNA of 90bp in *lane 1*. In the case study, as specified in Figure 7 – 1, all generated paths are unique in their corresponding lengths. In other words, it is possible to figure out the path simply based upon the length of dsDNA duplex. Therefore, the optimal path from company (vertex 6) to hospital (vertex 4) is likely to be the path

³⁸ **GelRed:** a typical fluorescent nucleic acid gel stain which is designed to replace previous toxic ethidium bromide.

“vertex 6 → vertex 2 → vertex 4”, which is exactly the same as the theoretically predicted result specified in section 7.2.

Figure 7 – 6: 2D 12% native polyacrylamide gel image and corresponding optimal paths

7.6 Conclusion

The DNA-based optimal route selection apparatus, as theoretically proposed in section 7.3, is experimentally demonstrated to estimate the optimal path(s) between any two physical locations, which are manually specified by user of the biomolecular device, of a road map including six locations. The entire experimental procedures are described in section 7.4. And the final results are presented in section 7.5.

The specifically designed algorithm is believed to maximize the property of biomolecular computing, that is, massive parallelism. The procedures indirectly reveal the fact that biomolecular computing is extremely efficient in handling the problems where massive searching ability is required by the algorithm. The only drawback of current version biomolecular computing algorithm is that with the size of problem increases, *i.e.*, the number of physical locations and bidirectional edges in the road map increases, the single-stranded DNA (ssDNA), which is demanded to be artificially synthesized, is increased accordingly. Despite that, the developed DNA-base apparatus has several advantages over conventional silicon-based methods in solving similar problems, such as conducting massive data storage and simultaneous processing for path selections via much fewer materials than conventional silicon devices. The utilizing length of DNA strands, as the biasing parameter, is believed to be superior to other alternatives, such as varying the pH value or concentration of DNA solution.

Furthermore, with proper coding the map into DNA edge duplex, the optimal paths can be deduced directly by the length of PCR outcomes. Hence the current apparatus avoids the tremendous processing cost and time in DNA sequencing, which is required by many precedent DNA-based systems to address mathematical problems.

Chapter 8

Biomolecular Mediated Normative Decision Making Apparatus

In nature, most of our daily activities, such as interacting with information received from the outside world and responding with appropriate reactions, are processed by our brains, which is a magnificent decision making apparatus. The actual mechanism on how our genetic material, deoxyribonucleic acid, is translated into functional amino acids and eventually dedicated in the construction of sophisticated decision making system still remains mysterious. The intrinsic characteristics of DNA molecules, on the other hand, reveal the feasibility that artificially-synthesized DNA can be eventually utilized as information storage and processing substrate to make rational decisions based upon expected utility hypothesis.

In Chapter 6, a novel-based biomolecular mediated device has been proposed to handle a specific combinatorial optimization problem, namely, strategic assignment problem [111]. The feasibility of constructing an functional in vitro biomolecular mediated device, which is able to determine the most optimal route(s) by manually assigning the starting location and terminating location, has been experimentally proved in Chapter 7.

In this chapter, the potential of biomolecular mediated computing is further extended into other seemingly unrelated discipline, namely, normative decision maker based upon expected utility hypothesis. The developed biomolecular apparatus is capable of handling a typical situation, which is known as a regional military conflict, in a famous board game – the risk®, based on the expected utility hypothesis.

This chapter is organized as follows: it begins with a brief introduction on how a digital decision making apparatus is able to make the most rational choice among three predefined options, as included in section 8.1. The method on how the entire decision-making situation can be simply represented by a decision tree and how the toe-hold mediated strand displacement mechanism is capable of representing various probabilities are depicted in section 8.2. The detailed information

encoding scheme are described in section 8.3, whereas the DNA sequence design motif are visually expressed in Figure 8 – 6. The laboratory procedures are described in section 8.4.

8.1 Normative Decision Making Apparatus

Normative decision-making apparatus is a typical device, which is employed to simulate the rational behaviors of individuals when facing situations with the involvement of risk. Expected utility hypothesis, abbreviated EU, which is vastly used in the analysis of decision making under risk, is adopted as a normative model of rational choice [115 – 118].

The classical expected utility model of decision making under risk can be abstractly described as follows: Given a situation with a set of predefined alternative options O_i and by assuming individual decision maker follows a series of predefined axioms, the rational choice is subjected to the option, whereas the expected utility U , as described by equation 8.1, is maximized.

$$U = \max \sum P_j u(x_j) \quad (8.1)$$

Where P_j denotes the probability at each possible outcome x_j , and $u(x_j)$ denotes the utility of receiving outcome x_j .

Biomolecular normative decision making apparatus is a device, which is able to accomplish the identical function as that can be delivered by a conventional silicon-based computer, by in-vitro³⁹ manipulating artificially-encoded DNA molecules. It is demonstrated to solve a strategic situation in a simplified version of a famous strategic board game, namely, the risk®, which was produced by Parker Brothers, an American toy and game manufacturer. The objective of the game is to occupy every single territory on the game board by eliminating all the military infantries owned by other player(s).

The selected case is a regional military conflict in the game between player 1, indicated by red node, and player 2, indicated by blue node, as shown in Figure 8 – 1. The numbers as specified in these nodes are units of military infantry defending that corresponding territories. As specified in Figure 8 – 1, player 1 has already accumulated 11 units of infantry in Greenland prior to his attacking phase. Based upon game rule of risk [119 – 121], in the attacking phase of player 1, 10 available infantry units can be deployed in invading the adjacent territories, namely Northwest Territory (2 units of defending infantry of player 2), Ontario (7 units of defending infantry), and Quebec (5 units of defending infantry).

³⁹ **In-Vitro:** in molecular biology, there are two types of manipulation methods called *in-vitro* and *in-vivo*. The in-vitro method is mainly used to study the functions of individual components of an organism by isolating them from their biological surrounding.

Figure 8 – 1: Regional map of a strategic board game – risk

Suppose the player 1 drafted three invading plans, which are also known as *options*, as specified below:

- *Option 1*: sending 10 infantry units to Ontario, where is defended by 7 infantry units of player 2, as shown in Figure 8 – 2 (a).
- *Option 2*: sending 7 infantry units to Quebec, where is defended by 5 infantry units of player 2; and sending 3 infantry units to Northwest Territory, where is defended by 2 infantry units of player 2, as shown in Figure 8 – 2 (b).
- *Option 3*: sending 2 infantry units to Northwest Territory, where is defended by 2 infantry units of player 2; and sending 8 infantry units to Ontario, where is defended by 7 infantry units of player 2, as shown in Figure 8 – 2 (c).

In order to determine the most rational choice among three available options based upon the expected utility hypothesis, two information is compulsory – the probabilities of possible outcomes and the corresponding utility of receiving that corresponding outcome. The probabilities of attacker, player 1, winning a battle in risk, with different number of attacking and defending infantry units had been determined by [122], which can be listed in Table 8 – 1.

(a) Option 1

Expected Utility of Option 1:

$$\begin{aligned}
 U_1 &= P_{NT}u(X_N) + P_{ON}u(X_{ON}) + P_{QU}u(X_{QU}) \\
 &= 0\% \times 2 + 80\% \times 7 + 0\% \times 5 \\
 &= 5.6
 \end{aligned}$$

(b) Option 2

Expected Utility of Option 2:

$$\begin{aligned}
 U_2 &= P_{NT}u(X_N) + P_{ON}u(X_{ON}) + P_{QU}u(X_{QU}) \\
 &= 66\% \times 2 + 0\% \times 7 + 74\% \times 5 \\
 &= 5.02
 \end{aligned}$$

(c) Option 3

Expected Utility of Option 3:

$$\begin{aligned}
 U_3 &= P_{NT}u(X_N) + P_{ON}u(X_{ON}) + P_{QU}u(X_{QU}) \\
 &= 36\% \times 2 + 64\% \times 7 + 0\% \times 5 \\
 &= 5.2
 \end{aligned}$$

Figure 8 – 2: Expected utility of three options

		Number of attacking armies									
		1	2	3	4	5	6	7	8	9	10
Number of defending armies	1	42%	75%	92%	97%	99%	>99%	>99%	>99%	>99%	>99%
	2	11%	36%	66%	79%	89%	93%	97%	98%	99%	99%
	3	3%	21%	47%	64%	77%	86%	91%	95%	97%	98%
	4	1%	9%	31%	48%	64%	74%	83%	89%	93%	95%
	5	<1%	5%	21%	36%	51%	64%	74%	82%	87%	92%
	6	<1%	2%	13%	25%	40%	52%	64%	73%	81%	86%
	7	<1%	1%	8%	18%	30%	42%	54%	64%	73%	80%
	8	<1%	<1%	5%	12%	22%	33%	45%	55%	65%	72%
	9	<1%	<1%	3%	9%	16%	26%	36%	46%	56%	65%
	10	<1%	<1%	2%	6%	12%	19%	29%	38%	48%	57%

Table 8 – 1: *The various probabilities of attacker winning a battle in risk® [9]*

In addition to the probabilities as mentioned above, another compulsory information is the utility of receiving various outcomes. As mentioned in the beginning, the objective of risk® game is to eliminate all the military infantries owned by other player(s) on the game board. On the perspective of the attacker, in this case, player 1, the utility of receiving potential outcomes for player 1 can be measured in terms of corresponding units of defending infantries have eliminated in that battle by invading that specific territory. Therefore, by neglecting all the other possible factors, such as the geometrical advantage, the utility of receiving various outcomes by occupying Northwest Territory, Ontario, and Quebec are 2 units, 7 units, and 5 units, respectively. In addition to that, in order to simply the procedure, the tactic withdraw is not allowed once the option is implemented, *i.e.*, once the battle (the action of invading) begins, it terminates only if either players' infantries in the corresponding territories are completely eliminated. Therefore, based upon the information provided above together with equation 8.1, it is possible to determine the value of expected utility for each option as indicated from Figure 8 – 2 (a) to (c).

By simply referring to the result as indicated in Figure 8 – 2 (a) to (c), it is possible to conclude that the most rational choice based upon expected utility hypothesis is likely to be option 1. In the remaining of this chapter, the method of biomolecular computing is proposed to solve the same decision involved situation, based on the expected utility hypothesis, which is able to return with the identical result as that predicted by a digital computer.

8.2 Methods

At the beginning, the entire decision-making situation can be simply expressed in terms of a decision tree, as shown in Figure 8 – 3.

Figure 8 – 3: Decision tree of regional military conflict

The top hierarchy of tree diagram is a *decision node*, which stands for the stage the decision maker selects an option from a set of predefined alternative options; in this case, there are totally three available options. Each option leads to one round node, which is known as the *chance node*. The chance node is associated with various probabilities, as indicated in Table 8 – 1, leading to various outcomes utility nodes. In addition to the original decision tree, one additional node, which is named as *terminating node*, is included in the graph. By doing so, the original decision tree is converted into a directed network as shown in Figure 8 – 3. In this directed graph, any

paths, which are encoded by various DNA strands, have to start from choice node and ends at the terminating node.

In DNA-based normative decision making apparatus, majority of the laboratory procedures are similar to that were employed in Chapter 6 (DNA-based computing of strategic assignment problem) and Chapter 7 (DNA as an optimal route selection apparatus). The only difference is that the DNA-based normative decision making apparatus involves the representation of various probabilities and utilities of receiving various possible outcomes.

In this case, one additional in-vitro biomolecular method is introduced, which is known as toe-hold mediated strand displacement. The procedures of this methodology can be visually described by Figure 8 – 4.

Figure 8 – 4: *A schematic representation of toe-hold strand displacement method*

As indicated in Figure 8 – 4, the single-stranded DNA, also known as invading strand, consists of two regions – region α and β . The one-sided sticky end of double-stranded DNA (dsDNA) is also known as the toe-hold region, which is denoted by β' . The regions of α and β are complementary to the regions of α' and β' , respectively. After the annealing of invading strand to the toe-hold region, it starts to invade through region of α' . After the strand displacement, the original γ strand is completely peeled off from dsDNA with one sticky end. The entire process is called toe-hold mediated strand displacement.

The probability and utility of receiving one specific outcome can be represented by employing the toe-hold mediated strand displacement mechanism. The concept can be visually demonstrated by Figure 8 – 5. In Figure 8 – 5 (a), there are 10 units of single-stranded DNA (ssDNA), containing region of α , which is complementary to region of α' of ssDNA. In addition to that, there are 10 units of single-stranded DNA (ssDNA), containing region of β , which is complementary to region of β' of ssDNA. Assuming all these 10 units of ssDNA are involved in the same solution, it is able to generate 10 units of dsDNA of two sticky ends upon the completion of hybridization process.

Figure 8 – 5: *Probability controlled by utilizing toe-hold mediated strand displacement mechanism*

In Figure 8 – 5 (b), one additional DNA structure is introduced into the solution as shown in Figure 8 – 5 (a). The introduced DNA structure is a double-stranded DNA (dsDNA) duplex with one sticky end. The sticky end is served as the toe-hold region, as described in Figure 8 – 4, which is able to anneal with the single-stranded DNA (ssDNA) of region α' and β' by subsequently displacing the originally attached oligonucleotide, as shown in Figure 8 – 5 (b). The displaced short oligonucleotide, which is encircled in Figure 8 – 5 (b), is termed as waste. Assuming the introduced dsDNA with one sticky is in x units, where x is less than 10 units. By introducing such new dsDNA structure into the original solution, it is going to compete with the original two ssDNA in the formation of dsDNA duplex of two sticky ends. Such that, supposing the introduced dsDNA structure is of x units. The dsDNA duplex of two sticky ends is likely to be $10 - x$ units. In this sense, the probability of forming dsDNA of two sticky ends can be simply controlled by adjusting the value of x . The method is going to be employed in subsequent section, section 8.3.

8.3 Information Encoding Scheme

As being analogous to the very beginning stage of contemporary silicon-based computer, whereas the information is firstly digitized in terms of binary expressions, the modified decision tree, as specified in Figure 8 – 3, is first converted into a combination of DNA sequences, namely, A (Adenine), T (Thymine), G (Guanine), and C (Cytosine), based upon a specifically designed DNA sequence design motif, which is also called information encoding scheme, as described in Figure 8 – 6.

The information encoding scheme starts from two types of nodes: decision node, denoted by D, and termination node, denoted by T. These two nodes are 20 bp partial double-stranded DNA (dsDNA) of one-sided sticky end of length of 20 mer. The sticky end of DNA sequences, which encodes the decision node, is complementary to former 20 mer of single-stranded DNA (ssDNA), which encodes edge E_{DO_i} , where $i \in \{1,2,3\}$. Similarly, the sticky end of DNA sequences, which encodes termination node, is complementary to rear 20 mer of single-stranded DNA (ssDNA), which encodes edge $E_{u(x_k)T}$, where $k \in \{NT, ON, QU\}$. These two partial double-stranded DNA (dsDNA) is also served for another purpose, that is, the dsDNA portion of these two DNA molecules is acting as the template during the process of polymerase chain reaction (PCR), where the selected primers are annealed to these regions after the template denaturation in each cycle of PCR.

The DNA structures, which are used to encode three options, are 6 bp partial double-stranded DNA (dsDNA) of two-sided sticky ends of length of 20 mer for each end. The sticky ends are complementary to the corresponding DNA sequences of edges. The former 20 mer sticky end is complementary to rear 20 mer DNA sequences encoding edge E_{DO_i} . Analogously, the rear 20 mer sticky end is complementary to former 20 mer DNA sequences encoding edge $E_{O_iP_j}$, where $i \in \{1,2,3\}, j \in \{NT, ON, QU\}$. The only sequential difference among these DNA structures, which are used to encode three different options, is on the portion of 6 bp double-stranded DNA (dsDNA), the middle sections. In addition to that, these middle sections, 6 bp dsDNA, contains a recognition site with respect to one specific restriction endonuclease – *option 1: Proteus vulgaris* (PvuII); *option 2: Haemophilus parainfluenzae* (HpaI); *option 3: Streptomyces tubercidicus* (StuI). The selected restriction endonucleases have two properties in common, that is, all these chosen enzymes are only effective on double-stranded DNA (dsDNA) and able to cleave the target 20 bp dsDNA into two equal length dsDNAs of 10 bp with blunt ends.

Figure 8 – 6: DNA sequence design motif

Various probabilities, as specified in Figure 8 – 3, are encoded with normal 40 mer single-stranded DNA (ssDNA) sequences, denoted by P_j , where $j \in \{NT, ON, QU\}$. The probabilities are controlled by introducing another class of DNA sequences, namely, threshold DNA sequences, denoted by Th_{P_j} . Without the presence of threshold DNA (Th_{P_j}), the DNA sequence, which represents probability (P_j), is capable of joining the corresponding DNA sequences of edge $E_{O_iP_j}$ and edge $E_{P_ju(x_k)}$, as indicated in Figure 8 – 6 (a), due to its designated nature – the former 20 mer single-stranded DNA (ssDNA) is complementary to the rear 20 mer ssDNA of $E_{O_iP_j}$, and rear 20 mer ssDNA is complementary to former 20 mer ssDNA of $E_{P_ju(x_k)}$. The obtained result promotes the formation of dsDNA duplex encoding solution path as described in Figure 8 – 6 (a). By introducing manually controlled concentration of threshold DNA sequences (Th_{P_j}) into the solution, DNA sequence representing probability (P_j) is likely to react with the corresponding threshold DNA sequence (Th_{P_j}) due to toe-hold mediated strand displacement mechanism. The resultant DNA sequences lead to two distinct parts, as shown in Figure 8 – 6 (b). These parts are treated as waste, which is going to be removed from the solution in the subsequent procedures – solution purification and gel electrophoresis. It is because none of these structures contributes to the formation of DNA sequence, known as solution path, as specified in Figure 8 – 6 (a). In other words, the formation of wastes inhibits the formation of solution path as the ssDNAs, which encode various probabilities P_j are “grabbed” by corresponding threshold DNA sequences Th_{P_j} .

By following the toe-hold mediated strand displacement mechanism, the probabilities of obtaining specific utilities of various outcomes, as mentioned in Figure 8 – 3, can be therefore controlled by using various concentration of threshold DNA sequences (Th_{P_j}). The corresponding concentration ratio, denoted by R_{P_j} , between threshold DNA sequence (Th_{P_j}) and probability DNA sequence (P_j) can be simply estimated by using equation 8.2.

$$R_{P_j} = 1 - P_j \quad (8.2)$$

Therefore, in the case study, the value of R_{ON} for option 1 is 0.2; the value of R_{NT} and R_{QU} for option 2 is 0.34 and 0.26, respectively; the value of R_{NT} and R_{ON} for option 3 is 0.64 and 0.36, respectively.

The DNA structures, which are used to encode the utilities, are similar to that is used to encode various options. They are 6 bp partial double-stranded DNA (dsDNA) of two-sided sticky ends of length of 20 mer in each end. The sticky ends are complementary to the corresponding DNA sequences of edges. The former 20 mer sticky end is complementary to rear 20 mer DNA sequences encoding edge $E_{P_j u(x_k)}$. Similarly, the rear 20 mer sticky end is complementary to former 20 mer DNA sequences encoding edge $E_{u(x_k)T}$, where $k \in \{NT, ON, QU\}$. Similar to the encoding method, which is applied to encode the options, the only sequential difference among these DNA structures, which are used to encode three possible utilities from different outcomes, is on the 6 bp double-stranded DNA (dsDNA), the middle portion. Analogously, the middle sections of these partial dsDNA are specifically designed, which contains a recognition site corresponding to one specific designed restriction endonuclease – $u(x_{NT})$: *Pseudomonas maltophilia* (PmlI); $u(x_{ON})$: *Escherichia coli* (EcoRV); $u(x_{QU})$: *Streptomyces caespitosus* (ScaI). In addition to that, the concentration of the DNA sequences, which are used to encode the utilities of potential outcomes, is determined with respect to the value as specified in Figure 8 – 3. The corresponding ratio of utilities of outcomes, denoted by $R_{u(x_k)}$, for attacking territory Northwest Territory (NT), Ontario (ON), and Quebec (QU) are 0.2 ($R_{u(x_{NT})}$), 0.7 ($R_{u(x_{ON})}$) and 0.5 ($R_{u(x_{QU})}$), respectively.

For the remaining edges, as indicated in Figure 8 – 6, only provides one single function, that is, acting as a “connecting bridge” which links two respectful DNA strands. There are totally four edges constitute the formation of double-stranded DNA encoding the final solution path, as specified in Figure 8 – 6 (a). These edges are E_{DO_i} , $E_{O_i P_j}$, $E_{P_j u(x_k)}$, and $E_{u(x_k)T}$. All the edges are encoded by 40 mer single-stranded DNA (ssDNA). In order to avoid the undesired hybridization in the subsequent procedures, all DNA sequences used to represent edges are accomplished based on the rules as specified in [114].

8.4 Implementation

After all DNA strands, single-stranded DNA and partial double-stranded DNA, as specified in Figure 8 – 6, are artificially synthesized in individual test tubes, a small amount of solution, which represents vertices and edges in graph G , is pipetted from individual test tubes into one empty test tube. The synthesized DNA solutions are of the identical concentration of approximately $100 \mu M$.

For DNA sequences, which represent the decision node (D), termination node (T), and various edges ($E_{DO_i}, E_{O_i P_j}, E_{P_j u(x_k)}, E_{u(x_k) T}$), $2 \mu l$ of such solution of equal concentration is taken from each test tube. For option (O_i) and probability (P_j), $1 \mu l$ of such solution of equal concentration is taken from each test tube. The partial double-stranded DNA (dsDNA), which represents the utility ($u(x_k)$) and threshold (Th_{P_j}) in graph G , the amount of solution added into the final mixture is determined by simply referring to the concentration ratios, the values of R_{P_j} and $R_{u(x_k)}$, as mentioned above, which is concluded in Table 8 – 1.

Concentration ratio of R_{P_j} , where $j \in \{ON, NT, QU\}$	
Option	Concentration ratio of threshold DNA sequence
1	$R_{ON} = 0.2$
2	$R_{NT} = 0.34$ $R_{QU} = 0.26$
3	$R_{NT} = 0.64$ $R_{ON} = 0.36$
Concentration ratio of $R_{u(x_k)}$, where $k \in \{ON, NT, QU\}$	
Utility of outcomes	Concentration ratio of probability DNA sequence
Northwest Territory, NT	$R_{u(x_{NT})} = 0.2$
Ontario, ON	$R_{u(x_{ON})} = 0.7$
Quebec, QU	$R_{u(x_{QU})} = 0.5$

Table 8 – 2: *The concentration ratios of corresponding DNA sequences*

By adopting the strategy of adjusting the concentration of corresponding DNA solutions, it promotes the formation of DNA duplex, encoding solution path, as shown in Figure 8 – 6. Meanwhile, by controlling the concentration of various DNA solutions, which encode the threshold DNA sequences, the probability of option leads to various outcomes can be appropriately controlled. As the artificially synthesized DNA solutions are of identical concentration, it is possible to control the concentration of individual DNA solutions by simply controlling the solution volume added into the mixture. The individual volume of solutions of DNA added into the mixture is listed in Table 8 – 2.

Graph G	No. of Test Tube		Solution Volume
Decision node, D	1	(partial dsDNA, 20 bp)	2.00 μ l
Option 1 to 3	2 to 4	(partial dsDNA, 6 bp)	1.00 μ l
Probability P_{NT}, P_{ON}, P_{QU}	5 to 7	(ssDNA, 40 mer)	1.00 μ l
Threshold $Th_{P_{ON}}$ (option 1)	8	(partial dsDNA, 10 bp)	0.20 μ l
Threshold $Th_{P_{NT}}$ (option 2)	9	(partial dsDNA, 10 bp)	0.34 μ l
Threshold $Th_{P_{QU}}$ (option 2)	10	(partial dsDNA, 10 bp)	0.26 μ l
Threshold $Th_{P_{NT}}$ (option 3)	11	(partial dsDNA, 10 bp)	0.64 μ l
Threshold $Th_{P_{ON}}$ (option 3)	12	(partial dsDNA, 10 bp)	0.36 μ l
Utility $u(x_{NT})$	13	(partial dsDNA, 6 bp)	0.20 μ l
Utility $u(x_{ON})$	14	(partial dsDNA, 6 bp)	0.70 μ l
Utility $u(x_{QU})$	15	(partial dsDNA, 6 bp)	0.50 μ l
Termination node, T	16	(partial dsDNA, 20 bp)	2.00 μ l
Edge $E_{DO_i}, E_{O_iP_j}, E_{P_ju(x_k)}, E_{u(x_k)T}$	17 to 20	(dsDNA, 40 bp)	2.00 μ l

Table 8 – 3: *Selected volume of individual DNA solutions*

At the beginning, one empty test tube is prepared. All the solutions as listed above are pipetted into the test tube, resulting in a total mixture of 21.2 μ l. To obtain the optimized hybridization result, 2.12 μ l of 10 $\times$ ligation buffer is added into the mixture. The resulting solution is kept at the room temperature (25 $^{\circ}$ C) for 30 minutes to allow for the completion of entire hybridization process. During the process, complementary single-stranded DNAs (ssDNAs) and double-stranded DNAs (dsDNAs) duplexes with sticky end(s) will readily annealed to each other by following the Watson-Crick complementary principle. In addition to the hybridization process, the toe-hold mediated strand displacement occurs at the same time, whereas the formation of double-stranded DNA, the waste, inhibits the formation of dsDNA encoding solution path.

After these combined process, complementary DNA strands are held together through relatively weak hydrogen bonds. 2 μ l of NEB kinase is added into the mixture is to supply phosphate group to all 5' end of all DNA strands. After the addition of NEB kinase, sufficient time is allowed for the completion of reaction. Subsequently, 2 μ l of NEB T4 DNA ligase is added into the solution. The purpose of adding T4 DNA ligase is to “repair” the covalent bond between two adjacent DNA strands by joining corresponding hydroxyl (OH) group (also known as 3' end) and phosphate group (also known as 5' end). This process is called DNA ligation. After adding T4 DNA ligase into solution, the resulting solution is incubated at room temperature, 25 $^{\circ}$ C, for 20 minutes. At the end of ligation process, the strong covalent bonds are established among annealed DNA strands. For every 50 pmol DNA solution, there are approximately 1.5×10^{12} copies of associated DNA encoding various parts as specified in Figure 8 – 6.

The resulting 27.32 μ l solution is immediately subjected to the solution purifier. In the purification process, the QIAquick® PCR purification kit is exploited. The detailed procedures of

purification process are described in section 7.4, solution purifier stage. The objective of solution purification is to remove all the waste and other possible impurities during the procedures of hybridization and toe-hold mediated strand displacement.

After the purification process, the collected solution is subjected to the process of polymerase chain reaction (PCR). Two specifically designed DNA primers are added into the solution. They are 20 mer single-stranded DNA (ssDNA). One primer is complementary to forward 20 mer DNA encoding decision node, which is in the sequence of $3' - CCTGGCTGTGTGTTTCTGGA - 5'$. Another primer is complementary to rear 20 mer DNA encoding termination node, which is in the sequence of $3' - CGGTTACTGGAAAGTCCATG - 5'$. In addition to the primers, the QIAGEN fast cycling PCR kit is employed prior to the process of PCR. A total $12\ \mu l$ of reagents are added into the solution, which includes $10\ \mu l$ of PCR master mix and $2\ \mu l$ of Q solution. The following protocol is employed in the process of polymerase chain reaction:

1. Increase solution temperature to $95\ ^\circ\text{C}$ and maintain at the temperature for 5 minutes.
2. Increase solution temperature to $96\ ^\circ\text{C}$ and maintain at the temperature for 5 seconds to allow for the denaturation process.
3. Gradually reduce the temperature of solution to $43\ ^\circ\text{C}$ and retain at that temperature for 5 seconds to promote the annealing of DNA primers.
4. Increase the solution temperature to $68\ ^\circ\text{C}$ and maintain at that temperature for 5 seconds, which promotes the extension of primers.
5. Repeat protocol from step 2 to step 4 for another 29 cycles.
6. Increase the temperature of solution to $72\ ^\circ\text{C}$ and retain at the temperature for another 1 minute.
7. Gradually reduce the solution temperature to room temperature ($25\ ^\circ\text{C}$).

As a result of polymerase chain reaction (PCR) process, only those solution paths begin with DNA sequences encoding decision node D and ends with DNA sequences encoding termination node T are amplified. In addition to that, even though the difference between solution paths, which represent various expected utility is extremely small after the process of hybridization and toe-hold mediated strand displacement. Such difference has been magnified by 2^{30} times at the end of process of PCR.

The collected solution is equally divided into three empty test tubes, which can be labeled as “TT1” (test tube 1), “TT2” (test tube 2), and “TT3” (test tube 3). After that, the corresponding restriction endonucleases are added into these test tubes as described below:

- Test tube 1: *Haemophilus parainfluenzae* (for option 2), *Streptomyces tubercidicus* (for option 3), *Pseudomonas maltophilia* (for utility $u(x_{NT})$), and *Streptomyces caespitosus* (for utility $u(x_{QU})$).
- Test tube 2: *Proteus vulgaris* (for option 1), *Streptomyces tubercidicus* (for option 3), and *Escherichia coli* (for utility $u(x_{ON})$).
- Test tube 3: *Proteus vulgaris* (for option 1), *Haemophilus parainfluenzae* (for option 2), and *Streptomyces caespitosus* (for utility $u(x_{QU})$).

In *test tube 1*, “TT1”, it is desired to retain all the solution paths containing DNA sequences, which encode option 1. Therefore, two restriction endonucleases, *Haemophilus parainfluenzae* (for option 2), *Streptomyces tubercidicus* (for option 3), are added. With the addition of these enzymes, the double-stranded DNA (dsDNA) duplexes, which contain recognition sites for option 2 and option 3, are cleaved into two equal-length portions. Such that, in the subsequent procedure, which is gel electrophoresis, these strands are not able to satisfy the migration length requirement, and therefore, they are rejected from the final solution. The addition of restriction endonuclease *Pseudomonas maltophilia* (for $u(x_{NT})$) and *Streptomyces caespitosus* (for $u(x_{QU})$) is due to the probability leading to these two outcomes are zero. Therefore, these dsDNA encoding solution paths, which contain the recognition sites, are analogously removed at the end of gel electrophoresis. The same method is extended to *test tube 2* (“TT2”) as well as *test tube 3* (“TT3”).

All these test tubes, which contain a mixture of DNA solutions, appropriate restriction enzyme, and selected buffer, are immersed into a water tank maintaining the temperature of 37 °C for 10 minutes to promote effective incubation. At the end of digestion process, the DNA solution is subjected to the purification process again, which aims to remove any potential impurities, to improve the resolution of DNA bands retained in the gel slab after the process of electrophoresis.

Gel electrophoresis employs native polyacrylamide gel as sieving material to distribute linear DNA molecules with respect to its length. In this experiment, 12% native polyacrylamide gel is prepared according to the recipe as mentioned in Table 7 – 5, section 7.4.

There are totally five solution slots, also known as wells, are built in the gel slab prior to the process of gel electrophoresis. These solution slots are used to contain different DNA solutions – *well 1*: DNA ladders ranging from 10 bp to 250 bp in the interval of 20 bp; *well 2*: DNA solution in the test tube 1; *well 3*: DNA solution in the test tube 2; *well 4*: DNA solution in the test tube 3; *well 5*: linear DNA solution of known length of 212 bp. The 212 bp is the theoretically predicted

length of double-stranded DNA, which encodes solution path as indicated in Figure 8 – 6 (a). The gel slab is firstly immersed in the TBE buffer. The corresponding DNA solutions are pipetted into the corresponding solution slots. After that, the gel slab is subjected to the electrophoresis by setting power of 18 *W*, voltage of 560 *V*, and current of 32 *mA*. The entire electrophoresis process is slightly longer as compared to that of DNA-based optimal route selection apparatus. It takes approximately 2 hours to complete. At the end of gel electrophoresis, by simply observing the migration lengths of DNA bands in *well 2*, *well 3*, and *well 4* to that of *well 5*. The DNA band in *well 2* has sharper resolution as compared to that of *well 3* and *well 4*. This is mainly because the concentration of DNA band in *well 2*, which has the same migration length as that of DNA band in *well 5*, is highest. Therefore, it is possible to obtain the result that the best strategy, as determined by biomolecular normative decision making apparatus, is likely to be option 1, which is same as the theoretically predicted result.

8.5 Conclusion

Biomolecular-based computing or simply DNA-based computing (if DNA molecules are used as information storing and processing substrate), has inestimable potentials to be extended into other seemingly-unrelated disciplines, and ultimately change the existing configurations in these fields. In previous two chapters, biomolecular-based method has already employed in strategic assignment problem and optimal route selection apparatus. In this chapter, the method has been extended to the discipline of normative decision theory. A case study, a regional military conflict in a board game risk® between two players, is used to illustrate how the method of biomolecular computing is capable of handling the situations based upon the expected utility hypothesis. It is believed that the DNA-based technique may facilitate effective complex decision-making in a rapid changing environment.

Chapter 9

Conclusions and Future Research

Mathematical Strategame Theory and Biomolecular Mediated Computing is a self-defined and fulfilled cross disciplinary research, which involves knowledge from seemingly unrelated disciplines, involving applied mathematics, financial engineering, graph theory, computer science, and molecular biology.

The entire dissertation, as specified by its title, can be further separated into two equal portions. Mathematical Strategame Theory is regarded as the theoretical framework of the research. As defined in Chapter 1, the word “Strategame” is a combination of two words – the former part of word “Strategy” and the word “Game”, where the term “Game” is originated from two games involved in Parrondo’s paradox, a special complementary to contemporary game theory. This part of the dissertation is therefore set out to investigate Parrondo’s paradox and other related paradoxical effects. Furthermore, it is aiming to extend the existing framework into other potential applications in the contemporary financial market.

Biomolecular Computing, on the other hand, is treated as the practical framework of the research. It is set out to develop a series of biomolecular mediated devices, which is capable of automatically processing, storing and displaying information, analogous to limited-function early computers.

In this final chapter, I will review the research contributions of this dissertation in section 9.1, and discuss the directions for future research in section 9.2.

9.1 Contributions

Mathematical Strategame Theory is the theoretical framework of the entire research. It is dedicated in the investigation of Parrondo's paradox as well as its associated paradoxical phenomenon and the extension of its original framework to other potential situations, where strategically thinking is desired. The followings are the main research contributions of theoretical framework of the dissertation.

- **A hierarchical framework** that describes the relationship between the theoretical framework, Mathematical Strategame Theory, and practical framework, Biomolecular Mediated Computing of the entire research (Chapter 1). This conclusive framework provides an organized view of the entire research work and a natural guide towards future research, as discussed in section 9.2.
- **A thorough investigation on Parrondo's paradox and related paradoxical phenomenon** (Chapter 2). Two versions of Parrondo's paradox is introduced and systematically analyzed in terms of a discrete-time Markov chain and a probability-based convex linear combination. The counter-intuitive phenomenon generated by two versions of Parrondo's paradox and related paradoxical phenomenon are reproduced in MATLAB and mathematically analyzed.
- **A study of other variations of Parrondo's paradox** (Chapter 3). The original version of Parrondo's paradox is mainly emphasized on a special combination of two initially losing games in the production of a winning expectation. A study is provided to investigate whether the remaining seven combinations, which are referred to as schemes, of two winning and/or losing games are capable of generating other paradoxical effects. Based on the simulation results, it is proved that three out of seven schemes are able to generate analogous effects. Typically, the result indicates two winning games may also lead to the production of a losing expectation, which is a reversed version of original Parrondo's paradox.
- **Verification of issues surrounding Parrondo's paradox** (Chapter 3). Two major issues proposed by other researchers are reassessed and verified. It is examined that the analogous paradoxical effects, delivered by Parrondo's paradox, cannot be simply reproduced by using one single controlling parameter, which is the capital of the player. The major problem associated with Parrondo's paradox is shown to be its potential applications. It can be regarded as an appealing demonstration on how random noises in

a system can alter the outcome of the entire system. However, it rarely has any possible applications to the current financial industry, such as constructing a well-diversified portfolio based upon the inspirations obtained from the framework of Parrondo's paradox, which is able to compete and even outperform the contemporarily most frequently used delta-neutral portfolio.

- **New conceptual model of financial option insurance** (Chapter 4). Although the mathematical model of Parrondo's paradox has very restricted applications in financial industry, the philosophy of controlling two isolated losing games by introducing one additional switching mechanism, known as pre-defined integer M (in capital-dependent Parrondo's paradox), leading to the production of winning expectation does inspire the thought of jumping out of the existing financial instruments as well as institutions. It ultimately leads to the creation of conceptual model of financial option insurance. The created model, unlike the traditional way of hedging, which utilizes financial option as a negatively correlated financial derivative to an existing portfolio, it can be issued by a completely independent financial institution, which is referred to as third entity (or option insurer). The operations of such institution are very alike to that of an insurance company: option holders pay insurance premium to option insurer in protecting their positions against any possible adverse market movements; meanwhile, the option insurer is engaged in the redistribution of risks among a larger pool of financial derivative investors and mitigates the loss of individuals. Two individual insurance policies have been proposed, as shown in section 5.2, with respect to two targeting groups of financial derivative (option) market participants – speculators and portfolio managers.
- **Newly designed mathematical model as a specific business strategy for the execution of financial option insurance business** (Chapter 4). The profitability of financial option insurer merely relies on seeking a pair of option investors with entirely reversed market expectations, *i.e.*, a call and a put of identical strike price and expiration date. Such case usually occurs under the perfect market assumption, where there is always a balance between two equal-sized parties of calls and puts, rather than in the real financial market. Therefore, in order to enhance the executional ability of business of option insurance, a mathematical model, known as business strategy of matching is proposed, which allows the matching of calls and puts of different moneyness.

Biomolecular Computing is the practical framework of the research. It is set out to develop a series of biomolecular mediated devices, which is capable of automatically tackling specific class of mathematical problems, which is analogous to limited-function early computers. In this dissertation, there are totally three biomolecular mediated devices have been proposed: automatic strategic assignment apparatus, optimal route selection apparatus, and normative decision making apparatus. The developed devices provide insights to the future research direction of biomolecular computer, which alike to the contemporary electronic-based general computer, which is able to process, store, and display the information. The followings are the main research contributions of practical framework of the dissertation.

- **A guide and a conclusive review of biomolecular computing** (Chapter 5). It briefly reviews the essential knowledge prior to the biomolecular computing as the study is a relatively new computing paradigm which utilizes artificially synthesized biomolecular structures, mainly nucleic acids and its analogs, as information storing and processing substrates to tackle computational problems. It describes the developed limited-function biomolecular mediated devices and the research trend of the entire discipline.
- **Conceptual design of biomolecular mediated device in solving strategic assignment problem** (Chapter 6). In the theoretical framework, a mathematical model, as specified in section 5.3, has been proposed to enhance the executional ability of option insurance business when facing imperfect market situations. The proposed model, which can be mathematical classified in terms of a specific class of combinatorial optimization problems, namely strategic assignment problem. A conceptual design of biomolecular mediated device is proposed to handle the problem. The detailed executional procedures are described from information encoding (represent graphical information in terms of DNA sequences) to a combination of in-vitro operational methods. The specifically designed algorithm, on the other hand, reveals the philosophical difference between conventional silicon-based digital computing and biomolecular mediated computing, that is, the biomolecular mediated devices is theoretically more effective in handling the mathematical problems, in which the massive searching ability is demanded by the algorithm.
- **A functional biomolecular mediated optimal route selection apparatus** (Chapter 7). After the theoretical foundation, a conceptual design of biomolecular mediated device in solving strategic assignment problem, as indicated in Chapter 6, has been established, it is desirable to experimentally demonstrate the feasibility of biomolecular computing by

creating a workable biomolecular mediated apparatus. In Chapter 7, a fully functional biomolecular optimal route selection apparatus has been developed. The developed device is able to “store” information in its unique “database”. By manually assigning two physical locations, which is treated as user input, to the device, it is able to automatically process the information and display the obtained result in terms of a 2D image of 12% native acrylamide gel. The device maximizes the advantage of biomolecular computing, that is, massive parallelism. The introduced length-based encoding scheme as biasing parameter is experimentally proved to be superior to other existing methods, such as the ones involving pH value and solution concentration.

- **Conceptual design of biomolecular decision making apparatus** (Chapter 8). The potential of biomolecular computing is further theoretically extended into another seemingly unrelated discipline, namely normative decision theory, in the creation of normative decision making apparatus, by introducing one additional in-vitro biomolecular mechanism, namely toe-hold mediated strand displacement. It has demonstrated the biomolecular mediated computing has inestimable potentials to be extended into other seemingly-unrelated disciplines, and ultimately change the existing configurations on these fields.

9.2 Recommendations for Future Research

In the theoretical framework of this dissertation, Mathematical Strategame Theory, a thorough investigation on Parrondo's paradox, related paradoxical phenomenon, issues surrounding Parrondo's paradox is provided. It is proved Parrondo's paradox is a perfect mathematical model on demonstrating how random noises can alter the ultimate outcome of the entire system. However, due to its special mathematical structure, that is, game B of Parrondo's paradox is a second-order Markov chain and the switching mechanism is either capital-dependent or history-dependent scenarios, the potential applications is really restricted in the provoked disciplines such as game theory and financial engineering. In order to apply the framework of Parrondo's paradox to these disciplines, it is believed the switching mechanism, the predefined integer and history-dependent scenarios have to be modified. Beyond the provoked disciplines, it is possible to treat the counterintuitive phenomenon generated by Parrondo's paradox as an appealing demonstration of the importance of fuzzy logic. The fuzzy logic is a multi-valued logic which has a truth table varying between 0 and 1. In the scheme of traditional logic, game A and game B are defined as 0 as both of them are losing games. No matter what additional conditions applied to these games, the resulting mixed games is always a losing game, which is a 0. However, the losing games in the scheme of fuzzy logic can be defined as losing expectation producers, whereas the information of individual trials are perfectly preserved. Therefore, by introducing one additional parameter, the reversing outcome is able to be generated.

Prior to the implementation of developed conceptual model of financial option insurance, as a new financial instrument, in contemporary financial market, there is a lot of future works need to be fulfilled by researchers from various disciplines. First of all, an appropriate pricing structure of financial option insurance should be developed. The yardstick of 50%, which is used in the case study of section 5.2, reveals the probabilities of price of underlying security either rises or drops are completely identical. In reality, this value should always be tweaked according to the condition of the prevailing financial market. Therefore, an appropriate pricing structure should be developed to determine the actual value of the yardstick. Although a mathematical model, namely business strategy of matching, is proposed in section 5.3, which enhances the executional ability of business of financial option insurance, a much more sophisticated mathematical model is recommended to be developed by taking following parameters, including expiration date, strike price, option premium, the states of the position (whether it is a naked position or a protected position), into the consideration. Based upon the developed mathematical models, an effective

system of business execution is suggested to be established, and ultimately, the system should be programmed in terms of a fully functional network- based trading platform.

In the practical framework of this dissertation, Biomolecular Mediated Computing, three biomolecular mediate devices have been proposed. The future research of these devices should be emphasized on the continuous improvement of the efficiencies on the developed algorithms.

In recent years, the emphasize of researches in biomolecular mediated computing are gradually moving from the development of biomolecular mediated limited-function apparatus into the creation of conceptual model of biomolecular mediated logic gates, which is analogous to the history if evolvement of electronic-based digital computer. These proposed conceptual models can be broadly classified in terms of two classes, namely *biochemically controlled logic circuits* and *strand displacement mediated logic circuits*. These two types of conceptual models of biomolecular mediated logic gates have their corresponding advantages. Unfortunately, neither of them has any future potentials due to their natural limitations.

The biochemically controlled logic circuits require the involvement of a specifically designed enzyme, called DNA-cleaving deoxyribozyme, during the realization of Boolean logic operations. This class of individual logic gates generally consists of three essential elements: input signal (specifically designed nucleic acid structure), logic operator (deoxyribozyme), and output signal (chemically modified or non-chemically modified nucleic acid structure). The biochemically controlled logic circuits have ultra-fast processing speed. With the introduction of high concentration specifically designed enzyme, the corresponding reaction speed is generally in split seconds. However, using enzyme as logic operator has one significant drawback, that is, the enzyme itself is counted as the “noise signal” to the logic circuits. In simple words, such “noise signal” is likely to be carried by output signal towards the subsequent logic operators. The amount of “noise signal” is likely to be accumulated and ultimately interrupt the subsequent logic operations. To date, there is no workable method to perfectly remove such “noise signal” from solution.

The strand displacement mediated logic circuits rely on the toe-hold mediated strand displacement mechanism. Such mechanism has also been employed in our conceptual design of biomolecular decision making apparatus (Chapter 7). As compared to biochemically controlled logic circuits, it successfully avoids the introduction of foreign ingredients (“noise signal”) as its input signal, logic operator, and output signal are all nucleic acids. Therefore, such type of logic circuits is always referred to as enzyme-free logic circuit. Unfortunately, the potential of strand

displacement mediated logic circuits is very restricted. It is because the process of toe-hold mediated strand displacement is relatively slow. In addition to that, a three-level of logic circuit composed of eight logic gates comprises 130 different oligonucleotides. It is hard to imagine there will be sufficient amount of nucleic acid in this planet to construct a general purpose biomolecular mediated computer.

The major drawback of previous two conceptual designs is mainly because of the involvement of foreign ingredients (for biochemically controlled logic circuits) and the usage of slow strand displacement mechanism in the design of logic operators (for strand displacement mediated logic circuits). In the future research of biomolecular mediated logic circuits; it is optimal to avoid these two components. To the best of my knowledge, there are several parameters that can be simply manipulated without the involvement of foreign ingredients and deliver the reasonable speed of information processing: solution temperature and concentration.

Most recently, six different sugar backbones nucleic acid analogues, abbreviated XNAs, have been successfully replicated, known as arabinose (ANA), 2'-fluoroarabinose (FNA), threose (TNA), "locked" ribose (LNA), cyclohexene (CeNA), and HNA. These analogues of nucleic acids have their unique physical properties. For LNA, it improves the overall thermal stability for the product of hybridization while obeying the standard Watson-Crick base pairing rules. Such unique physical property of LNA can be coupled with original DNA in the construction of, let's say, LNA: DNA mediated logic circuit. Such logic circuit design is established based upon the fact that the site-specific hybridization of DNA:DNA and DNA:LNA are likely to be affected by two physical factors as mentioned above, that is, solution temperature (thermal force) and concentration. At low temperature setting, the hybridization procedure is dominated by solution concentration. At high temperature setting, the hybridization procedure is dominated by thermal forces. By employing solution temperature as one input signal and presence or absence of LNA as another input signal, it is possible to develop the conceptual model of DNA: LNA mediated logic circuit.

In reality, the discipline of biomolecular mediated computing is closely related to the field of synthetic biology. With more and more synthetic methods, biomolecular mechanisms, and structures being invented, there will be lots of intriguing designs of biomolecular mediated logic gates being developed, accordingly. Among these conceptual models, there should be at least one acceptable model, which can be used in the development of biomolecular mediated logic circuits, and ultimately, general purpose biomolecular mediated computer.

Appendix

1. Capital-dependent Parrondo's paradox

Game A.m

```
%-----Function of game A-----%

function [Capital] = GameA(Capital, Bet, p1)

% Generate random number in the range of 0 to 1
coinFlip = rand(1);

% Check whether the result of the game is a win or loss of capital in
the perspective of the player
if (coinFlip < p1)
    Capital = Capital + Bet;
else
    Capital = Capital - Bet;

end
```

Game B. m

```
%-----Function of game B-----%

function [Capital] = GameB(Capital, Bet, M, p2, p3)

% Choosing which scenario will be followed in Game B
if (mod(Capital,M)==0)
    probB = p2;
else
    probB = p3;
end

% Generate random number in the range of 0 and 1
coinFlip = rand(1);

% Check whether the result of the game is a win or loss of capital in
the perspective of the player
if (coinFlip < probB)
    Capital = Capital + Bet;
else
    Capital = Capital - Bet;
```


238

end

end

Capital dependent Parrondos paradox.m

```
%-----Capital-dependent Parrondo's paradox-----%

clc; clear;

%-----Variables-----%

T = 10000;           % Number of trials (fixed)
G = 200;             % Number of bets made per trail (fixed)
Bet = 1;             % Amount of bet each time (fixed)
M = 3;               % Predefined integer M (fixed)
e = 0.005;           % Epsilon (biasing parameter, fixed)
p1 = 0.5 - e;        % Probability of winning of game A
p2 = 0.1 - e;        % Probability of winning of game B (Scenario
1)                   %
p3 = 0.75 - e;       % Probability of winning of game B (Scenario
2)                   %
RecA = zeros(G, 1);  % Record of capital fluctuations (game A)
RecB = zeros(G, 1);  % Record of capital fluctuations (game B)
RecR = zeros(G, 1);  % Record of capital fluctuations (random)
r = 0.5;             % Randomly mixing parameter (gamma, fixed)

count = 0;           % Counter initialization
game = 0;            % If game == 0, game A; If game == 1, gameB

%-----Game A-----%

for t = 1:T
    Capital = 0;
    for g = 1:G
        Capital = GameA(Capital, Bet, p1);
        RecA(g) = RecA(g) + Capital;
    end
end

%-----Game B-----%

for t = 1:T
    Capital = 0;
    for g = 1:G;
        Capital = GameB(Capital, Bet, M, p2, p3);
        RecB(g) = RecB(g) + Capital;
    end
end

%-----Randomly mix of game A and B-----%
```

```

for t = 1:T
    Capital = 0;
    for g = 1:G
        randNo = rand(1);
        if (randNo <= r)
            Capital = GameA(Capital, Bet, p1);
        else
            Capital = GameB(Capital, Bet, M, p2, p3);
        end
        RecR(g) = RecR(g) + Capital;
    end
end

%-----Result representation (in graph)-----%

axis([0 200 -2 2])
plot(1:G, RecA/T, 'm-', 'LineWidth', 2)
hold on
plot(1:G, RecB/T, 'b-', 'LineWidth', 2)
plot(1:G, RecR/T, 'k-', 'LineWidth', 2)
hold off
title(sprintf('Capital-dependent coin-tossing games (averaged over %d
trials)', T))
xlabel('Number of games played')
ylabel('Capital')

```

2. History-dependent Parrondo's paradox

History dependent Parrondos paradox.m

```
%-----History-dependent Parrondo's paradox-----%

clear; clc;

%---Defined variables---%

T = 50000;           %Total number of trials (fixed)
G = 200;             %Number of bets made per trail(fixed)
Bet = 1;             %Amount of bet each time
e = 0.005;           %Epsilon (biasing parameter, fixed)
p1 = 0.5 - e;         %Winning probability of game A
p2 = 0.9 - e;         %Winning probability of game B (Scenario 1)
p3 = 0.25 - e;        %Winning probability of game B (Scenario 2)
p4 = 0.25 - e;        %Winning probability of game B (Scenario 3)
p5 = 0.7 - e;         %Winning probability of game B (Scenario 4)

CapitalA = zeros(G,1); %Initialize the capital of game A
CapitalB = zeros(G,1); %Initialize the capital of game B
CapitalR = zeros(G,1); %Initialize the capital of randomly mixed game
a = 1; b = 1;         %a/b = +1 or -1, standing for previous and
previous game results
count = 0; game = 0;

%---Game A---%
for t = 1 : T
    Capital = 0;
    for g = 1 : G
        coinFlip = rand(1);
        if (coinFlip < p1)
            Capital = Capital + Bet;
        else
            Capital = Capital - Bet;
        end
        CapitalA(g) = CapitalA(g) + Capital;
    end
end

%---Game B---%
for t = 1 : T
    Capital = 0;
    for g = 1 : G
        coinFlip = rand(1);
        %Lose, Lose}
        if (a == -1)&&(b == -1)
            if (coinFlip < p2)
                Capital = Capital + Bet;
            end
        end
    end
end
```

```

        a = b;
        b = 1;
    else
        Capital = Capital - Bet;
        a = b;
        b = -1;
    end
    CapitalB(g) = CapitalB(g) + Capital;

    %{Lose, Win}
    elseif (a == -1)&&(b == 1)
        if (coinFlip < p3)
            Capital = Capital + Bet;
            a = b;
            b = 1;
        else
            Capital = Capital - Bet;
            a = b;
            b = -1;
        end
        CapitalB(g) = CapitalB(g) + Capital;

    %{Win, Lose}
    elseif (a == 1)&&(b == -1)
        if (coinFlip < p4)
            Capital = Capital + Bet;
            a = b;
            b = 1;
        else
            Capital = Capital - Bet;
            a = b;
            b = -1;
        end
        CapitalB(g) = CapitalB(g) + Capital;

    %{Win, Win}
    else
        if (coinFlip < p5)
            Capital = Capital + Bet;
            a = b;
            b = 1;
        else
            Capital = Capital - Bet;
            a = b;
            b = -1;
        end
        CapitalB(g) = CapitalB(g) + Capital;
    end
end
end

end

%Randomly mix of game A and game B
for t = 1 : T
    Capital = 0;

```

```

for g = 1 : G
    randNo = rand(1);
    if (randNo <= 0.5)
        coinFlip = rand(1);
        if (coinFlip < p1)
            Capital = Capital + Bet;
        else
            Capital = Capital - Bet;
        end
        CapitalR(g) = CapitalR(g) + Capital;

    else
        %{Lose, Lose}
        if (a == -1)&&(b == -1)
            if (coinFlip < p2)
                Capital = Capital + Bet;
                a = b;
                b = 1;
            else
                Capital = Capital - Bet;
                a = b;
                b = -1;
            end
            CapitalR(g) = CapitalR(g) + Capital;
            %{Lose, Win}
        elseif (a == -1)&&(b == 1)
            if (coinFlip < p3)
                Capital = Capital + Bet;
                a = b;
                b = 1;
            else
                Capital = Capital - Bet;
                a = b;
                b = -1;
            end
            CapitalR(g) = CapitalR(g) + Capital;
            %{Win, Lose}
        elseif (a == 1)&&(b == -1)
            if (coinFlip < p4)
                Capital = Capital + Bet;
                a = b;
                b = 1;
            else
                Capital = Capital - Bet;
                a = b;
                b = -1;
            end
            CapitalR(g) = CapitalR(g) + Capital;
            %{Win, Win}
        elseif (a == 1)&&( b == 1)
            if (coinFlip < p5)
                Capital = Capital + Bet;
                a = b;
                b = 1;
            end
        end
    end
end

```

```

        else
            Capital = Capital - Bet;
            a = b;
            b = -1;
        end
        CapitalR(g) = CapitalR(g) + Capital;
    end
end
end

%-----Result representation (in graph)-----%
plot(CapitalA/T, 'b-', 'LineWidth', 2)
hold on
plot(CapitalB/T, 'm-', 'LineWidth', 2)
plot(CapitalR/T, 'k-', 'LineWidth', 2)
hold off
title('History-dependent coin-tossing games (averaged over %d trials)',
T)
xlabel('Number of games played')
ylabel('Capital')

```

3. Variations of paradoxical effect

Variations of paradoxical effect.m

```
%-----Variations of Paradoxical effect-----%

clc; clear;

%-----Variables-----%

T = 10000;           % Number of trials (fixed)
G = 200;             % Number of bets made per trail (fixed)
Bet = 1;             % Amount of bet each time (fixed)
M = 3;               % Predefined integer M (fixed)
e = 0.005;           % Epsilon (biasing parameter, fixed)
%Various Probabilities of p1,p2,and p3 with respect to various schemes%
p1 = 0.5 - e;         % Probability of winning of game A
p2 = 0.1 - e;         % Probability of winning of game B (Scenario
1)
p3 = 0.75 - e;        % Probability of winning of game B (Scenario
2)
%Scheme #2%
p1 = 0.5 + e; p2= 0.1 - e; p3 = 0.75 - e;
%Scheme #3%
p1 = 0.5 - e; p2= 0.9 + e; p3 = 0.75 - e;
%Scheme #4%
p1 = 0.5 - e; p2= 0.1 - e; p3 = 0.25 + e;
%Scheme #5%
p1 = 0.5 + e; p2= 0.9 + e; p3 = 0.75 - e;
%Scheme #6%
p1 = 0.5 + e; p2= 0.1 - e; p3 = 0.25 + e;
%Scheme #7%
p1 = 0.5 - e; p2= 0.9 + e; p3 = 0.25 + e;
%Scheme #8%
p1 = 0.5 + e; p2= 0.9 + e; p3 = 0.25 + e;

%-----
--%
RecA = zeros(G, 1);   % Record of capital fluctuations (game A)
RecB = zeros(G, 1);   % Record of capital fluctuations (game B)
RecR = zeros(G, 1);   % Record of capital fluctuations (random)
r = 0.5;              % Randomly mixing parameter (gamma, fixed)

count = 0;            % Counter initialization
game = 0;             % If game == 0, game A; If game == 1, gameB

%-----Game A-----%
```



```

for t = 1:T
    Capital = 0;
    for g = 1:G
        Capital = GameA(Capital, Bet, p1);
        RecA(g) = RecA(g) + Capital;
    end
end

%-----Game B-----%

for t = 1:T
    Capital = 0;
    for g = 1:G;
        Capital = GameB(Capital, Bet, M, p2, p3);
        RecB(g) = RecB(g) + Capital;
    end
end

%-----Randomly mix of game A and B-----%

for t = 1:T
    Capital = 0;
    for g = 1:G
        randNo = rand(1);
        if (randNo <= r)
            Capital = GameA(Capital, Bet, p1);
        else
            Capital = GameB(Capital, Bet, M, p2, p3);
        end
        RecR(g) = RecR(g) + Capital;
    end
end

%-----Result representation (in graph)-----%

axis([0 200 -2 2])
plot(1:G, RecA/T, 'm-', 'LineWidth', 2)
hold on
plot(1:G, RecB/T, 'b-', 'LineWidth', 2)
plot(1:G, RecR/T, 'k-', 'LineWidth', 2)
hold off
title(sprintf('Capital-dependent coin-tossing games (averaged over %d
trials)', T))
xlabel('Number of games played')
ylabel('Capital')

```

4. Simple capital-dependent game

Simple capital dependent game.m

```
%-----Simple capita-dependent game-----%
clear; clc;

%-----Variables-----%

G = 199; %Total number of games
CapitalA = zeros(200); %Initialize capital of game A
CapitalA(1)= 0; %Starting capital of game A
CapitalB = zeros(200); %Initialize capital of game B
CapitalB(1)= 1; %Starting capital of game B
CapitalD = zeros(200); %Initialize capital of game
ABABABAB... %Starting capital of game ABABABAB...
CapitalD(1)= 0; %Initialize capital of randomized
game %Starting capital of randomized game
CapitalR(1) =0; %Mixing count
count = 0; %Total games as displayed in graph
NumGame = 1:200;

%Game A
for g = 1 : G
    if (mod(CapitalA(g),2)==0)
        CapitalA(g+1) = CapitalA(g) - 1;
    else
        CapitalA(g+1) = CapitalA(g) - 2;
    end
end

%Game B
for g = 1 : G
    if (mod(CapitalB(g),2)==0)
        CapitalB(g+1) = CapitalB(g) - 7;
    else
        CapitalB(g+1) = CapitalB(g) + 6;
    end
end

CapitalB(1:200)

%Deterministically mix of game A and game B
for g = 1 : G
    if (count==0)
        if (mod(CapitalD(g),2)==0)
            CapitalD(g+1) = CapitalD(g) - 1;
```

```

        else
            CapitalD(g+1) = CapitalD(g) - 2;
        end
        count=1;
    else
        if (mod(CapitalD(g),2)==0)
            CapitalD(g+1) = CapitalD(g) - 7;
        else
            CapitalD(g+1) = CapitalD(g) + 6;
        end
        count=0;
    end
end
end

%Randomized mix of game A and game B
for g = 1 : G
    RandNum = rand(1);
    if (RandNum <= 0.5)
        if (mod(CapitalR(g),2)==0)
            CapitalR(g+1) = CapitalR(g) - 1;
        else
            CapitalR(g+1) = CapitalR(g) - 2;
        end
    else
        if (mod(CapitalR(g),2)==0)
            CapitalR(g+1) = CapitalR(g) - 7;
        else
            CapitalR(g+1) = CapitalR(g) + 6;
        end
    end
end
end

plot (NumGame, CapitalA(1:200), 'm-', 'LineWidth', 2)
hold on
plot (NumGame, CapitalB(1:200), 'b-', 'LineWidth', 2)
plot (NumGame, CapitalR(1:200), 'k-', 'LineWidth', 2)
hold off
title('Simple captial-dependent game')
xlabel('Number of games played')
ylabel('Capital')

```

Reference

1. J. V. Neumann and O. Morgenstern, "[Theory of games and economic behavior](#)", *Princeton University Press*, 1944.
2. J. Prost, J. F. Chauwin, L. Peliti, and A. Ajdari, "[Asymmetric pumping of particles](#)", *Physical Review Letters*, Vol.: 72, Issue: 16, pp. 2652 – 2655.
3. J. M. R Parrondo and P. Espanol, "[Criticism of Feynman's analysis of the ratchet as an engine](#)", *American Journal of Physics*, Vol.: 64, Issue: 9, pp. 1125 – 1130, 1996.
4. G. P. Harmer and D. Abbott, "[Game theory - Losing strategies can win by Parrondo's paradox](#)", *Nature*, Vol.: 402, Issue: 6764, pp. 864, 1999.
5. G. P. Harmer and D. Abbott, "[Parrondo's paradox](#)", *Statistical Science*, Vol.: 14, Issue: 2, pp.206 – 213, 1999.
6. G. P. Harmer, D. Abbott, and P. G. Taylor, "[The paradox of Parrondo's games](#)", *Proceedings of the Royal Society A*, Vol.: 456, Issue: 1994, pp.247 – 259, 2000.
7. G. P. Harmer, D. Abbott, P. G. Taylor, "[Brownian ratchets and Parrondo's games](#)", *Chaos*, Vol.: 11, Issue: 3, pp. 705 – 714, 2001.
8. J. M. R. Parrondo, G. P. Harmer, and D. Abbott, "[New paradoxical games based on Brownian ratchets](#)", *Physical Review Letters*, Vol.: 85, Issues: 24, pp. 5226 – 5229, 2000.
9. R. Toral, "[Cooperative Parrondo's games](#)", *Fluctuation and Noise Letters*, Vol.: 1, Issue: 1, pp. L7 – L12, 2001.
10. A.P. Flitney, J. Ng, and D. Abbott, "[Quantum Parrondo's games](#)", *Physica A*, Vol.: 314, Issue: 1 – 4, pp. 35 – 42, 2002
11. A. P. Flitney, and D. Abbott, "[Quantum models of Parrondo's games](#)", *Physica A*, Vol.: 324, Issue: 1 – 2, pp. 152 – 156, 2003.
12. M. Boman, S. J. Johansson, and D. Lyback, "[Parrondo's strategies for artificial traders](#)", *Intelligent Agent Technology*, pp.150 – 159, 2001.
13. Von Smoluchowski M., "[Experimentall nachweisbare, der ü blichen thermodynamic widersprechende molekularchänomene](#)", *Physikalische Zeitschrift*, XIII, pp. 1069 – 1080, 1912.
14. Feynman R. P., Leighton R. B., and Sands M., "[The Feynman lectures on physics](#)", Vol.: 1, pp. 46.1 – 46.9, 1963.

15. C. R. Doering, "[Randomly rattled ratchets](#)", *Nuovo Cimento Della Societa Italiana Di Fisica D*, Vol.: 17, Issue: 7 – 8, pp. 685 – 697, 1995.
16. S. Goldberg, "[Probability: an introduction](#)", *Dover Publications*, 1987.
17. R. D. Yates and D. J. Goodman, "[Probability and stochastic processes: a friendly introduction for electrical and computer engineers](#)", *John Wiley & Sons*, 2nd Edition, Chapter 12, 2005.
18. J. R. Norris, "[Markov chains](#)", *Cambridge University Press*, 1997.
19. A. Allison, and D. Abbott, "[Control systems with stochastic feedback](#)", *Chaos*, Vol.: 11, Issue: 3, pp. 715 – 724, 2001.
20. J. Buceta, K. Lindenberg, and J. M. R. Parrondo, "[Stationary and oscillatory spatial patterns induced by global periodic switching](#)", *Physical Review Letters*, Vol.: 88, Issue: 2, Article number: 024103, 2002.
21. D. A. Meyer, and H. Blumer, "[Parrondo games as lattice gas automata](#)", *Journal of Statistical Physics*, Vol.: 107, Issue: 1 – 2, pp. 225 – 239, 2002.
22. R. Toral, "[Capital redistribution brings wealth by Parrondo's paradox](#)", *Fluctuation and Noise Letters*, Vol.: 2, Issue: 4, pp. L305 – L311, 2002.
23. F.J. Cao, L. Dinis, and J. M. R. Parrondo, "[Feedback control in a collective flashing ratchet](#)", *Physical Review Letters*, Vol.: 93, Issue: 4, Article number: 040603, 2004.
24. P. Amengual, and R. Toral, "[Transfer of information in Parrondo's games](#)", *Fluctuation and Noise Letters*, Vol.: 5, Issue: 1, pp. L63 – L71, 2005.
25. D. M. Wolf, V. V. Vazirani, and A. P. Arkin, "[Diversity in times of adversity: probabilistic strategies in microbial survival games](#)", *Journal of Theoretical Biology*, Vol.: 234, Issue: 2, pp. 227 – 253, 2005.
26. A. G. Allison, C. E. M. Pearce, and D. Abbott, "[Finding keywords amongst noise: Automatic text classification without parsing](#)", *Proceedings of the Society of Photo Optical Instrumentation Engineers*, Vol.: 6601, pp. U275 – U286, 2007.
27. I. Peterson, "[Losing to win](#)", *Science News*, Vol.: 157, Issue: 3, pp.47, 2012.
28. J. L. Casti, "[Losing games for your winning plays](#)", *Complexity*, Vol.: 6, Issue: 6, pp. 11 – 14, 2001.
29. A. Costa, M. Fackrell, and P. G. Taylor, "[Two issues surrounding Parrondo's paradox](#)", *Annals of the International Society of Dynamic Games*, Vol.: 7, pp. 599 – 609, 2005.
30. T. K. Philips and A.B. Feldman, "[Parrondo's paradox is not paradoxical](#)", *Social Science Research Network*, Classification A19, A20, Z00, 2004.

31. H. Martin, and H. C. von Baeyer, "[Simple games to illustrate Parrondo's paradox](#)", *American Journal of Physics*, Vol.: 72, Issue: 5, pp. 710 – 714, 2004.
32. R. Iyengar and R. Kohli, "[Why Parrondo's paradox is irrelevant for utility theory, stock buying, and the emergence of life](#)", *Complexity*, Vol.: 9, Issue: 1, 2004.
33. R. Chepesiuk, "[Where the chips fall: environmental health in the semiconductor industry](#)", *Environmental Health Perspectives*, Vol.: 107, Issue: 9, pp. A452 – A457, 1999.
34. G. E. Moore, "[Cramming more components onto integrated circuits](#)", *Proceedings of the IEEE*, Vol.: 86, Issue: 1, pp. 82 – 85, 1998. (Reprinted from *Electronics*, pp. 114 – 117, 1965).
35. N. C. Seeman, "[DNA in a material world](#)", *Nature*, Vol.: 42, Issue: 6921, pp. 427 – 431, 2003.
36. J. D. Watson, and F. H. C. Crick, "[Molecular structure of nucleic acids – A structure for deoxyribose nucleic acid](#)", *Nature*, Vol.: 171, Issue: 4356, pp. 737 – 738, 1953.
37. L. M. Adleman, "[Molecular computation of solutions to combinatorial problems](#)", *Science*, Vol.: 266, Issue: 5187, pp. 1021 – 1024, 1994.
38. R. J. Lipton, "[DNA solution of hard computational problems](#)", *Science*, Vol.: 268, No.: 5210, pp. 542 – 545, 1995.
39. Q. Ouyang, P. D. Kaplan, S. M. Liu, and A. Libchaber, "[DNA solution of the maximal clique problem](#)", *Science*, Vol.: 278, Issue: 5337, pp. 446 – 449, 1997.
40. F. Guarnieri, M. Fliss, and C. Bancroft, "[Making DNA add](#)", *Science*, Vol.: 273, Issue: 5272, pp. 220 – 223, 1996.
41. Y. Masahito, M. Bobuo, S. Toshikazu, and O. Azuma, "[DNA solution of the shortest path problem by concentration control](#)", *Genome Informatics*, Vol.: 11, pp. 466 – 467, 2000.
42. Z. X. Yin, F. Y. Zhang, and J. Xu, "[A Chinese postman problem based on DNA computing](#)", *Journal of Chemical Information and Computer Sciences*, Vol.: 42, Issue: 2, pp. 222 – 224, 2002.
43. S. Y. Wang, "[DNA computing of bipartite graphs for maximum matching](#)", *Journal of Mathematical Chemistry*, Vol.: 31, Issue: 3, pp. 271 – 279, 2002.
44. J. Y. Lee, S. Y. Shin, T. H. Park, and B. T. Zhang, "[Solving traveling salesman problems with DNA molecules encoding numerical values](#)", *Biosystems*, Vol.: 78, Issue: 1 – 3, pp. 39 – 47, 2004.

45. D. M. Xiao, W. X. Li, and Z. Z. Zhang, and L. He, "[Solving maximum cut problems in the Adleman-Lipton model](#)", *Biosystems*, Vol.: 82, Issue: 3, pp.203 – 207, 2005.
46. A. S. C. Alberto, and N. Y. Soma "[A polynomial-time DNA computing solution for the bin-packing problem](#)", *Applied Mathematics and Computation*, Vol.: 215, Issue: 6, pp.2055 – 2062, 2009.
47. R. Deaton, M. Garzon, R. C. Murphy, J. A. Rose, D. R. Franceschetti, and S. E. Stevens, "[Reliability and efficiency of a DNA-based computation](#)", *Physical Review Letters*, Vol.: 80, Issue: 2, pp. 417 – 420, 1998.
48. C. P. Collier, E. W. Wong, M. Belohradsky, F. M. Raymo, J. F. Stoddart, P. J. Kuekes, R. S. Williams, and J. R. Heath, "[Electronically configurable molecular-based logic gates](#)", *Science*, Vol.: 285, Issue: 5426, pp. 391 – 394, 1999.
49. C. D. Mao, T. H. LaBean, J. H. Reif, and N. C. Seeman, "[Logical computation using algorithmic self-assembly of DNA triple-crossover molecules](#)", *Nature*, Vol.: 407, Issue: 6803, pp. 493 – 496, 2000.
50. P. Wasiewicz, A. Malinowski, R. Nowak, J. J. Mulawka, P. Borsuk, P. Weglenski, and A. Plucienniczak, "[DNA computing: implementation of data flow logical operations](#)", *Future Generation Computer Systems*, Vol.: 17, Issue: 4, pp.361 – 378, 2001.
51. M. N. Stojanovic, T. E. Mitchell, and D. Stefanovic, "[Deoxyribozyme-based logic gates](#)", *Journal of the American Chemical Society*, Vol.: 124, Issue: 14, pp. 3555 – 3561, 2002.
52. V. Balzani, A. Credi, and M. Venturi, "[Molecular logic circuits](#)", *CHEMPHYSCHEM*, Vol.: 4, Issue: 1, pp. 49 – 59, 2003.
53. Q. H. Liu, L. M. Wang, A. G. Frutos, A. E. Condon, R. M. Corn, and L. M. Smith, "[DNA computing on surfaces](#)", *Nature*, Vol.: 403, Issue: 6766, pp. 175 – 179, 2000.
54. A. Saghatelian, N.H. Volcker, K. M. Guckian, V. S. Y. Lin, and M. R. Ghadiri, "[DNA-based photonic logic gates: AND, NAND, and INHIBIT](#)", *Journal of American Chemical Society*, Vol.: 125, Issue: 2, pp. 346 – 347, 2003.
55. A. Okamoto, K. Tanaka, and I. Saito, "[DNA logic gates](#)", *Journal of American Chemical Society*, Vol.: 126, Issue: 30, pp. 9458 – 9463, 2004.
56. X. Chen, Y. F. Wang, Q. Liu, Z. Z. Zhang, C. H. Fan, and L. He, "[Construction of molecular logic gates with a DNA-cleaving deoxyribozyme](#)", *Angewandte Chemie International Edition*, Vol.: 45, Issue: 11, pp. 1759 – 1762, 2006.
57. N. H. Voelcker, K. M. Guckian, A. Saghatelian, and M. R. Ghadiri, "[Sequence-addressable DNA logic](#)", *Small*, Vol.: 4, Issue: 4, pp. 427 – 431, 2008.

58. K. S. Park, C. Jung, and H. G. Park, "['Illusionary' polymerase activity triggered by metal ions: use for molecular logic gate operations](#)", *Angewandte Chemie International Edition*, Vol.: 49, No. 50, pp. 9757 – 9760, 2010.
59. A. Lake, S. Shang, and D. M. Kolpashchikov, "[Molecular logic gates connected through DNA four-way junctions](#)", *Angewandte Chemie International Edition*, Vol.: 49, Issue: 26, pp. 4459 – 4462, 2010.
60. J. Elbaz, O. Lioubashevski, F. Wang, F. Remacle, R. D. Levine, and I. Willner, "[DNA computing circuits using libraries of DNzyme subunits](#)", *Nature Nanotechnology*, Vol.: 5, Issue: 6, pp.417 – 422, 2010.
61. A. Tamsir, J. J. Tabor, and C. A. Voigt, "[Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'](#)", *Nature*, Vol.: 469, Issue: 7329, pp. 212 – 215, 2011.
62. S. Regot, J. Macia, N. Conde, K. Furukawa, J. Kjellen, T. Peeters, S. Hohmann, E.de Nadal, F. Posas, and R. Sole, "[Distributed biological computation with multicellular engineered networks](#)", *Nature*, Vol.: 469, Issue: 7329, pp. 207 – 211, 2011.
63. Brian M. Frezza, Scott L. Cockroft, and M. Reza Ghadiri, "[Modular multi-level circuits from immobilized DNA-based logic gates](#)", *Journal of American Chemical Society*, Vol.: 129, Issue: 48, pp. 14875 – 14879, 2007.
64. F. Xia, X. L. Zuo, R. Q. Yang, R. J. White, Y. Yao, D. Kang, X. Gong, A. A. Lubin, A. V. Belisle, J. D. Yuen, B. Y. B. Hsu, and K. W. Plaxco, "[Label-free, dual-analyte electrochemical biosensors: a new class of molecular-electronic logic gates](#)", *Journal of the American Chemical Society*, Vol.: 132, Issue: 25, pp. 8557 – 8559, 2010.
65. L. L. Qian, W. Erik, "[Scaling up digital circuit computation with DNA strand displacement cascades](#)", *Science*, Vol.: 332, Issue: 6034, pp. 1196 – 1201, 2011.
66. B. L. Li, J. Jang, C. Xi, and A. D. Ellington, "[Probing spatial organization of DNA strands using enzyme-free hairpin assembly circuits](#)", *Journal of the American Chemical Society*, Vol.: 134, Issue: 34, pp. 13918 – 13921, 2012.
67. L. L. Qian, W. Erik, and J. Bruck, "[Neural network computation with DNA strand displacement cascades](#)", *Nature*, Vol.: 475, Issue: 7356, pp. 368 – 372, 2011.
68. A. J. Genot, J. Bath, and A. J. Turberfield, "[Reversible logic circuits made of DNA](#)", *Journal of American Chemical Society*, Vol.: 133, Issue: 50, pp. 20080 – 20083, 2011.
69. Condon, A., "[Molecular programing DNA and the brain](#)", *Nature*, Vol.: 475, Issue: 7356, pp. 304 – 305, 2011.

70. Genot, A. J., Bath, J. and Turberfield, A. J., "[Reversible logic circuits made of DNA](#)", *Journal of the American Chemical Society*, Vol.: 133, Issue: 50, pp. 20080 – 20083, 2011.
71. Chakraborty, B., Jonoska, and N., Seeman, N. C., "[A programmable transducer self-assembled from DNA](#)", *Chemical Science*, Vol.: 3, Issue: 1, pp. 168 – 176, 2012.
72. Cornett, E. M., Campbell, E. A., Gulenay, G., Peterson, E., Bhaskar, N., and Kolpashchikov, D. M., "[Molecular logic gates for DNA analysis: detection of rifampin resistance in M. tuberculosis DNA](#)", *Angewandte Chemie-International Edition*, Vol. 51, Issue: 36, pp. 9075 – 9077, 2012.
73. Prokup A., Hemphill J., and Deiters, Alexander, "[DNA computation: a photochemically controlled AND gate](#)", *Journal of the American Chemical Society*, Vol.: 134, Issue: 8, pp. 3810 – 3815, 2012.
74. Hockenberry A. J., and Jewett M. C., "[Synthetic in vitro circuits](#)", *Current Opinion in Chemical Biology*, Vol.: 16, Issue: 3 – 4, pp. 253 – 259, 2012.
75. Binglin L., Jiang Yu, Chen X., and Ellington A. D., "[Probing spatial organization of DNA strands using enzyme-free hairpin assembly circuits](#)", *Journal of American Chemical Society*, Vol.: 134, Issue: 34, pp. 13918 – 13921, 2012.
76. S. F. J. Wickham, J. Bath, Y. Katsuda, M. Endo, K. Hidaka, H. Sugiyama, and A. J. Turberfield, "[A DNA-based molecular motor that can navigate a network of tracks](#)", *Nature Nanotechnology*, Vol.: 7, Issue: 3, pp. 169 – 173, 2012.
77. X. Chen, "[Expanding the rule set of DNA circuitry with associative toehold activation](#)", *Journal of the American Chemical Society*, Vol.: 134, Issue: 1, pp. 263 – 271, 2012.
78. J. Elbaz, F. Wang, F. Remacle, and I. Willner, "[pH-programmable DNA logic arrays powered by modular DNAzyme libraries](#)", *Nano Letters*, Vol.: 12, Issue: 12, pp. 6049 – 6054, 2012.
79. A. J. Genot, J. Bath, and A. J. Turberfield, "[Combinatorial displacement of DNA strands: application to matrix multiplication and weighted sums](#)", *Angewandte Chemie International Edition*, Vol.: 52, Issue: 4, pp. 1189 – 1192, 2013.
80. T. Miyamoto, S. Razavi, R. DeRose, and T. Inoue, "[Synthesizing biomolecular-based Boolean logic gates](#)", *ACS Synthetic Biology*, Vol.: 2, Issue: 2, pp. 72 – 82, 2013.
81. J. L. Doob, "[Stochastic processes](#)", *John Wiley & Sons, Inc.*, New York, 1953.
82. D. Abbot, "[Developments in Parrondo's paradox](#)", *Applications of Nonlinear Dynamics-Model and Design of Complex Systems*, pp. 307 – 321, 2009.

83. J. SantaLucia, Jr., H. T. Allawi, and A. Seneviratne, "[Improved nearest-neighbor parameters for predicting DNA duplex stability](#)", *Biochemistry*, Vol.: 35, Issue: 11, pp. 3555 – 3562, 1996.
84. N. Peyret, P. A. Seneviratne, H. T. Allawi, and J. SantaLucia, Jr., "[Nearest-neighbor thermodynamics and NMR of DNA sequence with internal A center dot A, C center dot C, G center dot G, and T center dot T mismatches](#)", *Biochemistry*, Vol. 38, Issue: 12, pp. 3468 – 3477, 1999.
85. N. E. Watkins, and J. SantaLucia, Jr., "[Nearest-neighbor thermodynamics of deoxyinosine pairs in DNA duplexes](#)", *Nucleic Acids Research*, Vol.: 33, Issue: 19, pp. 6258 – 6267, 2005.
86. F. Sanger, S. Nicklen, and A. R. Coulson, "[DNA sequencing with chain-terminating inhibitors](#)", *Proceedings of the National Academy of Science of the United States of America*, Vol.: 74, Issue: 12, pp. 5463 – 5467, 1977.
87. H. Ohmori, "[Introduction of a new method for sequencing DNA](#)", *Seikagaku*, Vol.: 49, Issue: 6, pp. 429 – 432, 1977.
88. J. Shendure, G. J. Porreca, N. B. Reppas, X. X. Lin, J. P. McCutcheon, A. M. Rosenbaum, M. D. Wang, K. Zhang, R. D. Mitra, and G. M. Church, "[Accurate multiplex polony sequencing of an evolved bacterial genome](#)", *Science*, Vol.: 309, Issue: 5741, pp. 1728 – 1732, 2005.
89. J. Shendure, and H. L. Ji, "[Next-generation DNA sequencing](#)", *Nature Biotechnology*, Vol.: 26, Issue: 10, pp. 1135 – 1145, 2008.
90. E. W. Dijkstra, "[A note on two problems in connection with graphs](#)", *Numerische Mathematik*, Vol.: 1, pp. 269 – 271, 1959.
91. R. C. Prim, "[Shortest connection networks and some generalizations](#)", *Bell System Technical Journal*, Vol.: 36, pp. 1389 – 1401, 1957.
92. J. L. Perez and M. A. Banuelos, "[Electronic models of fuzzy gates](#)", *Instrumentation and Development*, Vol.: 3, No. 5, 1995.
93. Taleb, N. Nicholas, "[The black swan: the impact of the highly improbable](#)", *Random House Press*, 2007.
94. A. F. George, and G. Tom, "[The stock market course](#)", *John Wiley & Sons, Inc.*, pp. 117 – 124, 2001.
95. Urrutia, J., "[The cost of deposit insurance: derivations of a risk-adjusted premium](#)", *Insurance Mathematics & Economics*, Vol.: 9, Issue: 4, pp. 281 – 290, 1990.

96. S. H. Cox and, R. G. Schwebach, "[Insurance futures and hedging insurance price risk](#)", *Journal of Risk and Insurance*, Vol.: 59, Issue: 4, pp. 628 – 644, 1992.
97. L. Allen and A. Saunders, "[Forbearance and valuation of deposit insurance as a callable put](#)", *Journal of Banking & Finance*, Vol.: 17, Issue: 4, pp. 629 – 643, 1993.
98. J. A. Conover and D. A. Dubofsky, "[Efficient selection of insured currency positions: protective puts vs. fiduciary calls](#)", *Journal of Financial and Quantitative Analysis*, Vol.: 30, Issue: 2, pp. 295 – 312, 1995.
99. J. D. Cummins and M. A. Weiss, "[Convergence of insurance and financial markets: hybrid and securitized risk-transfer solutions](#)", *Journal of Risk and Insurance*, Vol.: 76, Issue: 3, pp. 493 – 545, 2009.
100. J. Hull, "[Options, futures, and other derivatives](#)", *Pearson Education, Inc.*, 7th edition, pp.45 – 50, 2011.
101. Tjalling C. Koopmans and Martin Beckmann, "[Assignment problems and the location of economic activities](#)", *Econometrica*, Vol. 25, Issue: 1, pp. 53 – 76, 1957.
102. James Munkres, "[Algorithms for the assignment and transportation problems](#)", *Journal of the Society for Industrial and Applied Mathematics*, Vol. 5, Issue: 1, pp. 32 – 38, 1957.
103. D. Gale and L. S. Shapley, "[College admissions and the stability of marriage](#)", *The American Mathematical Monthly*, Vol. 69, No. 1, pp. 9 – 15, 1962.
104. Alvin E. Roth, "[Two-sided matching with incomplete information about others' preferences](#)", *Games and Economic Behavior*, Vol. 1, pp. 191 – 209, 1989.
105. Erwin Kreyszig, "[Advanced engineering mathematics](#)", 9th Edition, *John Wiley & Sons, Inc.*, pp. 982 – 985.
106. Jian-Jun Shu, "[On generalized Tian Ji's horse racing strategy](#)", *Interdisciplinary Science Reviews*, Vol. 37, Issue 2, pp. 187 – 193, 2012.
107. Marvin L. Minsky, "[Recursive unsolvability of Post's problem of 'Tag' and other topics in theory of Turing machines](#)", *Annals of Mathematics*, Vol.: 74, Issue: 3, pp. 437 – 455, 1961.
108. Corrado Böhm and Giuseppe Jacopini, "[Flow diagrams, Turing machines and languages with only two formation rules](#)", *Communications of the ACM*, Vol.: 9, Issue: 5, pp. 366 – 371, 1966.
109. John Gill, "[Computational complexity of probabilistic Turing machines](#)", *SIAM Journal on Computing*, Vol.: 6, Issue: 4, pp. 675 – 695, 1977.

110. Joel David Hamkins and Andy Lewis, "[Infinite time Turing machines](#)", *Journal of Symbolic Logic*, Vol.: 65, Issue: 2, pp. 567 – 604, 2000.
111. J. J. Shu, Q.-W. Wang and K.-Y. Yong, "[DNA-based computing of strategic assignment problems](#)", *Physical Review Letters*, Vol. 106, No. 18, 188702, 2011.
112. J. B. Lundberg, "[Alternative algorithms for the GPS static positioning solution](#)", *Applied Mathematics and Computation*, Vol.: 119, Issue: 1, pp. 21 – 34, 2001.
113. E. F. Moore, "[The shortest path through a maze](#)", *Proceedings of an International Symposium on the Theory of Switching*, Harvard University Press, pp. 285 – 292, 1959.
114. F. Tanaka, Kameda, A., Yamamoto, M. and Ohuchi, A., "[Design of nucleic acid sequences for DNA computing based on a thermodynamic approach](#)", *Nucleic Acids Research*, Vol.: 33, Issue: 3, pp. 903 – 911, 2005.
115. R.L. Keeney and H. Raiffa, "[Decisions with multiple objectives: Preferences and value tradeoffs](#)", *Cambridge University Press*, 1993.
116. D. Ellsberg, "[Risk, ambiguity, and the savage axioms](#)", *Quarterly Journal of Economics*, Vol.: 75, No.: 4, pp. 643–669, 1961.
117. M. K. Starr, "[Discussion of some normative criteria for decision-making under uncertainty](#)", *IMR-Industrial Management Review*, Vol.: 8, Issue: 1, pp. 71 – 78, 1966.
118. M. Cohen and J.-Y. Jaffray, "[Decision making in a case of mixed uncertainty: A normative model](#)", *Journal of Mathematical Psychology*, Vol.: 29, Issue: 4, pp. 428 – 442, 1985.
119. S. A. Maliphant and Smith "[Mini-risk – strategies for a simplified board game](#)", *Journal of the Operational Research Society*, Vol.: 41, Issue: 1, pp. 9 – 16, 1990.
120. B. Sharon, "[Risky business: an in-depth look at the game RISK](#)", *Undergraduate Math Journal*, Vol.: 3, Issue: 2, 2002.
121. J. A. Osborne, "[Markov chains for the RISK board game revisited](#)", *Mathematics Magazine*, Vol.: 76, Issue: 2, pp. 129 – 135, 2003.
122. Hindemith, "[Battle outcome probabilities for the RISK board game](#)", 2011, URL: <http://www.spreadsheetadvice.com/2011/04/battle-outcome-probabilities-for-the-risk-board-game/>.