Biologically Inspired Computation toward
Management Engineering Applications

January 2008

Waseda University
Graduate School of Information, Production and Systems
Major in Information, Production and Systems Engineering
Management Engineering Laboratory

Jyh-Fu Jeng
In memory of my father, Ying-San;
To my mother, Mei-Chih,
and beloved, Hui-Fen.
# Table of Contents

Table of Contents ii
List of Tables iv
List of Figures vi
Citations to Previously Published Work ix
Abstract x
Acknowledgements xi

1 Introduction 1

2 Bio-soft Computing with DNA 6
  2.1 Preliminary ............................................. 6
  2.2 DNA: An Overview .................................... 11
  2.3 Techniques for DNA Computing ...................... 15
  2.4 Adleman-Lipton Model ............................... 26
  2.5 Background ........................................... 27
  2.6 Challenges ............................................ 30

3 Bio-soft Computing with Fixed-length DNA to a Group Control Optimization Problem 37
  3.1 Overview ............................................... 37
  3.2 Elevator Dispatching Problem ...................... 39
  3.3 Background ............................................ 41
  3.4 Molecular Algorithm for Solving Elevator Dispatching Problem 44
  3.5 Empirical Study ..................................... 46
  3.6 Concluding Remarks ................................ 50
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Bio-inspired Evolutionary Method for Cable Trench Problem</td>
<td>53</td>
</tr>
<tr>
<td>4.1</td>
<td>Overview</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>Cable Trench Problem</td>
<td>54</td>
</tr>
<tr>
<td>4.3</td>
<td>Background</td>
<td>57</td>
</tr>
<tr>
<td>4.4</td>
<td>Molecular Algorithm for Solving CTP</td>
<td>59</td>
</tr>
<tr>
<td>4.5</td>
<td>Empirical Study</td>
<td>61</td>
</tr>
<tr>
<td>4.6</td>
<td>Concluding Remarks</td>
<td>65</td>
</tr>
<tr>
<td>5</td>
<td>Bio-inspired Fuzzy Forecasting: A New Forecasting Technique</td>
<td>68</td>
</tr>
<tr>
<td>5.1</td>
<td>Overview</td>
<td>68</td>
</tr>
<tr>
<td>5.2</td>
<td>Fuzzy Time Series</td>
<td>70</td>
</tr>
<tr>
<td>5.3</td>
<td>Background</td>
<td>71</td>
</tr>
<tr>
<td>5.4</td>
<td>Modeling Time Series and Forecasting with DNA</td>
<td>72</td>
</tr>
<tr>
<td>5.5</td>
<td>Mean Absolute Forecasting Accuracy: The Performance Evaluation</td>
<td>78</td>
</tr>
<tr>
<td>5.6</td>
<td>Empirical Application</td>
<td>79</td>
</tr>
<tr>
<td>5.7</td>
<td>Concluding Remarks</td>
<td>82</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>93</td>
</tr>
<tr>
<td>A</td>
<td>MST with Minimum Path Length is NP-complete: A Proof</td>
<td>106</td>
</tr>
<tr>
<td>List of Related Publications</td>
<td></td>
<td>109</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Comparison between DNA computer and conventional computer . . . 31

3.1 DNA sequences associated with each floor node and cost for the six-
floor elevator routing problem. . . . . . . . . . . . . . . . . . . . . 48

3.2 Result from DTG-PCR. Two different DNA strands represent the pos-
sible routes of the same length which cannot be separated by conven-
tional gel electrophoresis. . . . . . . . . . . . . . . . . . . . . . . . . 51

4.1 DNA sequences associated with each vertex and weight for the 6-vertex,
8-edge, 4-weight cable trench problem. Each vertex is randomly associated
with a 20-mer sequence of DNA. Weight sequences are designed to have different
melting temperatures depending on its weight. The lighter in weight, the lower the
melting temperature is. The melting temperature is correlated to the GC content. . 63

4.2 Result from DTG-PCR. Three different DNA strands represent the possible
routes of the same length which cannot be separated by the conventional gel elec-
trophoresis. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66

4.3 Possible routes resulting from DTG-PCR and their distinct behavior
in melting temperature and GC content . . . . . . . . . . . . . . . . 67

5.1 Memberships of fuzzy time series $k_t = \frac{\Delta X_t}{T}$ w.r.t linguistic variables . 74

5.2 The ratio of primer for each test tube . . . . . . . . . . . . . . . . . 75

5.3 The ratios of primer and memberships for JPY/USD . . . . . . . . . 81

5.4 The ratios of primer and memberships for EUR/USD . . . . . . . . . 81
5.5 The ratios of two DNAs and memberships for JPY/USD after reaction 82
5.6 The ratios of two DNAs and memberships for EUR/USD after reaction 83
5.7 Memberships of DNA concentration $C = \frac{50 - DNA_{1\%}}{50}$ w.r.t linguistic variables 83
5.8 Forecasting performance comparison of JPY/USD 84
5.9 Forecasting performance comparison of EUR/USD 85
5.10 Forecasting accuracy comparison between Best ARIMA and DNA computing 86
List of Figures

2.1 Pascalina displayed at Musée des Arts et Métiers, Paris Inv 823-1. Image retrieved from Wikipedia. Copyright ©2005 David Monniaux. 7

2.2 Turing Machine in a finite state representation. Each circle represents a “state” of the TABLE. “Direction” of a state transition is shown by an arrow. The label (e.g., 0/P,R) of the outgoing state specifies the scanned symbol that causes a particular transition (e.g., 0) followed by a slash /, followed by the subsequent “behaviors” of the machine, e.g., “P Print” then move tape “R Right”. No general accepted format exists. 8

2.3 ENIAC, the first general purpose electronic computer. “U.S. Army Photo”. 10

2.4 EDVAC, the first stored program electronic computer. “U.S. Army Photo”. 10

2.5 An overview of the structure of DNA. Image retrieved from Wikipedia, created by Michael Ströck, 2006. Released under the GFDL. 13

2.6 The chemical backbone of DNA. 14

2.7 The insight of DNA that made it all come together: complementary pairing of the bases. 14

2.8 DNA denaturing and annealing. Example from genome of SARS (range= chr1: 1 – 20). 16

2.9 DNA ligation promoted by lagase enzyme. Example in EcoRI. 17

2.10 Primer extension of DNA or RNA. 19
2.11 Schematic representation of PCR. $N_0$ copies of duplex template DNA are subjected to $n$ cycles of PCR. During each cycle, duplex DNA is denatured by heating, which then allowed primers to anneal to the targeted sequence. In the presence of DNA polymerase and dNTPs, primer extension takes place. The desired blunt-ended duplex product (red solid bars with arrows) appears during the third cycle and accumulates exponentially during subsequent cycles. Following $n$ cycles of exponential PCR, there will be $N_0(1+Y)^{n-1}$ copies of the duplex target sequence.

2.12 The Thermal cycler. (A laboratory apparatus used for PCR, also known as a thermocycler, PCR machine or DNA amplifier.)

2.13 Gel electrophoresis process.

2.14 A set of gel electrophoresis apparatus.

2.15 A gel image visualized under UV light. The very left column is the DNA ladder (marker) for sizing.

2.16 An example of nucleic acid has subsequences that complement each other fold up into hairpins structure.

3.1 Graph of all possible movements for a single elevator

3.2 Example of a six-story building equipped with two identical elevator cars, A and B

3.3 Example of the encoding scheme. An oligonucleotide $V_{1\rightarrow 2}$ is created that is the 3’ 10-mer of $V_1$ followed by the weight sequence of path length, and then the 5’ 10-mer of $V_2$.

3.4 The distinct behavior in melting temperature among the possible optimal solutions. $\bigcirc$: car call; $\triangle$: up hall call; $\triangledown$: down hall call.

4.1 The KSRP facilities layout, available routes and their weights.

4.2 A 6-vertex, 8-edge, 4-weight CTP translated from KSRP layout plan.

4.3 Example of the encoding scheme. An oligonucleotide $V_{1\rightarrow 2}$ is created that is the 3’ 10-mer of $V_1$ followed by the weight sequence of path length, and then the 5’ 10-mer of $V_2$. 
5.1 DNA encoding scheme. DNA1 and DNA2 are designed in represent “up” trend and “down” trend respectively. DNA1 and DNA2 have it own special design primers. ............................................... 74
5.2 Each test tube is set up to contain the template of DNA1 and DNA2 accompanied with their own specific PCR primers, and the up trend primers (i.e., UF, UR) and down trend primers (i.e., DF, DR) are inputted in ratio according to the △Xt. ............................................... 75
5.3 Trend of Exchange Rate: JPY/USD and EUR/USD ................. 80
A.1 The original graph for proving MST with minimum path length is NP-complete. ................................................................. 107
A.2 Each vertex of the planar graph is replaced with a gadget. ........ 108
A.3 Replaced with a special “crossover gadget”. ....................... 108
Citations to Previously Published Work

Large portions of Chapters 3, 4, and 5 have appeared in the following three papers, respectively:


Partial of Chapter 2 and some portion of Chapter 1 have appeared in:

Abstract

Ever since scientists discovered that conventional silicon-based computers have physical upper limits, they have been searching for alternative media with which to solve computational problems. That search has led them, among other places, to deoxyribonucleic acid (DNA). Since Adleman’s pioneering investment, various researches have been made in DNA computing theoretically and experimentally. However, none of any application-based research was found in literature. The aim of this work is to investigate the possibility of bio-soft computing implementing the engineering applications.

Three advances were made in my study that can be divided into two parts. In the first part, thermodynamic controlled fixed-length DNA computing algorithms were proposed for two applications, group control optimization and cable trench problem, in two strategic considerations. The fixed-length DNA molecules were used to represent the numerical values where the weights are thermodynamically controlled by DNA melting temperature. Two empirical applications were demonstrated individually and DNA computing approach to engineering applications were shown to be achievable.

A novel forecasting technique biologically-inspired is developed in the second part. The historical data were fuzzified and DNA templates that best fit the historical trend were used to forecast the future. Currency exchange rates forecasting were demonstrated in empirical application and experiment results shown our proposed algorithm can outperform the best ARIMA model.
Acknowledgements

Completing this doctoral work has been a wonderful and overwhelming experience. It is hard to know whether it has been grappling with the combination of computer science, mathematics, biology and chemistry which have been the real learning experience, or grappling with how to write a paper, give a coherent talk and tutorial, work in a group, recover a crashed hard drive, stay up until the sunrise, and many more.

I have been very privileged to have undoubtedly the most intuitive, smart and supportive advisor anyone could ask for, namely Junzo Watada. Ever since I learned from him, I have been stimulated and excited by his constant flow of good ideas. Professor Watada is friendly, open-minded, patient, humor and has an ability to cut through difficult problems with a simple explanation that I will always admire. He is always available when I am asking for, where I can even find him in his office at midnight. He has also known when (and how) to give me a little push in the forward direction when I needed it. It has truly been a pleasure (without pressure) to be a part of his research family and I look forward to continue what I hope is just the beginning. Professor Watada is, and will be my lifelong mentor.

I would like to thank my committee members, Professor Kotaro Hirasawa, Professor Toshitsugu Ueda, and Professor Harutoshi Ogai, for volunteering their precious time to review this work, as well as their many suggestions and constant support.

I would like to acknowledge the financial supports received throughout the years of my study in Japan. They include Mitsubishi UFJ Trust Scholarship, Honor Scholarship, Okawa Information and Communication Academy Scholarship, Japan Ministry of Education, Culture, Sports, Science and Technology Tuition Deduction Scholarship, and Waseda University Doctorial Scholarship. My grateful appreciation to those organizations.

This work has benefited tremendously from many external interactions and would
not have been possible otherwise. I wish to express my sincere appreciation to the following professors for their invaluable guidance favors. From the world of mathematics, Professor Berlin Wu of National Chengchi University, Taiwan, offers discussion and creative ideas constantly. From the world of biology and biochemistry, Professor Jui-Yu Wu of Taipei Medical University, Taiwan, guides me to the fundamental knowledge theoretically and experimentally that necessitate to my research. From the world of computer science, Professor Kwong-Sak Leung of Chinese University of Hong Kong, China, provides fascinating conversations and cares during my visiting. Thank you all for having been wonderful influences.

I am grateful to the following people for fruitful technical discussions and personal friendship. They include Professor Jui-Chao Chen of Fu Jen Catholic University, Taiwan; Dr. Keiichi Horio of Kyushu Institute of Technology, Japan; Professor Kun-Huang Huarng of Feng Chia University, Taiwan; Dr. Zuwairie B. Ibrahim of Universiti Teknologi Malaysia, Malaysia; Professor Marzuki B. Khalid of Center for Artificial Intelligence and Robotics (CAIRO), Malaysia; Professor Feng-Tse Lin of Chinese Culture University, Taiwan; Professor Masahiroyo Nakano of University of Occupational and Environmental Health, Japan; Professor Hung T. Nguyen of New Mexico State University, U.S.A; Professor Jeng-Shyang Pan of National Kaohsiung University of Applied Sciences, Taiwan; Professor Witold Pedrycz of University of Alberta, Canada; Professor Yan Shi of Kyushu Tokai University, Japan; Professor Shao Chin Sung of Aoyama Gakuin University, Japan; Professor Fumiaki Takeda of Kochi University of Technology, Japan; Professor Lanh Tran of Indiana University, U.S.A; Professor Tzong-Ru Tsai of Tamkang University, Taiwan; Professor Takeshi Yamakawa of Kyushu Institute of Technology, Japan; Professor Chang-Biau Yang of National Sun Yat-Sen University, Taiwan, and many more. (In alphabetical order by last name.)

Living in Japan while completing my doctorial study is definitely an invaluable experience in my life, especially in the parts of culture exchange and language learning. IPS has been a wonderful workplace because of its friendly and diverse environment. I am indebted to Cathy Yi-Chun Yeh for her great assistance in my first time living in Japan and Dr. Joanna Pawlat for her information sharing. My heartiest thanks also go to the former and present members of the Watada Lab for their friendship and assistance over the years. They include Rohani Abu Bakar, Eugene Hsiao, Shinya Imai, Ikno Kim, Zalili Musa, Kunio Shibata, Michelle Tsai, and Bing Xu. We have
certainly been through a lot together such as research discussion, conferences attending, hot spring trip, lab network crash, to name a few. I know I have met many of my most trusted friends in the Watada Lab, and look forward to recount the good times as we meet again.

To my uncle, Jao-Hong Jeng, aunt, Ming-Hui Wu, and my two cousins, Ing-Chu and Ing-Yi, thank you for making my life in Japan more enjoyable and fulfilling.

Last but not least, I reserve my deepest thanks for my family whom I love very much. This dissertation is dedicated in memory of my father, Albert Ying-San Jeng. His smart, integrity, brave and moderate will always be with me. My grateful thanks to my mother, May Mei-Chih Chen, for her unconditional love and fully support. I owe them a great debt of gratitude which I could never be able to repay for they have made so many sacrifices, placing their children’s well-being and future on a pedestal high above their own. Thanks to my younger brother, Paul Chih-Chun Cheng, who is always there for me. Most of all, to my wife, Jasmine Hui-Fen Liang, who has been my guiding light and love throughout my study. She has seen my best and my worst, and provided support whenever I need. When it comes down to it, they are ultimately responsible for making this dissertation a reality.

Thank you all from the bottom of my heart.

Don J.-F. Jeng
January 2008
Kitakyushu, Japan
Chapter 1

Introduction

“I never failed once. It just happened to be a 2000-step process.”

- Thomas Edison, American inventor,
  responding to a reporter who asked how it felt
to fail 2000 times before successfully inventing the light bulb.

Management and industrial engineering refer to ways of taking the disciplines, practices and processes of engineering, and transferring them to operational management processes. It feedbacks on decisions and used to improve operational strategy. Information technology applications are developed in support of operational processes, such as operational systems, enterprise resource management systems, analytical processing systems, process analysis tools, and strategic management applications, etc. Many of these tools are involved with state-of-the-art methodologies, for example, optimization, scheduling, and logistics planning. The field of soft computing, which includes neural networks, fuzzy logic, and genetic algorithms, are often applied to search for an optimal and/or intelligent solution.

Meanwhile, computer chip manufacturers are furiously racing to make the next
microprocessor for toppling the speed records. Microprocessors made of silicon, how-
however, will eventually reach its physical limitation and miniaturization. Scientists have
found a material which has potential to be the media of next generation micropro-
cessor, the DNA, and the technique is named DNA computing. Biologically inspired
computing with DNA began in earnest with Adleman [2], who solved a directed
Hamiltonian path problem in 1994. With DNA molecules’ enormous parallelism, it
provides an opportunity to overcome the limitation of a conventional computer.

Many researches in DNA computing have been done since Adleman’s pioneer-
ing investigation. However, none of any application-oriented research is observed.
This dissertation focuses on biologically inspired computing methodologies toward
the management engineering applications in particular. It falls naturally into three
parts:

• Review of DNA computing (Chapter 2)

• Study of biologically inspired computation toward engineering applications (Chap-
ters 3, 4, and 5)

• Conclusion and future directions (Chapter 6)

DNA computing is briefly reviewed in the first part. Chapter 2 starts from how
humans evolve ways of computation and moving their direction to biologically inspired
one. Then, the overview of DNA and some fundamental biological techniques used
to manipulate DNA is introduced. The very first and still commonly applied DNA
computing model is presented followed by recent significant advances in the research
of DNA computing. The challenges for DNA computing are also addressed.
The second part forms the main body of the thesis, and they are all devoted to the study of biologically inspired computing with DNA molecules toward the management engineering problems. These probably contain the most significant new results; this is reflected in the choice of dissertation title. These studies include, which are relatively independent:

- Bio-soft computing with fixed-length DNA to a group control optimization problem (Chapter 3)
- Bio-inspired evolutionary method for a cable trench problem (Chapter 4)
- Bio-inspired fuzzy forecasting: a new forecasting technique (Chapter 5)

In Chapter 3, a bio-soft computing method with fixed-length DNA for a group control optimization problem is presented. In the example of a multi-elevator dispatching problem, fixed-length DNA strands were used to represent nodes and costs, where the costs were varied by the melting temperature of DNA strands. The optimal solution to a 6-story 2-elevator dispatching problem was searched by biochemical techniques based on thermodynamic properties of designed DNA strands. This research has shown the potential of bio-soft computing to solve engineering applications, and could be implemented in future bio-systems.

A bio-inspired evolutionary method with DNA is presented for solving a cable trench problem in Chapter 4. The cable trench problem is a combination of the shortest path problem and minimum spanning tree problem, which makes it difficult to be solved by a conventional computing method. DNA computing is applied to overcome the limitation of a silicon-based computer. Numerical values are represented by fixed-length DNA strands, and weights are designed that vary from DNA melting
temperatures. Biochemical techniques with DNA thermodynamic properties are used to effective local search for the optimal solution.

A new forecasting technique biologically inspired is developed in Chapter 5. There are many forecasting techniques including exponential smoothing, ARIMA model, GARCH model, Neural Networks and genetic algorithm, etc. These methods, however, have their drawbacks and advantages. Since financial time series may be influenced by many factors, such as trading volume, business cycle, oil price, seasonal factor, to name a few, conventional model based techniques and traditional hard computing methods seem inadequate in the prediction. In recent years, the innovation and improvement of forecasting techniques have caught more attention, and also provides indispensable information in decision-making process, especially in the fields of financial economics, management planning and control. In this chapter, a new forecasting technique biologically inspired from natural selection is developed. The new forecasting technique may be of use to a nonlinear time series forecasting. The method combines mathematical, computational, and biological sciences, and the consideration includes fuzzy logic, DNA encoding, polymerase chain reaction, and DNA quantification. In an empirical application, a novel approach to forecast the exchange rates through bio-inspired forecasting is demonstrated. Mean absolute forecasting accuracy method is defined for evaluating the performance of linguistic forecasting, and the result compared with ARIMA model is also illustrated.

The conclusion is recapped in the last part where future research directions are also addressed. This dissertation makes major contributions to both the computer science and management engineering research communities. Solving a management
engineering problem is shown to be achievable through biologically inspired computing with DNA molecules. This milestone will encourage the communities to further develop DNA algorithms and methodologies for engineering problems that are applied in future bio-systems. Also, certain general limitations in DNA computation are raised in this dissertation for future studies.
Chapter 2
Bio-soft Computing with DNA

“Progress might have been all right once, but it has gone on too long.”
- Ogden Nash, American poet.

2.1 Preliminary

People often relate computing with computers, that we seem to neglect the electronic computers are relatively new in our computing history. In the history of humanity, people in the stone age needed to count and compute, either for time measuring, commerce or constructions. The means used for performing calculations were whatever was available, and thus gradually progressed from manual to mechanical, and from there on to electrical devices.

Human started off counting by fingers, stones, and drawing. The very early mechanical computing devices include the abacus and slide rule which dates from about 150 – 100 BC. At the end of Middle Ages in 1623, Wilhelm Schickard, a German polymath who built the first mechanical computing machine named Speeding Clock.
Later in 1625, Blaise Pascal invented the second mechanical calculator, a decimal machine, based on a gear system called Pascalina as shown in Fig. 2.1.

However, none of the above mentioned devices are programmable. In 1801, Joseph Marie Jacquard, a French silk weaver and inventor, improved on the original punched card design of Jacques de Vaucanson’s loom of 1745. He used a series of punched paper cards as a template to allow the loom to weave intricate patterns automatically. The resulting Jacquard loom was an important step in the development of computers because the use of punched cards to define woven patterns can be viewed as an early, albeit limited, form of programmability.

In 1837, Charles Babbage was the first to conceptualize and design a fully programmable mechanical computer called Analytical Engine, though it never been built due to the budget. Later, Herman Hollerith, an American statistician who developed a mechanical tabulator based on punched cards to rapidly tabulate statistics
Figure 2.2: Turing Machine in a finite state representation. Each circle represents a “state” of the TABLE. “Direction” of a state transition is shown by an arrow. The label (e.g., 0/P,R) of the outgoing state specifies the scanned symbol that causes a particular transition (e.g., 0) followed by a slash /, followed by the subsequent “behaviors” of the machine, e.g., “P Print” then move tape “R Right”. No general accepted format exists.

from millions of pieces of data. The machine is built for 1890 U.S. Census, and Hollerith’s firm, the Tabulating Machine Company, later became IBM. The significance of Hollerith’s invention is that he marked the transition from a decimal system to a binary system, 0 and 1, corresponding to the card punched in or not. The binary representation is then led to the next stage of computing evolution, an electronic device.

The theoretical foundation of modern electronic computers was laid by Alan Turing in 1936 [102]. Turing machines are extremely basic abstract symbol-manipulating devices which, despite their simplicity, can be adapted to simulate the logic of any computer that could possibly be constructed. As illustrated in Fig. 2.2, the concept of the Turing machine is based on the idea of executing a well-defined procedure by changing the contents (i.e., read, write and erase symbols) of an unlimited paper tape, which is divided into squares that can contain one of a finite set of symbols. The procedure is similar to modern day’s RAM.
The first general purpose electronic computer, ENIAC (Electronic Numerical Integrator And Computer), shown in Fig. 2.3, was built for the U.S. Army’s Ballistics Research Laboratory in 1946. ENIAC contained 17,468 vacuum tubes, 7,200 crystal diodes, 1,500 relays, 70,000 resistors, 10,000 capacitors and around 5 million hand-soldered joints. It is capable of being reprogrammed to solve a full range of computing problems. However, the programming of ENIAC is problem-dependent by manipulating its switches and cables. John von Neumann addressed the issue and introduced the idea of a computer should contain a conditional go-to instruction and the storing of the data and the program together in the same memory unit, which meant that the machine itself could alter either its data or program. The thought led to the first generation of modern programmable electronic computer, EDVAC (Electronic Discrete Variable Automatic Computer) as shown in Fig. 2.4. Unlike its predecessor the ENIAC, it was binary rather than decimal, and was a stored program machine.

In 1947, the transistor was discovered by Bell Lab, and later in 1958, the first integrated circuit (IC) was invented by Jack Kilby of Texas Instruments. These are the two advances that truly released the potential of electronic computers. In 1971, Intel and Texas Instruments almost simultaneously invented the microprocessor. By the 1980s, computers had become sufficiently small and inexpensive. In conjunction with the widespread growth of the Internet since the 1990s, we witness the computer revolution, the fastest growing technology in human’s history.

This review is intended to point out that electronic computers are only the latest in the long period of human’s history. We attempt to compute by investigating all the possibilities. Five decades ago, the revolution of computing has dramatically changed from mechanical to electronic one. Nowadays, almost all modern electronic devices
Figure 2.3: ENIAC, the first general purpose electronic computer. “U.S. Army Photo”.

Figure 2.4: EDVAC, the first stored program electronic computer. “U.S. Army Photo”.

contain a computer of some kind. Indeed, human did not start computing with electronic computers, and there is no reason why it should end with them. In reality, the electronic computers are reaching their limitations in physical laws. Scientists are searching for alternative tools for computation to break down the barriers, instead of electrical, that revolution led to the biological ones [51].

In fact, as early as 1959, Richard P. Feynman already showed the biological system as an example in contrast with miniaturizing the computer [33]. Though Feynman intended to discuss the problem of manipulating and controlling things on a small scale, he had highlighted the potential of biological devices being used as tools for computation. In 1974, Aviram and Ratner [13] illustrated a theoretical molecular rectifier. In 1981, Carter [19] proposed the idea of a molecular electronic device. Their concept was to synthesize molecules that show electronic functions.

Later in 1994, a trial was carried out by L. M. Adleman [2] who solved a mathematical problem using DNA molecules. In Adleman’s pioneering work, the DNA was used for calculation, and it opens the door of DNA computing.

2.2 DNA: An Overview

DNA is a nucleic acid molecule that contains the genetic instructions used in the development and functioning of all cellular forms of life and many viruses. The main role of DNA in the cell is the long-term storage of information. DNA contains the instructions needed to construct other components of cells, such as ribonucleic acid (RNA) molecules and proteins.

In 1952, Rosalind Franklin took an X-ray diffraction image of DNA. Based on this image and the information that bases were paired, James D. Watson and Francis
Crick suggested the first accurate model of DNA structure in 1953[110].

Figure 2.5 illustrates an overview structure on partial DNA double helix. The backbone of the DNA strand is made from alternating phosphate and sugar residues [38] as shown in Fig. 2.6. The sugar in DNA is 2-deoxyribose, which is a pentose (five carbon) sugar. The sugars are joined together by phosphate groups that form phosphodiester bonds between the third and fifth carbon atoms of adjacent sugar rings. These asymmetric bonds mean a strand of DNA has a direction. In a double helix the direction of the nucleotides in one strand is opposite to their direction in the other strand. This arrangement of DNA strands is called antiparallel. The asymmetric ends of a strand of DNA bases are referred to as the 5′ (five prime) and 3′ (three prime) ends [16].

The DNA double helix is stabilized by hydrogen bonds between the bases attached to the two strands. The four bases in DNA are adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T). These four bases are attached to the sugar/phosphate to form the complete nucleotide as shown in Fig. 2.7.

These bases are classified into two types; as illustrated in Fig. 2.7, A and G are fused five- and six-membered heterocyclic compounds called purines, while C and T are six-membered rings called pyrimidines. Each type of base on one strand forms a bond with just one type of base on the other strand. This is called complementary base pairing. Here, purines form hydrogen bonds to pyrimidines, with A bonding only to T, and C bonding only to G. This arrangement of two nucleotides binding together across the double helix is called a base pair [83].

The two types of base pairs form different numbers of hydrogen bonds, AT forming two hydrogen bonds, and GC forming three hydrogen bonds as shown in Fig. 2.7. The
Figure 2.5: An overview of the structure of DNA. Image retrieved from Wikipedia, created by Michael Ströck, 2006. Released under the GFDL.
Figure 2.6: The chemical backbone of DNA.

Figure 2.7: The insight of DNA that made it all come together: complementary pairing of the bases.
GC base pair is therefore stronger than the AT base pair. As a result, it is both the percentage of GC base pairs and the overall length of a DNA double helix that determine the strength of the association between the two strands of DNA. Long DNA helices with a high GC content have stronger-interacting strands, while short helices with high AT content have weaker-interacting strands [21]. In the laboratory, the strength of this interaction can be measured by finding the temperature required to break the hydrogen bonds, their melting temperature (also called Tm value) [47].

2.3 Techniques for DNA Computing

Mathematically, a single strand of DNA consists of a combination of four different symbols, A, G, C, and T, which is an information encoding scheme of four tuple $\sum = \{A, G, C, T\}$. The combinations of the following primitive biological operations can be performed on computation with DNA. The bio-techniques that applied in the study of this dissertation include:

DNA Synthesizing

It is a technique of synthesizing a desired DNA having a predetermined sequence. Biochemically, when DNA is synthesized, the free 3’ hydroxyl (OH) group from the growing strand of DNA attacks the phosphate on the next base to be added. Pyrophosphate is released and the new base forms a phosphodiester bond with the growing strand of DNA. The free 3’ OH group is then free to attack the next base to be added [78]. This reaction is catalyzed by an enzyme named DNA polymerases. To synthesize a strand of DNA is now a routine job which can be performed by an instrument named DNA Synthesizer.
Denaturing

When a DNA solution is heated enough, the non-covalent forces that hold the two strands together weaken and finally break. When this happens, the double-stranded DNA breaks apart into its single-stranded complementary components as shown in Fig. 2.8. This process is known as DNA melting or denaturation \textit{in vitro}. The temperature at which the DNA strands are half denatured is called the \textit{melting temperature}, or $T_m$.

Heating, however, is not the only way to denature the DNA. Organic solvents such as dimethyl sulfoxide and formamide, or high pH, disrupt the hydrogen bonding between DNA strands can also promote denaturation [73].

Annealing

Annealing is the opposite of denaturing as shown in Fig. 2.8. Once the two strands
of DNA separate, they can, under the proper conditions, come back together again. This is called annealing or hybridization in vitro.

The best temperature for DNA hybridization is about 25°C below its $T_m$. This temperature is low enough that it does not promote denaturation, but high enough to allow rapid diffusion of DNA molecules and to weaken the transient bonding between mismatched sequences and short intra-strand base-paired regions [73].

**Ligating**

Ligating is a process to join together two DNA molecule ends, either from the same or different molecules. Specifically, it involves creating a phosphodiester bond between the 3’ OH of one nucleotide and the 5’ phosphate of another as shown in Fig. 2.9. The reaction is catalyzed by DNA ligase enzyme. This enzyme will ligate DNA fragments having blunt or overhanging/sticky ends.
Affinity Purification

Affinity purification is a technique to extract the DNA strands that contain a given pattern as a substring. With affinity purification, single-stranded DNA (ssDNA) containing a given subsequence $v$ can be filtered out from a heterogeneous pool of other strands. Two methods can achieve the affinity purification:

- Biotin Separation

  1. Create many copies of complement $v$;
  2. Attach the last step products to a surface in fixed matrix form;
  3. Pour the contents of heterogeneous DNA pool over the surface, the strands containing $v$ will then anneal to the anchored complementary strands;
  4. Wash the surface to remove the strands that did not anneal, which leaves only the strands that contain $v$.

- Magnetic bead separation

  1. Create many copies of complement $v$;
  2. Attach the last step products to tiny magnetic beads;
  3. Pour the contents of heterogeneous DNA pool over the container of magnetic beads, the strands containing $v$ will then anneal to the anchored complementary strands;
  4. Use a magnet to pull the beads out of the pool, which with the target strands $v$ attached.

Primer Extension
Figure 2.10: Primer extension of DNA or RNA.

Primer extension is used to map the 5′ ends of DNA or RNA fragments. It is done by annealing a specific oligonucleotide primer to a position downstream of that 5′ end as shown in Fig. 2.10. The primer is labeled, usually at its 5′ end, with $^{32}$P. This is extended with reverse transcriptase, which can copy either an RNA or a DNA template, making a fragment that ends at the 5′ end of the template molecule. DNA polymerase can also be used with DNA templates [64].

**Polymerase Chain Reaction (PCR)**

PCR is a method for amplifying DNA *in vitro*, it consists of the following key processes:

1. Initialization: a mix solution of template, primer, dNTP and enzyme is heated to 94 – 98°C for 1 – 9 minutes to ensure that most of the DNA template and primers are denatured;

2. Denaturation: heat the solution to 94 – 98°C for 20 – 30 seconds for separation
of DNA duplexes;

3. Annealing: lower the temperature enough (usually between 50–64°C) for 20–40 seconds for primers to anneal specifically to the ssDNA template;

4. Elongation/Extension: raise temperature to optimal elongation temperature of Taq or similar DNA polymerase (70 – 74°C) for the polymerase adds dNTP’s from the direction of 5' to 3' that are complementary to the template;

5. Final Elongation/Extension: after the last cycle, a 5 – 15 minutes elongation may be performed to ensure that any remaining ssDNA is fully extended.

Step 2 to 4 is repeated for 20 – 35 times; less cycles results less product, too many cycles increases fraction of incomplete and erroneous products. The overview of PCR is illustrated in Fig. 2.11. PCR is a routine job in the laboratory that can be performed by an apparatus named thermal cycler as shown in Fig. 2.12.

**Denaturation Temperature Gradient PCR**

Denaturation temperature gradient PCR (DTG-PCR) is a modified PCR method that the denaturation temperature changes with cycle [61]. In DTG-PCR, conventional PCR is performed where the temperature of the denaturation step (the step 2 of PCR procedure above mentioned) is gradually increased in cycles.

**Quantitative PCR**

Quantitative PCR (Q-PCR) is a modification of the PCR used to rapidly measure the quantity of DNA, complementary DNA (cDNA) or RNA present in a sample [112]. It may be used to determine a DNA sequence is presented in a sample, and
Figure 2.11: Schematic representation of PCR. $N_0$ copies of duplex template DNA are subjected to $n$ cycles of PCR. During each cycle, duplex DNA is denatured by heating, which then allowed primers to anneal to the targeted sequence. In the presence of DNA polymerase and dNTPs, primer extension takes place. The desired blunt-ended duplex product (red solid bars with arrows) appears during the third cycle and accumulates exponentially during subsequent cycles. Following $n$ cycles of exponential PCR, there will be $N_0(1 + Y)^{n-1}$ copies of the duplex target sequence.
the number of its copies produced in PCR.

**Gel Electrophoresis**

This process uses electricity to separate DNA fragments (as well as RNA and protein) by size as they migrate through a gel matrix. As illustrated in Fig. 2.13, the negatively charged nucleic acid molecules can be moved through an agarose matrix with an electric field (electrophoresis). Shorter molecules move faster and migrate further than longer ones [92]. A set of gel electrophoresis apparatus is shown in Fig. 2.14. After gel electrophoresis, the result as shown in Fig. 2.15 can be visualized with UV light and gel imaging device.

**Temperature Gradient Gel Electrophoresis**
Agarose gel
Negatively charged
DNA moving through
gel matrix toward the
positive pole
Power on
Gel visualized
with UV light

The gel is submersed in a tank filled with a salt solution that conducts electricity.

Using pipette to load DNA and blue dye into slots made in the agarose gel.

Negatively charged DNA moving through gel matrix toward the positive pole

The phosphate group in the DNA backbone carry negatively-charged oxygens, which given a DNA molecules an overall negative charge.

Figure 2.13: Gel electrophoresis process.
Figure 2.14: A set of gel electrophoresis apparatus.

Figure 2.15: A gel image visualized under UV light. The very left column is the DNA ladder (marker) for sizing.
Temperature Gradient Gel Electrophoresis (TGGE) is a form of electrophoresis that studies the behavior of substances under different temperatures. TGGE is able to reveal the presence of sequence heterogeneity in a given heteroduplex sample by inducing a thermal denaturing gradient that results in differences in the average electrophoretic mobilities of DNA that is identical in size, but different in sequence. Given that the melting profile of a DNA molecule is dependent on its sequence, DNA fragments of like size but different sequence denature at different temperatures. When performing electrophoresis under partial melting conditions, individual species display different states of equilibrium between the unmelted DNA fragment and the partially melted form. Because the fraction of time spent by the DNA molecules in the slower, partially melted form varies among specific sequences, less stable fragments maintain a lower average electrophoretic mobility during the separation than more stable fragments. In this way, the mobility difference induced by the thermal gradient allows TGGE to resolve DNA fragments of like size, but of different sequence, thus revealing the presence of mutations in a given sample [56].

DNA Sequencing

DNA sequencing is a method for determining the order of the nucleotide bases (i.e. A, G, C, T) in a DNA oligonucleotide. DNA sequencing of a relatively short strand (300 – 1000 nucleotides long) can now be sequenced automatically by an instrument named DNA sequencer.
2.4 Adleman-Lipton Model

The DNA computing proposed by Aldeman [2] and Lipton [63] are described in the following. These works will be basic fundamental operations in this dissertation. The Adleman-Lipton model: A test tube is a set of molecules of DNA (i.e., a multi-set of finite strings of $\sum\{A, C, G, T\}$). Given a tube, one can perform the following operations:

1. $Merge(T_1, T_2)$: for two given test tubes $T_1, T_2$ it stores the union $T_1 \cup T_2$ in $T_1$ and leaves $T_2$ empty;

2. $Copy(T_1, T_2)$: for a given test tube $T_1$ it produces a test tube $T_2$ with the same contents as $T_1$;

3. $Detect(T)$: Given a test tube $T$ it outputs “yes” if $T$ contains at least one strand, otherwise, outputs “no”;

4. $Separation(T_1, X, T_2)$: for a given test tube $T_1$ and a given set of strings $X$ it removes all single strands containing a string in $X$ from $T_1$, and produces a test tube $T_2$ with the removed strands;

5. $Selection(T_1, L, T_2)$: for a given test tube $T_1$ and a given integer $L$ it removes all strands with length $L$ from $T_1$, and produces a test tube $T_2$ with the removed strands;

6. $Cleavage(T, \sigma_0\sigma_1)$: for a given test tube $T$ and a string of two (specified) symbols $\sigma_0\sigma_1$ it cuts each double strand containing $\left(\frac{\sigma_0\sigma_1}{\sigma_0\sigma_1}\right)$ in $T$ into two double strands as follows:
\[
\begin{pmatrix}
\alpha_0\sigma_0\sigma_1\beta_0 \\
\alpha_1\bar{\sigma}_0\bar{\sigma}_1\bar{\beta}_1
\end{pmatrix} \Rightarrow \begin{pmatrix}
\alpha_0\sigma_0 \\
\alpha_1\bar{\sigma}_0
\end{pmatrix}, \begin{pmatrix}
\sigma_1\beta_0 \\
\bar{\sigma}_1\bar{\beta}_1
\end{pmatrix};
\]

7. **Annealing**\((T)\): for a given test tube \(T\) it produces all feasible double strands in \(T\). The produced double strands are still stored in \(T\) after **Annealing**;

8. **Denaturation**\((T)\): for a given test tube \(T\) it dissociates each double strand in \(T\) into two single strands;

9. **Discard**\((T)\): for a given test tube \(T\) it discards the tube \(T\);

10. **Append**\((T, Z)\): for a given test tube \(T\) and a given short DNA single strand \(Z\) it appends \(Z\) onto the end of every strand in the tube \(T\);

11. **Read**\((T)\): for a given tube \(T\), the operation is used to describe a single molecule, which is contained in the tube \(T\). Even if \(T\) contains many different molecules each encoding a different set of bases, the operation can give an explicit description of exactly one of them.

### 2.5 Background

Bio-inspired computing with DNA is one of the most exciting new developments in computer science, from both technological and theoretical points of view. When most of people think of a “DNA computer”, the first impression that arises to mind is a personal computer-like interface with many DNA microarray chips lined up inside a central processor and a keyboard that connect directly with several different linear DNA molecule’s 5’ ends. Perhaps in the future, DNA central processors and devices with DNA circuit gates will become reality. At this moment, however, “DNA
computer” is still a slightly misleading title applied to experiments in which DNA molecules have any intrinsic computational roles or even with conductivity as metal wires.

Often DNA sequences of about $8 - 20$ base pairs are used to represent bits, and numerous methods have been developed to manipulate and evaluate them. DNA is a convenient choice, as it is both self-complementary (i.e., $A = T$ and $G \equiv C$), allowing ssDNA to match its own Watson-Crick complement, and can easily be copied. Also, molecular biologists have found many biological tools can be used for manipulating DNA such as restriction enzyme digestion, ligation, sequencing, amplification, and fluorescent labeling, etc. Those all are giving DNA a big surplus over alternative computational media. This exceptional combination of computer science and molecular biology has captivated the world for over a decade, perhaps because it finally links two popular but greatly different disciplines that we have always hoped would be innately linked as seen in scientific fictions.

DNA computing is fundamentally comparable to the parallel computing, where the advantage is taken from many different DNA molecules trying on many various possibilities/combinations. Computing with DNA is also known as part of molecular computing which is a new approach to massively parallel computation based on a ground-breaking work by Adleman [2]. DNA molecules were first designed to solve a seven-node Hamiltonian path problem, a special case of an NP-complete problem that attempts to visit every node in a graph exactly once.

Since Adleman’s pioneering accomplishment in 1994, research on DNA computing is ongoing; Lipton [63] and Adleman [3] have extended on Adleman’s original work [2] with more efficient designs. Research on DNA computing has incorporated both
experimental and theoretical aspects. Theoretically, mathematical models of the computing process have been developed and their *in vitro* feasibility examined [10, 6, 89]. On the other hand, DNA computing has been applied to various fields experimentally including combinatorial optimization [79], cryptography [4], massive parallel computing [67], Boolean circuit development [7, 8, 80], nanotechnology [113, 90], very large scale database [87], forbidding-enforcing systems [31], etc.

In reality, various aspects of the ability to implement a DNA computer have been experimentally investigated. Gehani and Reif [37] and McCaskill [71] have constructed the microfluidic devices consisting of small gated channels that only one molecule can pass through at a time, where controlled biochemical reactions take place in a shorter time and more accurately than in conventional laboratories. These microfluidic devices can implement a dataflow-like architecture for processing DNA, for example, amplification, hybridization, separation and detection. Benenson *et al.* [15] focused on *in vivo* computing by using the technology on a smaller scale inside the cells. The realistic aim is to demonstrate control at a molecular level, who he built a programmable and autonomous computing machine made of biomolecules. This finite automaton is similar to the hypothetical Turing Machine, which is an outstanding achievement as a Turing Machine is capable of performing all mathematical operations and is regarded as the basis of today’s computers. Stojanovic and Stefanovic [100] and later expanded by Macdonald *et al.* [68] have built the first DNA-based molecular automaton, the MAYA (molecular array of YES and AND gates). MAYA-I/II can be programmed to play the game of tic-tac-toe triggered by a circuit of artificial enzymes.
The nature of DNA characteristics and biochemical reactions give a DNA computer a number of useful properties, for example, massively parallel processing capabilities and a huge memory capacity. A comparison between DNA computer and conventional computer is illustrated in Table 2.1.

2.6 Challenges

The potential of DNA computing is enormous; however, many researches have shown that it is constrained by major limitations. The method of Adleman’s model shows exponentially increasing volumes of DNA and linearly increasing time. Although DNA can store a trillion times more information than current storage media, the way in which the information is processed necessitates a massive amount of DNA if large scale problems are to be solved. Ouyang et al. [79] showed that restriction enzymes could be used to solve the NP-complete clique problem. The maximum number of vertices that they can process is limited to 27 because the size of the pool with the size of the problem exponentially increases. Furthermore, DNA chemistry produces in the results of the computation is uncertain. DNA strand synthesis is liable to errors, and is highly dependent on the accuracy of the enzymes involved.

Failures in DNA computation can derive from the following [69]:

**Failure to generate random solution.** In Adleman-Lipton model, the computation started from generating a random pool and filter out those unreasonable candidates. It becomes a problem that the amount of DNA needed to generate the initial pool that contains all possible solutions. This problem can be referred as coupon collector’s problem, which if we want to collect the entire set
Table 2.1: Comparison between DNA computer and conventional computer

<table>
<thead>
<tr>
<th></th>
<th>DNA Computer</th>
<th>Conventional Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Resource</td>
<td>Performance</td>
</tr>
<tr>
<td>STORAGE</td>
<td>Nucleic acid</td>
<td>Ultra-high efficiency / low cost</td>
</tr>
<tr>
<td>ENERGY</td>
<td>Chemical</td>
<td>Ultra-high efficiency / low cost</td>
</tr>
<tr>
<td>OPERATION</td>
<td>Chemical reaction</td>
<td>Simultaneous / parallel</td>
</tr>
<tr>
<td>PROCESS</td>
<td>Stochastic</td>
<td>Fast</td>
</tr>
<tr>
<td>PROCESSOR</td>
<td>Experiment</td>
<td>Slow (under development)</td>
</tr>
<tr>
<td>MOBILITY</td>
<td>In Laboratory</td>
<td>Labor intensive / slow</td>
</tr>
</tbody>
</table>
of $n$ different coupons by randomly drawing a coupon each trial, the drawing of $O(n \log n)$ is expected for collecting the entire set. Likewise, the amount of DNA required for creating all $2^n$ potential binary solutions shall be $O(n2^n)$ [62].

**DNA hydrolyzation.** DNA tends to be hydrolyzed if it is soaked in water over long periods of time.

**Failure of complementation.** For two strands to anneal, they must bump into each other in the solution. This failure is similar as problems are scaled up to real-world complexity with an exponential increase in the number of distinct strands in the solution. The time to anneal grows as the square root of the length of the strands and the reciprocal of the concentration of the desired “mates” [97].

**Failure to properly discriminate in annealing.** Liu *et al.* [66] discovered that in the surface-based DNA computation, a set of 16 different 16-mer sequences, only 9 of the complements could perfectly discriminate between the 16 target sequences at $37^\circ C$; the other 7 even annealed to some incorrect sequences at $40^\circ C$. Aoi *et al.* [12] tested errors in the process of annealing and found that an 8-mer sequence could only tolerate a single base mismatch and still successfully anneal. They also remark that $G$ and $T$ readily annealed at a single mismatch locus.

**Failure of annealing due to a secondary structure in the strands.** For a strand of DNA (or RNA) has subsequences that complement each other, it will tend to fold up into hairpins and other secondary structures as shown in Fig. 2.16. This can prevent the desired annealing between strands.
Figure 2.16: An example of nucleic acid has subsequences that complement each other fold up into hairpins structure.

**Incorporation of incorrect bases during PCR.** Errors could happen in polymerization during a PCR process. Depending on the polymerase enzyme used, the error can be between $7 \times 10^{-7}$ (for Pfu) and $2 \times 10^{-4}$ (for Taq) per base pair [20]. Moreover, PCR is designed with the assumption that the proportion of primer is far greater than the proportion of template in the solution. If there are a lot of templates, the interactions between the templates can interfere with the PCR [50].

**Failure to extract a target sequence from a solution.** Both Karp et al. [52] and Roweis et al. [88] replace a single separation with a series of separations. They trade off time and space (number of test tubes) for accuracy. If the separation step divides a set of DNA strands into a “true” test tube and a “false” test tube, the simplest version of this is to repeat the separation step on the “false” test tube, in order to lower the rate of false negatives. A more sophisticated version of this is to do repeated separations on both results, creating an array of test tubes $1, \ldots, n$ where test tube $k$ contains all those strands that have been processed as “true” $k$ times. Thus a DNA strand does a biased random walk across the array. Roweis et al. suggest fixing the size of the array and
making absorbing boundaries out of the test tubes at the ends. They point out that any DNA strands that reach the boundary test tubes can immediately be used in the next step of the computation. This creates a form of pipelining. By increasing the size of the array and thus the number of separation steps that must be performed, the error rates can be made arbitrarily small.

Khodor and Gifford [53] took a hard look at sequence separation based on biotinylation, with rigorous controls and measurements. Under ideal conditions, they were able to retrieve only 8 – 24% of the desired strands, although there were few false positives (< 1%). This suggests that if sequence separation is called for in an algorithm, there must be many copies present of the desired sequence. Khodor and Gifford also note that if the target sequence is rare relative to the other sequences in the solution, even a small false positive rate will leave the target sequence in the minority in the resulting “positive” vial. These problems have prompted researchers to avoid sequence separation by annealing and instead use restriction enzymes to digest undesired sequences [79, 9, 27].

**Loss of DNA strands.** DNA may stick to the walls of test tubes when a solution is poured from one tube into another. Khodor and Gifford [53] found that significant portions of the population of DNA strands were completely lost in the process of sequence separation, appearing in neither the positive nor the negative test tubes after the separation. This implies that a sequence separation step of an algorithm should generally be followed by PCR amplification.
Failure of the split operation. If an algorithm relies on equally dividing the contents of a test tube into two test tubes, there is no guarantee that the copies of a particular sequence in the test tube will be equally divided between the two tubes. Bach et al. [14] model this with a “probabilistic split” operator and use their analysis to derive bounds on the amount of redundancy they need to find a desired solution with reasonable probability.

Failure to separate by length in a gel electrophoresis. This failure can result from a host of problems. Gels are not perfectly even in their porousness. Strands of DNA can stick to each other, or form secondary structures that affect their movement through the gel. They can also simply impede each other’s progress in microscopic traffic jams. It can be extremely hard to separate strands of similar length. After two cycles of electrophoresis and PCR, Adleman had to repeat the electrophoresis four times on the product of the third PCR [50], each time removing the 120-bp band in an attempt to reduce contamination.

Failure of restriction enzymes. Restriction enzymes might fail to cut the strands at their restriction sites due to lack of time, secondary structure in the strands, or other unknown interferences. Amos et al. [11] found they were only able to destroy 2 out of 3 target sequences through the use of restriction enzymes. Their technique was to anneal a primer to the target sequence and then use a restriction enzyme that only cleaves double-stranded DNA (dsDNA). Their failure thus may have been due to problems in annealing of the primer to the target sequence. In contrast, Wang et al. [106] reported 97% efficiency of their restriction enzyme.
Failure of ligation due to secondary structure in the strands. Because ssDNA (and RNA) can fold up into complex shapes, these shapes may interfere with the ability of ligase to ligate the ends of two strands.
Chapter 3

Bio-soft Computing with Fixed-length DNA to a Group Control Optimization Problem

“All progress depends on the unreasonable man.”

- George Bernard Shaw, Irish dramatist, literary critic, and socialist.

3.1 Overview

Group control are playing an increasingly important role in transportation and manufacturing. Many group control systems are required to operate in environments that involve uncertainty. Elevator group control, for example, is one of applications which many possible situations are comprised. Factory scheduling is such another application.

The elevator group control system is a control system that manages multiple elevators in a group systematically to increase the service for passengers. Group elevator scheduling is a complicated control problem which has been researched extensively
due to its high practical significance. The problem is briefly stated: New passengers arrived at a bank of elevators at random times and floors, making hall-calls to signal for rides up or down. A ride destination is unknown until the passenger enters the car and makes a car call request a stop. The controller must assign a car to each hall call in a way that optimizes overall system performance [77].

At the early days when the first passenger elevators were introduced in the 1890’s, each car was individually controlled by an attendant riding the car. As building heights rose, however, so did the number and speed of the cars, it became impossible for the attendants to provide effective coordination and control. With the first semiautomatic elevator controller was introduced in the 1920’s, the attendant’s job was reduced to one of simply closing the doors and starting the car. By 1950, fully automated elevator controllers eliminated the attendant’s job altogether. The first automated elevator controller was a simple electromechanical relay system. By the 1970’s, microprocessor based elevator controllers were in common use [82]. The task of controlling a group of elevators to provide good traffic performance is complex. The routes of the cars play a vital role in the performance of the vertical transportation system. In the elevator community, this problem is referred to as “landing call allocation problem” or “elevator dispatching problem”.

The elevator dispatching problem may be considered as a version of the traveling salesman problem (TSP), which is a classified NP-hard optimization problem. In the context of elevator car routing, the TSP problem can be converted to multiple traveling salesmen problem, where salesmen (elevator cars) visit the cities (car calls and hall calls) so that each city (call) is visited only once and the cost function

$$\sum C(S_i, C_i \cup H_i)$$  \hspace{1cm} (3.1.1)
is minimized. The partial cost $C$ is gathered along the route when a set of cars and hall calls $C_i \cup H_i$ is visited by the elevator $i$, starting the roundtrip from the elevator’s initial state $S_i$. The size of the problem space is

$$P_s = M^n$$

where $M$ is the number of elevators and $n$ is the number of active hall calls [103].

### 3.2 Elevator Dispatching Problem

In the elevator group control, there are two types of calls. A hall call is given through buttons on a hall of a building, and a car call is given in an elevator by passengers. An elevator group control system has a pair of hall call buttons on each floor, one for up hall call and the other for down hall call. If a passenger presses a hall call button, an elevator is selected by the group control system for the passenger.

Assume a building of $N$ floors equipped with $M$ identical elevator cars, an elevator routing path can then be represented as a weighted-graph problem where each elevator floor position $1, 2, 3, \ldots, N - 1, N$ can be represented with floor nodes $F_1, F_2, F_3, \ldots, F_{N-1}, F_N$, respectively. The movement of each elevator consists of two directions, upward and downward, which is distinguished by a notation ‘(prime). The node with a prime stands for downward movement. Figure 3.1 illustrates all possible movements of a single elevator.

The elevator moves between floors, denoted $m$, must be consecutive, \textit{i.e.},

$$d(m_i) = o(m_{i+1}) \lor 1 \leq i < N$$

where
Figure 3.1: Graph of all possible movements for a single elevator

The traveling time $T$ between floors can be represented as

$$T(|d(m) - o(m)|) = \begin{cases} ||d(m) - o(m)|| C_t + C_s, \quad m=\text{designated}; \\ 0, \quad \text{otherwise.} \end{cases} \quad (3.2.2)$$

where

- $|d(m) - o(m)|$ is the total number of floor elevator moves
- $C_t$ is the time costs for elevator traveling between adjacent floors
- $C_s$ is the time costs for elevator stops

The output of the graph $G$ given by the sum of the costs thus represents the total time of traveling of the elevator $E$, i.e.,

$$G(E) = \sum_{m=1}^{N} T(|d(m) - o(m)|) \quad (3.2.3)$$
For a building with $M$ elevators, the graph of a single elevator movements shown in Fig. 3.2 can be duplicated $M$ times in representing the whole paths of elevator traveling. The total traveling time of $M$ elevators can thus be calculated by summing up the traveling time of each single elevator as

$$G(E_1, E_2, \ldots, E_{M-1}, E_M) = \sum_{k=1}^{M} G(E_k)$$  \hspace{1cm} (3.2.4)

The optimal travel route, denoted $O$, is thus given by the minimum total traveling time of all elevators with all initial conditions and requirements satisfied, i.e.,

$$O = \min\{G(E_1E_2, \ldots, E_{M-1}E_M)\}$$  \hspace{1cm} (3.2.5)

### 3.3 Background

As the elevator dispatching problem is too large to be solved systematically, there are many methods have been applied from previous researches. Examples that have employed are intelligent agents [26, 96], fuzzy logic [55, 104], neural networks [46, 84, 94] and evolutionary and genetic algorithms [34, 98].

The massive parallelism of the bio-soft computing provides an opportunity to solve a host of difficult problems, especially the NP (nondeterministic polynomial time) problems, which are a class of mathematical problems that have most likely exponential complexity. Many scientists are researching for an efficient solution to the NP problems with DNA computing [79, 90], however, none of these works involving with the representation of numerical data. In reality, many practical applications engage with weighted graph, which each branch in a graph is given a numerical weight.
Definition 3.3.1. A weighted graph $G = (V, E)$ is one where each edge $(i, j) \in E$ has associated with a weight $w_{ij}$.

The concept of a weighted graph is extremely useful with many applications, for example, the traveling salesman problem, the minimum spanning tree and the shortest path problem, and yet the representation of cost information for weighted-graph problem in bio-soft computing with DNA is a challenging problem. Hence, the numerical data representation in DNA strands is an important issue toward expanding the capability of DNA computing to solve numerical optimization problems [60].

The first attempt to solve the elevator dispatching problem by DNA computing was introduced by Watada et al. [109, 108, 48]. A preliminary bio-soft computing algorithm with length-based DNA was proposed where a dummy strand of DNA was designed to represent the cost. Later, Muhammad et al. [75] adopt the concept and refine the procedure.

The length-based DNA computing was originally proposed by Narayanan and Zorbalas [76] to solve a traveling salesman problem. The sequence of edge was represented by the length of DNA that was proportional to the edge weight. In this case, the representation of a heavier weight would have a longer DNA sequence. This method is inefficient in representing a wide range of weights because the DNA length of heavy weights would be extremely long. It is unrealistic and making experimental implementation impossible [60]. In addition, the fact that heavier weights are encoded as longer sequences is contrary to the biological fact, which as the length of the DNA strands being used increases, so does the probability of incorrect hybridization [6].

Yamamura et al. [115] proposed a method which represents the numerical values
by the concentration of DNA strands. The numerical data were encoded by controlling the DNA strands concentrations as input and output. This may benefit from finding the solution in candidate pool locally rather than searching entire space globally. This method, however, may not guarantee an optimal solution that result from the brightest band in gel electrophoresis. Besides, the brightest band might contain several kinds of DNA strands, and is technically impossible to extract a single optimal solution [60]. In this proposal, Yamamura et al. were unsuccessful in finding an optimal solution.

Shin et al. [95] proposed a method which represents the numerical values with fixed-length DNA strands. The concept is to design the DNA strands that represent various weights in a constant length but varied from the number of hydrogen bonds in the strands. In their proposal, the simulation was presented to prove its feasibility. Later, Lee et al. [59, 60] further modify the protocol. Instead of being controlled by the number of hydrogen bonds, they change the factor into melting temperature (\(T_m\)) of DNA strands, the energy required to break the hydrogen bonds. This method overcomes the drawbacks of previous cost representation methods and solved a 7-city 5-weight traveling salesman problem. The basic idea is to design the cost sequences with various \(T_m\) according to the costs. A smaller cost is represented by a DNA sequence with a lower \(T_m\), and therefore a more economical path has a lower \(T_m\).
3.4 Molecular Algorithm for Solving Elevator Dispatching Problem

For the elevator dispatching problem, an $N$-story building equipped with $M$ identical elevator cars given by up hall calls, down hall calls, and car calls. The optimal route is given by $\min \{G(1, 2, \cdots, N)\}$.

The key factor to cost sequence design is the $T_m$ of a DNA strand. The concept is to design the DNA sequences that have heavier weights with higher $T_m$ then those lighter weights. DNA amplification and detection techniques often depend on oligonucleotide $T_m$. The $T_m$ of a DNA duplex is defined as the temperature where one-half of the nucleotides are paired and one-half are unpaired [111]. The $T_m$ indicates the transition from double helical to random coil formation and is related to the DNA GC base content [70]. Usually expressed as a percentage, it is the proportion of GC-base pairs in the DNA molecule or genome sequence being investigated. As mentioned in Section 2.2, GC-pairs in the DNA are connected with three hydrogen bonds instead of two in the AT-pairs, which makes the GC-pair stronger and more resistant to denaturation by high temperatures. In our encoding scheme, the DNA sequences that represent floor nodes are fixed length, and costs are distinguished by $T_m$ of the given DNA strands. This design makes the economical path having the oligonucleotide with lighter weight/lower $T_m$.

All the possible solutions are randomly generated by DNA hybridization and ligation with the oligonucleotides representing floors, edges, and costs. To satisfy the condition of an elevator dispatching problem, the route must begin and end at a specified node, and the route must pass by each consecutive floor until reaches the
final destination.

The PCR can be applied to test the former requirement, which it is a technique for amplifying DNA that rapidly synthesize many copies of a specific DNA segment by providing specific complementary sequences (primers) and enzymes (DNA polymerases, for example, *Pfu* and *Taq*). The DNA strands corresponding to the original floor and complement of the final destination floor are used as two primers in two successive PCRs to reproduce the routes.

To test the latter requirement, agarose gel electrophoresis is applied. Agarose gel electrophoresis is a method to separate DNA strands by size, and to determine the size of the separated strands by comparison to strands of known length. All PCR products are sieved by agarose gel electrophoresis, and the unreasonable lengths are excluded. To verify the DNA strands pass by every consecutive floor, the product from the above step is affinity-purified with a biotin-avidin magnetic beads system.

To solve the elevator routing problem with our molecular algorithm, note that all possible end paths of elevator $i$ are jointed with the start path of elevator $i + 1$ so that the total output of the graph $G(1, 2, \cdots, N)$ representing the travel route of all elevators can be calculated.

DTG-PCR is a specified PCR protocol that modifies the denaturation temperature profile as mentioned in Section 2.3. If the denaturation temperature is decreased to a certain level in PCR, the DNA strands with denaturation temperatures lower than that temperature will be denatured and amplified. As the denaturation temperature is increased cycle by cycle in PCR, other DNA strands with higher denaturation temperature will also be amplified. However, the economical paths that have lighter weights will be amplified more and will occupy the major part of the solution and
hence can be easily detected [60]. Based on the electrophoretic mobility of DNA strands in different $T_m$, the TGGE mentioned in Section 2.3 is applied to detect the most economical route among other possibilities resulting from DTG-PCR.

The proposed bio-soft computing algorithm for solving the elevator dispatching problem is summarized following:

**Step 0** Design the fixed-length DNA sequences with thermodynamic control in weight;

**Step 1** Generate a random pool of solution by hybridization and ligation;

**Step 2** Retrieve the strand that satisfy the condition of elevator dispatching problem by PCR and gel electrophoresis;

**Step 3** Verify the strands that pass by every consecutive floor by affinity purification;

**Step 4** Joint the strands of elevators by ligation;

**Step 5** Sieve the economical path by DTG-PCR;

**Step 6** Detect the most economical path by TGGE;

**Step 7** Readout the most optimal route by DNA sequencing.

### 3.5 Empirical Study

A six-story building equipped with two identical elevator cars, A and B, is considered in the empirical study. As illustrated in Fig. 3.2, elevator A is currently at the 1st floor upward answers the car calls on 3rd and 5th floor. Elevator B is at the 6th floor downward answers the car calls on 2nd and 3rd floor. In addition, hall calls
are requested on 3rd, 4th, and 5th floor for down, up, and down, respectively. The objective is to find the optimal route for all elevators that fulfill all initial conditions and requirements defined. The optimal route will be given by $\min\{G(A, B)\}$.

Figure 3.2: Example of a six-story building equipped with two identical elevator cars, A and B

As listed in Table 3.1, each floor node is randomly associated with a 20-mer sequence of ssDNA, denoted $F_i$, which has a similar melting temperature due to node sequences should contribute equally to the thermal stability of paths. Weight sequences are designed to have different $T_m$ depending on its weight. The lighter in weight, the lower the $T_m$ is. In other words, more economical path has lower $T_m$. The edge between consecutive floors is generated by partial beginning node, cost sequence, and partial ending node. For each floor movement (edge) $i \rightarrow j$ in the graph, an oligonucleotide $F_{i\rightarrow j}$ is created that is the 3’ 10-mer complement of $F_i$ followed by the cost sequence of path length, and then the 5’ 10-mer complement of $F_j$. The edge from floor 1 to floor 2, for example, the 3’ 10-mer complement of $F_1$: ‘CATGACAACG’ followed by the cost sequence of $C_i$: ‘ATCTTGGATTATCCAAAG’, then the 5’ 10-mer complement of $F_2$: ‘TGGCTACATG’, as illustrated in Fig. 3.3.
In this study, nearest-neighbor (N-N) model is applied to calculate the \( T_m \), which is the most accurate method for predicting the \( T_m \) of oligonucleotide DNA through interactions between neighboring bases. The enthalpy (\( \Delta H \)) and entropy (\( \Delta S \)) of adjacent bases is considered in the formula [93]. The \( T_m \) in this study was calculated with the initial concentration of 1\( nM \) oligonucleotide and 50\( mM \) salt.

Table 3.1: DNA sequences associated with each floor node and cost for the six-floor elevator routing problem.

<table>
<thead>
<tr>
<th>Floor Nodes</th>
<th>DNA Sequence (5′ → 3′)</th>
<th>( T_m ) (°C)</th>
<th>GC Content (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_1 )</td>
<td>TCCTCGTAGTTAGTGGTTGC</td>
<td>46.02</td>
<td>50</td>
</tr>
<tr>
<td>( F_2 )</td>
<td>ACCGATGTACCTCTCAATGC</td>
<td>46.40</td>
<td>50</td>
</tr>
<tr>
<td>( F_3 )</td>
<td>TGGTCAGCTAATGACGTAG</td>
<td>46.42</td>
<td>50</td>
</tr>
<tr>
<td>( F_4 )</td>
<td>CGGTTCTAATCCGGTAC</td>
<td>46.51</td>
<td>50</td>
</tr>
<tr>
<td>( F_5 )</td>
<td>ATGGGACCCAGATGCGAAAG</td>
<td>46.92</td>
<td>50</td>
</tr>
<tr>
<td>( F_6 )</td>
<td>GTTAGACCTCGGTTGCTAT</td>
<td>46.97</td>
<td>50</td>
</tr>
<tr>
<td>( F'_1 )</td>
<td>CGTAACTCGTAGGCTAGA</td>
<td>46.58</td>
<td>50</td>
</tr>
<tr>
<td>( F'_2 )</td>
<td>TAGCCTTAGCTACGGGCTTA</td>
<td>46.84</td>
<td>50</td>
</tr>
<tr>
<td>( F'_3 )</td>
<td>CGTAACGTATGCGATGGA</td>
<td>46.22</td>
<td>50</td>
</tr>
<tr>
<td>( F'_4 )</td>
<td>GACGCTTAATGGTGAATTCGA</td>
<td>46.48</td>
<td>50</td>
</tr>
<tr>
<td>( F'_5 )</td>
<td>ATCGGAAATCGATCGTATAG</td>
<td>46.88</td>
<td>50</td>
</tr>
<tr>
<td>( F'_6 )</td>
<td>AGCTGGGATAAGCCATACCA</td>
<td>46.76</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Costs</th>
<th>DNA Sequence (5′ → 3′)</th>
<th>( T_m ) (°C)</th>
<th>GC Content (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_t )</td>
<td>ATCTTGGATTTTTACCAAG</td>
<td>36.98</td>
<td>30</td>
</tr>
<tr>
<td>( C_s )</td>
<td>GAGCCGACCAGGACACCCCA</td>
<td>55.84</td>
<td>70</td>
</tr>
</tbody>
</table>

The proposed fixed-length DNA based algorithm for solving the elevator dispatching problem began from generating of a random pool of possible routes by the hybridization of DNA strands that represent the floors and edges. All possible paths of the elevator dispatching problem was generated simultaneously under the massive
parallelism of DNA molecules. For each $F_i$ and for edge $i \rightarrow j$ were mixed in a single ligation reaction. Per [60], the added amount of an edge was varied according to weight, where as the weight increased, the amount was decreased. The oligonucleotide mixture was heated to $95^\circ C$ and cooled to $20^\circ C$ at $2^\circ C/min$ for hybridization. The reaction mixture was then subjected to a ligation.

The conventional gel electrophoresis excluded the unreasonable length of DNA strands from candidate pool. Then, the DNA strands that were not passing by every floor nodes between origin and destination were excluded by affinity separation. The complement of $F_1$ was conjugated to magnetic beads so that only those ssDNA molecules which contained the sequence $F_1$ annealed to the bound were retained. This process was repeated until each floor node was verified.

In DTG-PCR, the denaturation temperature started at low temperature in $70^\circ C$ in the beginning cycles of PCR, which is lower than the $T_m$ of template strands. Then, the denaturation temperature was gradually increased until reached $95^\circ C$ and maintain at the same temperature for the remaining cycle. After this process, one main band was observed in the gel which contained two different DNA strands of the possible routes as shown in Table 3.2. These strands, however, were of the same length which cannot be separated by the conventional gel electrophoresis.
Nevertheless, from the algorithm design, the weights have their distinct behaviors in $T_m$ and thus the more economical path would have a lower $T_m$. Thus, TGGE can be used to filter the DNA strands that have lowest $T_m$ among other strands of the same length. Based on the correlation of the melting characteristic of a DNA strand to its electromigration, the DNA strand of the most economical route would travel fastest in gel; hence, it can be distinguished from other possible routes. The $T_m$ and its GC content among those possible routes are shown in Fig. 3.4, which the DNA strands corresponded to the route for elevator A: ‘1 → 2 → 3 → 4 → 5’ that answers the hall call at 4th floor and 5th floor for up and down, respectively, and elevator B: ‘6 → 5 → 4 → 3 → 2’ that answers the hall call at 3rd floor for down showed the optimal solution.

![Figure 3.4: The distinct behavior in melting temperature among the possible optimal solutions.](image)

<table>
<thead>
<tr>
<th>Elevator A</th>
<th>Elevator B</th>
<th>$T_m$(°C)</th>
<th>GC Content(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 → 2 → 3 → △ → 5</td>
<td>6' → ▽ → 4 → ▽ → △ → 7</td>
<td>62.93</td>
<td>54.44</td>
</tr>
<tr>
<td>1 → 2 → 3 → △ → ▽</td>
<td>6 → 5 → 4 → ▽ → △</td>
<td>61.92</td>
<td>52.22</td>
</tr>
</tbody>
</table>

Figure 3.4: The distinct behavior in melting temperature among the possible optimal solutions. ○: car call; △: up hall call; ▽: down hall call.

### 3.6 Concluding Remarks

In this chapter, a bio-soft computing algorithm with fixed-length DNA was proposed to solve the elevator dispatching problem. The designing concept and empirical study were presented in details. It is significant that bio-soft computing application towards the problem of group control optimization have been shown to be achievable.
Table 3.2: Result from DTG-PCR. Two different DNA strands represent the possible routes of the same length which cannot be separated by conventional gel electrophoresis.

<table>
<thead>
<tr>
<th>DNA Sequence (5’ → 3’)</th>
<th>Oligo Length (Bases)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCCTCGTAGGAGCCATCTGATGCGTGATTTATTACCAAG</td>
<td>360</td>
</tr>
<tr>
<td>ACCGATGACCTCAATCGCGAGCGGACCACGCGACACCCA</td>
<td></td>
</tr>
<tr>
<td>TGGTCAGCTAAATGAGGAGCCTTTGGGATTTATTACCAAG</td>
<td></td>
</tr>
<tr>
<td>ACCGATGACCTCAATCGCGAGCGGACCACGCGACACCCA</td>
<td></td>
</tr>
<tr>
<td>TGGTCAGCTAAATGAGGAGCCTTTGGGATTTATTACCAAG</td>
<td></td>
</tr>
<tr>
<td>GAGCCGACCAGCGACACCACCCATCGGAATCGATCCGTATGC</td>
<td></td>
</tr>
<tr>
<td>ATCTTTGATTTTATTACCAAGGAGCCTTTGGGATTTATTACCAAG</td>
<td></td>
</tr>
<tr>
<td>GAGCCGACCAGCGACACCACCCATCGGAATCGATCCGTATGC</td>
<td></td>
</tr>
<tr>
<td>GAGCCGACCAGCGACACCACCCATCGGAATCGATCCGTATGC</td>
<td></td>
</tr>
<tr>
<td>GAGCCGACCAGCGACACCACCCATCGGAATCGATCCGTATGC</td>
<td></td>
</tr>
<tr>
<td>GAGCCGACCAGCGACACCACCCATCGGAATCGATCCGTATGC</td>
<td></td>
</tr>
</tbody>
</table>
There are a number of limitations that prevent us from finding the optimal solution in our proposal. Some of these limitations arise from the algorithm while others arise from the limitations of the biochemical techniques. Among the algorithmic limitations, our method has difficulty finding optimal solution real-time and continuously. So far we can only achieve a local optimization due to the insufficient development of DNA computing. Automation is certainly a future challenge for DNA computing that may possible fulfills our proposal.

Other limitations arise from the biochemical techniques themselves. For example, the precision of temperature control and the errors occurred in biochemical reactions. Although the proposed algorithm is still far from maturity, it is worth emphasizing the fundamental novelty of this study.
Chapter 4

Bio-inspired Evolutionary Method for Cable Trench Problem

“There are no secrets to success.

It is the result of preparation, hard work, and learning from failure.”

- Colin Powell, American military leader and statesman.

4.1 Overview

The cable trench problem (CTP) is classified as network routing problem. It is a combination of the shortest path problem (SPP) and minimum spanning tree (MST) problem. These two problems are universally discussed in operations research and management engineering, and its combination is even more difficult to be solved by a conventional computer. CTP is an NP-complete problem, in which there is a tradeoff between the fixed cost associated with constructing the network and a variable cost associated with operating it [86].
CTP comes from the fact that a physical application of this problem is the connection of buildings on a university campus with the building housing the main computer. The main computer building is $v_0$ and the buildings to be connected to the main computer building are the other vertices. The only allowable routes between two buildings for digging trenches and laying cables define the edges of the graph. Since only the edges that have endpoints that must be connected by cable to the main computer building are allowed as part of the solution spanning tree, the solution to the problem will not be a Steiner tree. A trench may carry more than one cable once it is dug and the trench cost is proportional to the total trench distance. The cable cost is proportional to the total length of the cable required [105].

Practically, CTP has many applications, for example, marine transmission cable underlying, oil pipeline layout, and network flow planning to name a few. The previous researches on CTP were done in conventional computing method. In this chapter, we expand the capability of DNA computing to solve a network routing problem. The thermodynamic controlling method with fixed-length DNA is applied to solve the CTP.

### 4.2 Cable Trench Problem

Conceptually, the CTP is a combination of the MST problem and SPP.

**Definition 4.2.1.** A MST of a weighted graph $G = (V, E)$ with edge weights $w_{ij}$ is the spanning tree $G' = (V', E')$ which minimizes the sum $\sum_{(i,j) \in E'} w_{ij}$.

**Definition 4.2.2.** A directed graph or digraph $G = (V, E)$ is a set $V$ of vertices and a set of ordered pairs $(i, j)$ where $i, j \in V : i \neq j$. (Each member of $E$ is called a
Definition 4.2.3. A SPP from $O$ to $D$ as finding the directed path $(n_1, \cdots, n_l)$ where $n_1 = O$ and $n_l = D$ which minimizes the sum $\sum_{i=1}^{l-1} w_{n_in_{i+1}}$ where $w_{ij}$ is the weight of the edge from vertex $i$ to vertex $j$.

A connected graph $G = (V, E)$ with specified vertex $v_0 \in V$, the length of each edge $e \in E$ where $l(e) \geq 0$, and let $\tau$ and $\gamma$ denote positive parameters. The solution to the CTP is the spanning tree $T$ that minimizes $\tau l_{\tau}(T) + \gamma l_{\gamma}(T)$ where $l_{\tau}(T)$ is the total length of the spanning tree $T$ and $l_{\gamma}(T)$ is the total path length in $T$ from $v_0$ to all other vertices of $V$. In other words, given a spanning tree $T$ of graph $G$, denote the total length of the spanning tree by $l(T)$ and the sum of the path lengths $p_i$ from node 0 to each node $i$ in $T$ by $s(T)$. The CTP examines the tradeoff between $s(T)$ and $l(T)$. The weighted sum version of the CTP is to find $T$ such that $\alpha l(T) + (1 - \alpha)s(T)$ is minimized.

CTP that associates the SPP and MST problem together becomes more practical. It has not been exploited in previous literature until Vasko et al. [105] proposed recently. The difficult of the CTP comes from the complexity of solving the weighted sum version depends heavily on the value of $\alpha$. If $\alpha \rightarrow 1$, then the solution to this problem is a MST. On the contrary, if $\alpha \rightarrow 0$, then the problem becomes a SPP. Solving this problem, however, for values of $\alpha$ arbitrarily close to one, i.e., finding among all MST the spanning tree $T$ that minimizes $s(T)$, is an NP-hard optimization problem, whereas the problem of finding the shortest paths tree that minimizes $l(T)$ can be solved in polynomial time. The cases $\alpha = 0$ and $\alpha = 1$ are of course both solvable in polynomial time, and hence, the ideal point can be computed in polynomial time. For general $\alpha$, this problem is NP-hard. Following proof of this fact is shown
Theorem 4.2.1. Finding the MST that minimizes the path length between a particular set of vertices \( s \) and \( t \) is NP-complete.

Proof. Eppstein provides a proof in [32] by a reduction from 3SAT. Readers may also refer to Appendix A.

Corollary 4.2.2. Finding the MST that minimizes total path length from \( v_0 \) to all other vertices in \( V \) is NP-complete.

Proof. Let \( s = v_0 \) and \( t \) be any vertex in \( V - \{v_0\} \), then this problem is NP-complete because it is a more general case of the problem proven to be NP-complete in Theorem 4.2.1.

Corollary 4.2.2 shows that, in general, finding the spanning tree is NP-complete.

Corollary 4.2.3. The CTP is NP-complete.

Proof. The CTP is a more general case of the problem proven to be NP-complete in Corollary 4.2.2.

Theorem 4.2.4. Finding a spanning tree that is a shortest-path solution from \( v_0 \) to all other vertices in \( V \) such that total edge length is minimized is in \( P \).

Proof. Find the shortest distance from \( v_0 \) to each vertex of \( V - \{v_0\} \). Find the set of all directed edges that can be part of a shortest path, i.e., an edge \((u, w)\) is in this set if \( l(u, w) = \text{distance}(v_0, w) - \text{distance}(v_0, u) \). Find a minimum spanning arborescence of this directed graph. All the above steps can be carried out in polynomial time.
The weighted sum version of the CTP exhibits the very interesting property that the difficulty of a particular instance depends heavily on the actual weight, and is hard to be solved by a conventional algorithm. As CTP is weight-dependent, this motivates us to investigate on possible solutions by fixed-length DNA computing with thermodynamic control.

4.3 Background

Both the SPP and the MST problem are the classic topics in graph theory which has been the target of many research efforts over the years. These research efforts have resulted in a number of different algorithms and a considerable amount of empirical findings with respect to performance.

The computation of shortest paths is an important task in many network and transportation related analyses. The development, computational testing, and efficient implementation of shortest path algorithms have remained important research topics within related disciplines such as operations research, management science, geography, transportation, and computer science [29, 28, 39, 5, 41]. These research efforts have produced a number of shortest path algorithms, as well as extensive empirical findings regarding the computational performance of the algorithms, for example, [40, 35, 74, 24].

Recently, a few researchers have proposed the algorithm based on DNA molecules in dealing with SPP. Ibrahim et al. [44] proposed the direct-proportional length-based DNA computing algorithm which the cost of an edge is encoded as direct-proportional length DNA. This is a protocol that improved from constant proportional length-based DNA computing proposed by Narayanan and Zorbalas [76]. Later, Ibrahim
et al. [45] further modified [44] and proposed the concentration-controlled direct-proportional length-based DNA computing. This protocol adds up the strategy of concentration control that takes the advantage of bias. Their design was to set the amount of DNA that representing the edges varies closely to the weight of edges. This is a combination of concentration-controlled DNA computing proposed by Yamamoto et al. [115] and biased in vitro experiment conducted by Lee et al. [60]. Wang et al. [107] presented a procedure for solving the SPP in the Adleman-Lipton model. Their procedure works in $O(n)$ steps for the SPP of an edge-weighted graph with $n$ vertices.

On the other hand, MST problem is one of the fundamental and best-studied optimization problems in computer science. It has obvious applications to network organization and touring problems. Many algorithms has proposed to solve the MST efficiently, the most classical algorithms are Kruskal’s [57] and Prim’s [85]. Chen and Zhang [22] define a constrained MST problem. They apply the divide-and-conquer technique to reduce the complexity of the problem and provide an $O(n^2)$ algorithm. However, other constrained versions of the MST problem, for example, Eppstein [32], Garey and Johnson [36], and Zhou and Gen [116], have proven to be NP-complete.

Several scholars have researched on the combination of SPP and MST problem. Booth and Westbrook [18] developed a linear algorithm that can be used to perform edge cost sensitivity analysis for both problems in a planar graph. Saltzman [91] provided a counterexample to the question “Is a MST also a shortest path tree for a determined vertex in a positive weighted connected graph?”. Khuller et al. [54] proposed an algorithm that finds a spanning tree in which the distance between any
vertex and the root of the shortest path tree is at most $1 + \sqrt{2}d$ times the shortest-path distance, and yet the total weight of the tree is at most $1 + \sqrt{2}/d$ times the weight of a MST where $d > 0$ is given. Dudás and Rudolf [30] examined the best possible algorithm in terms of complexity for three problems on Monge graphs: the MST problem, the problem of computing all-pairs shortest paths, and the problem of determining a minimum weighted 1-to-all shortest path tree. Liu and He [65] studied two network improvement problems with given discrete values: the inverse MST problem and the reverse SPP, where the decrements of the weight of the edges are given discrete values. They have shown the reverse SPP is strongly NP-complete. Eppstein [32] proved that finding the minimum spanning tree that minimizes the path length between a particular set of vertices is NP-complete. Vasko et al. [105] proposed a heuristic neighborhood search procedure that finds the optimal CTP solution by zero-one mixed integer linear programming.

### 4.4 Molecular Algorithm for Solving CTP

For a CTP, a graph $G = (V, E, w)$ with specified vertex $V_1 \in V$ and given edge $E$ with weight $w$. The objective is to find the optimal solution to CTP by a DNA molecular algorithm. A fixed-length DNA computing with thermodynamic control algorithm is proposed, which is similar to the one that solved the elevator dispatching problem, but differ from the strategy.

The concept of encoding scheme is to design the fixed length DNA strands in different weights that varied from $Tm$. The heavier weights with higher $Tm$ and vise versa. Hybridization and ligation are carried out to generate the initial DNA pool of possible solutions.
To satisfy the condition of a CTP, the route must begin at the specified vertex \( V_1 \) and the route must pass by all vertices \( V_2 \) to \( V_n \). PCR is applied to examine the former requirement through appointed primers. However, unlike the Hamiltonian path problem Adleman proposed [2], the route of CTP may end at any vertex that hard to predict. Hence, the optimal route is detected by the route begin at \( V_1 \) and pass by every vertex, then return with the same route that back to \( V_1 \). In other words, the CTP route encoded by DNA makes a round trip that back to where it original starts. The DNA strands of \( V_1 \) and complement of \( V_1 \) are used as two primers in two successive PCRs to reproduce the routes that both begin and end at \( V_1 \). Since the variation of \( Tm \) is increased doubly as the optimal solution is detected round-trip based, a more accurate result in temperature gradient methods (i.e., DTG-PCR and TGGE) can be benefited. The technique of gel electrophoresis is taken to verify the latter requirement, which the feasible solution should pass by all vertices.

The rest procedures are identical to the protocol we proposed for solving the elevator dispatching problem. The proposed bio-soft computing algorithm for solving CTP is summarized as follows:

**Step 0** Design the fixed-length DNA sequences with thermodynamic control in weight;

**Step 1** Generate a random pool of solution by hybridization and ligation;

**Step 2** Retrieve the strand that satisfy the condition of CTP by PCR and gel electrophoresis;

**Step 3** Verify the strands that pass by every consecutive floor by affinity purification;

**Step 4** Sieve the economical path by DTG-PCR;
Step 5 Detect the most economical path by TGGE;

Step 6 Readout the most optimal route by DNA sequencing.

4.5 Empirical Study

The newly developed Kitakyushu Science and Research Park (KSRP) is focusing its efforts on the formation of a nucleus for cutting-edge science and research through collaboration between industry and academia. In the park, a central security system is controlling the access to all facilities. As shown in Fig. 4.1, the central security control room and all other facilities are the vertices to be connected. The only allowable routes between two facilities for digging trenches and laying cables define the edges of the graph. A trench may carry more than one cable once it is dug and the trench cost is proportional to the total trench distance. The cable cost is proportional to the total length of the cable required. The weight is represented numerically on the edge that connecting two facilities. The objective is to find the most economical route to bury the cable that connecting all facilities to the central security room.

The above description can be further translate into a CTP which given a graph $G$ with vertices $V = \{1, 2, 3, 4, 5, 6\}$ and edges $E = \{(1, 2), (1, 3), (2, 4), (2, 5), (2, 6), (3, 4), (4, 5), (5, 6)\}$ with edge lengths 6, 5, 4, 4, 4, 3, 3, and 3, respectively, as shown in Fig. 4.2 to be solved by proposed DNA molecular algorithm.

As shown in Table 4.1, each vertex is randomly associated with a 20-mer DNA sequence, denoted $V_i$, which has a similar $Tm$ due to vertex sequences should contribute equally to the thermal stability of paths. Weight sequences are designed to
Figure 4.1: The KSRP facilities layout, available routes and their weights.

Figure 4.2: A 6-vertex, 8-edge, 4-weight CTP translated from KSRP layout plan.
differ from $T_m$ depending on its weight. The lighter in weight, the lower the melting temperature is. In this research, the $T_m$ is calculated by N-N model with the concentration of 1nM oligomer and 50mM salt.

The edge encoding scheme is identical to Adleman-Lipton model, which edge between vertices are generated by the second half of beginning vertex, weight sequence, and first half of ending vertex. For each edge $i \rightarrow j$ in the graph, an oligonucleotide $V_{i\rightarrow j}$ is created that is the 3’ 10-mer of $V_i$ followed by the weight sequence of path length, and then the 5’ 10-mer of $V_j$. The edge from vertex 1 to vertex 2, for example, is the 3’ 10-mer of $V_1$: ‘GTACTGTTGC’ followed by the weight sequence of 6: ‘GAGCGACCAGCGACACCCA’, then the 5’ 10-mer of $V_2$: ‘CACGATGTAC’, as illustrated in Fig. 4.3.

Table 4.1: DNA sequences associated with each vertex and weight for the 6-vertex, 8-edge, 4-weight cable trench problem. Each vertex is randomly associated with a 20-mer sequence of DNA. Weight sequences are designed to have different melting temperatures depending on its weight. The lighter in weight, the lower the melting temperature is. The melting temperature is correlated to the GC content.

<table>
<thead>
<tr>
<th>Vertex</th>
<th>DNA Sequence (5’ → 3’)</th>
<th>$T_m$ (°C)</th>
<th>GC Content (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_1$</td>
<td>TCCTCGTTAGGTACTGTTGC</td>
<td>46.02</td>
<td>50</td>
</tr>
<tr>
<td>$V_2$</td>
<td>CACGATGTACCTCTCAATGC</td>
<td>45.59</td>
<td>50</td>
</tr>
<tr>
<td>$V_3$</td>
<td>TTCATGTTGCTTGACGTGAG</td>
<td>47.59</td>
<td>50</td>
</tr>
<tr>
<td>$V_4$</td>
<td>GCGGTCTCAAATCCGTCAC</td>
<td>46.51</td>
<td>50</td>
</tr>
<tr>
<td>$V_5$</td>
<td>ATGGACCCAGATGCAAAGG</td>
<td>46.92</td>
<td>50</td>
</tr>
<tr>
<td>$V_6$</td>
<td>GTAGACCCTCGCCGTGCTAT</td>
<td>46.97</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weight</th>
<th>DNA Sequence (5’ → 3’)</th>
<th>$T_m$ (°C)</th>
<th>GC Content (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>ATGTTGGAGTTATTGCCAAC</td>
<td>43.18</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>GCGTAACGTACACGTTGCTGT</td>
<td>48.48</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>TGCTGGGTATGGGCTCACA</td>
<td>51.88</td>
<td>60</td>
</tr>
<tr>
<td>6</td>
<td>GAGCGACCAGCGACACCCA</td>
<td>55.84</td>
<td>70</td>
</tr>
</tbody>
</table>
Figure 4.3: Example of the encoding scheme. An oligonucleotide $V_{1\rightarrow 2}$ is created that is the 3′ 10-mer of $V_1$ followed by the weight sequence of path length, and then the 5′ 10-mer of $V_2$.

For each Watson-Crick complement of $V_i$ in a CTP graph, and for each edge $i \rightarrow j$ were mixed in a single ligation reaction. The added amount of an edge was varied according to weight, where as the weight increased, the amount was decreased. This bias was recommended by Lee et al. [60] in order to break through the current existed bottleneck of numerical represented DNA computing. The oligonucleotide mixture was heated to 95°C and cooled to 20°C at 2°C/min for hybridization. The reaction mixture was then subjected to a ligation.

To ensure the routes that begin and end at $V_1$, the DNA strands of $V_1$ and complement of $V_1$ were used as two primers in two successive PCRs. The products of PCR were then sieved by conventional gel electrophoresis for determining the size of the DNA strands. Only those DNA strands of the length that is at least twice of six-vertices long (round trip) was retained.

To verify the DNA strands pass by every vertex, the product from the above step is affinity-purified with a biotin-avidin magnetic beads system. Only those ssDNA
molecules that contained the sequence $V_i$ annealed to the bound were retained. This process was repeated six times with complement of six vertices to ensure every vertex has passed by.

In the beginning cycles of DTG-PCR, the denaturation temperature started at $70^\circ C$, which was lower than the $T_m$ of template strands, and then gradually increased until the denaturation temperature reached $95^\circ C$. The rest procedures were remained the same as conventional PCR. The products of DTG-PCR was analyzed in gel electrophoresis and the shortest band contained three DNA strands of the same length, as shown in Table 4.3, that cannot be separated by this conventional technique. With our thermodynamic control scheme, the advantage of TGGE was taken which the most economical route was detected by the distinct $T_m$ behaviors of DNA strands. Therefore, the most economical route can be distinguished from other possible routes.

The GC content and its $T_m$ among those possible routes are shown in Table 4.3, which the DNA strands corresponded to the route ‘1 $\rightarrow$ 3 $\rightarrow$ 4 $\rightarrow$ 2 $\rightarrow$ 4 $\rightarrow$ 5 $\rightarrow$ 6 $\rightarrow$ 5 $\rightarrow$ 4 $\rightarrow$ 3 $\rightarrow$ 1’ shown the most economical weight. The final sequencing results in finding the optimal solution of \{(1, 3), (2, 4), (3, 4), (4, 5), (5, 6)\} to our CTP. This answer is confirmed with the benchmark result from Vasko et al. [105].

4.6 Concluding Remarks

The fixed-length DNA computing with thermodynamic control algorithm was proposed to solve the CTP. With enormous parallelism of biochemical reaction, the proposed algorithm would show its advantage follow by the problem size increases.
Table 4.2: Result from DTG-PCR. *Three different DNA strands represent the possible routes of the same length which cannot be separated by the conventional gel electrophoresis.*

<table>
<thead>
<tr>
<th>DNA Sequence (5’ → 3’)</th>
<th>Oligo Length (Bases)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCCTCGTTAGGTACTTTGCGAGCCGACCACGGACACCCCA</td>
<td>420</td>
</tr>
<tr>
<td>CACGATGTACCTCTCAATGGCGTAACGTTACGGTTGTTG</td>
<td></td>
</tr>
<tr>
<td>GTTAAGCTCCTGGGCTATACGGTAACGTACCCTGTTG</td>
<td></td>
</tr>
<tr>
<td>CACGATGTACCTCTCAATGGCGTAACGTTACGGTTGTTG</td>
<td></td>
</tr>
<tr>
<td>ATGGACCCAGATGCAAGGCGGTAACGTACCGTTGTTG</td>
<td></td>
</tr>
<tr>
<td>CACGATGTACCTCTCAATGGCGTAACGTTACGGTTGTTG</td>
<td></td>
</tr>
<tr>
<td>TCCTCGTTAGGTACTTGTGGGCTGCTGGGGTATGGCCTCACA</td>
<td>420</td>
</tr>
<tr>
<td>TTCATGGTGCTGAGTGGAGTTGTGTTATGGCAAC</td>
<td></td>
</tr>
<tr>
<td>GCGGTCTCTAAATTCGTCACAGTGGTACGGTTGTTG</td>
<td></td>
</tr>
<tr>
<td>TTCATGGTGCTGAGTGGAGTTGTGTTATGGCAAC</td>
<td></td>
</tr>
<tr>
<td>GCGGTCTCTAAATTCGTCACAGTGGTACGGTTGTTG</td>
<td></td>
</tr>
<tr>
<td>TCCTCGTTAGGTACTTGTGGGCTGCTGGGGTATGGCCTCACA</td>
<td>420</td>
</tr>
<tr>
<td>TTCATGGTGCTGAGTGGAGTTGTGTTATGGCAAC</td>
<td></td>
</tr>
<tr>
<td>GCGGTCTCTAAATTCGTCACAGTGGTACGGTTGTTG</td>
<td></td>
</tr>
<tr>
<td>TTCATGGTGCTGAGTGGAGTTGTGTTATGGCAAC</td>
<td></td>
</tr>
<tr>
<td>GCGGTCTCTAAATTCGTCACAGTGGTACGGTTGTTG</td>
<td></td>
</tr>
<tr>
<td>TCCTCGTTAGGTACTTGTGGGCTGCTGGGGTATGGCCTCACA</td>
<td>420</td>
</tr>
<tr>
<td>TTCATGGTGCTGAGTGGAGTTGTGTTATGGCAAC</td>
<td></td>
</tr>
<tr>
<td>GCGGTCTCTAAATTCGTCACAGTGGTACGGTTGTTG</td>
<td></td>
</tr>
<tr>
<td>TTCATGGTGCTGAGTGGAGTTGTGTTATGGCAAC</td>
<td></td>
</tr>
<tr>
<td>GCGGTCTCTAAATTCGTCACAGTGGTACGGTTGTTG</td>
<td></td>
</tr>
<tr>
<td>TCCTCGTTAGGTACTTGTGGGCTGCTGGGGTATGGCCTCACA</td>
<td>420</td>
</tr>
<tr>
<td>TTCATGGTGCTGAGTGGAGTTGTGTTATGGCAAC</td>
<td></td>
</tr>
<tr>
<td>GCGGTCTCTAAATTCGTCACAGTGGTACGGTTGTTG</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.3: Possible routes resulting from DTG-PCR and their distinct behavior in melting temperature and GC content

<table>
<thead>
<tr>
<th>Route</th>
<th>$T_m$ (°C)</th>
<th>GC Content (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 → 2 → 6 → 2 → 5 → 2 → 1 → 3 → 4 → 3 → 1</td>
<td>82.15</td>
<td>52.10</td>
</tr>
<tr>
<td>1 → 2 → 6 → 2 → 1 → 3 → 4 → 5 → 4 → 3 → 1</td>
<td>82.09</td>
<td>51.50</td>
</tr>
<tr>
<td>1 → 3 → 4 → 2 → 4 → 5 → 6 → 5 → 4 → 3 → 1</td>
<td>80.57</td>
<td>47.89</td>
</tr>
</tbody>
</table>

The proposed algorithm for CTP is identical to the one that solved the elevator dispatching problem in Chapter 3, but differ from the strategy. Besides the limitation that came from the biochemical techniques as mentioned, another challenge is arisen, which DNA computing is problem-depended. Different solutions may have to be designed for targeting on different problems. This situation is similar to the ENIAC, which its programming is problem-depended by manipulating its switches and cables. The automation of DNA computing is needed to carry out, so that a problem-wide general solution can be investigated.
Chapter 5

Bio-inspired Fuzzy Forecasting: A New Forecasting Technique

“If you do not change direction, you may end up where you are heading.”
- Lao Tzu, Chinese taoist philosopher.

5.1 Overview

The bio-inspired forecasting process is applied for several reasons: First, a problem specific computing will be easier to design and implement with less need for model/functional complexity and flexibility. Secondly, the types of soft computational problems that bio-soft based computing may be able to effectively solve social competence for economic conditions that a dedicated processor would be naturally reasonable. As well, these problems will be likely to require extensive time that would preclude the need for a more versatile and interactive system that may be able to be implemented with a universal computing machine.

The main advantage offered by most proposed models of DNA based computation
is the ability to handle millions of operations in parallel. The massively parallel processing capabilities of DNA computers may give them the potential to find tractable solutions to otherwise intractable problems, as well as potentially speeding up large, but otherwise solvable, polynomial time problems requiring relatively few operations [1]. Many different models exist within the paradigm of “classical” DNA computation [17, 42, 49, 72] and [81], each of them with different advantages and degrees of applicability to classes of problems.

On the other hand, in the field of humanity and social science, fuzzy statistics and fuzzy forecasting have attracted many attentions lately. This is a natural result because of the complicated phenomenon of humanity and society which are hard to be fully explained by traditional models. Taking stock market as an example, the essence of closing price is uncertain and indistinctive. Moreover, there are many factors could influence the closing price, such as trading volume and exchange rate, etc. Therefore, if we merely consider the closing price of previous day to construct our forecasting model, we are capable of estimating the future trend, rather than unexpected or unnecessary loss.

In this chapter, we propose a fuzzy time series modeling process with DNA molecules inspired by natural selection. This method is applied to financial time series data by forecast their future trend, as well as comparing the forecasting performance with traditional model. From the empirical studies, it is shown that our proposed method demonstrates an appropriate performance on predicting the future exchange rates.
5.2 Fuzzy Time Series

Let $Y(t)$ ($t = \ldots, 0, 1, 2, \ldots$), a subset of $\mathbb{R}$, be the universe of discourse on which fuzzy sets $f_i(t)$ ($i = 1, 2, \ldots$) are defined and let $F(t)$ be a collection of $f_i(t)$. Then, $F(t)$ is called a fuzzy time series on $Y(t)$ ($t = \ldots, 0, 1, 2, \ldots$). Let $F(t)$ and $F(t - 1)$ be fuzzy time series on $Y(t)$ and $Y(t - 1)$ ($t = \ldots, 0, 1, 2, \ldots$). For any $f_j(t) \in F(t)$, there exists an $f_i(t - 1) \in F(t - 1)$ such that there is a first-order fuzzy relation $R(t, t - 1)$ and $f_j(t) = f_i(t - 1) \circ R_{ij}(t, t - 1)$, then $F(t)$ is said to be caused by $F(t - 1)$ only. Denote this as $f_i(t - 1) \rightarrow f_j(t)$ or equivalently $F(t - 1) \rightarrow F(t)$. Song and Chissom [99] derived the first-order model based on the first-order relation and extended to $l$th-order model.

In the above statement, $F(t)$ can be understood as a linguistic variable and $f_i(t)$ ($i = 1, 2, \ldots$) as the possible linguistic values of $F(t)$. Because at different times, the values of $F(t)$ can be different, $F(t)$ is a function of time $t$. Also, since the universes of discourse can be different at different times, thus $Y(t)$ is used for the universe at time $t$.

According to Song and Chissom [99], the forecasting procedures with fuzzy time series in general are as follows:

**Step 1.** Define the universes of discourse on which some fuzzy sets will be defined.

**Step 2.** Collect historical data (usually linguistic values).

**Step 3.** Define fuzzy sets on the universes of discourse using the historical linguistic data.

**Step 4.** Set up fuzzy relationships using historical data.
Step 5. Sum up all the relationships defined in Step 4. The summation will be the model.

Step 6. Apply the input to the model and calculate the output for forecast. The output will be the forecasted value.

Step 7. Defuzzify the output of the model (sometimes this step is not necessary).

5.3 Background

In statistics, the patterns of collected data can be numerical, qualitative formats, or linguistic values (such as data derived from testing). These kinds of data are hard to be analyzed by traditional time series models. Nevertheless, with fuzzy sets, the patterns of data will not be restricted and a more suitable model can be established. There is no certain rule for the optimal partition in building fuzzy range sets. Generally, the more partition we do, the more precision we have; however, the more complicated calculation is required. In short, the determination between accuracy and complexity is entirely up to the individual requirements.

Upon applying fuzzy logic in the time series analysis, the first step is to identify the way of integrating linguistic variable analysis methods in solving the autoregressive relation problem of dynamic data. Chiang et al. [25] presented self-learning methods to modify fuzzy models for dynamic system in linguistic field. Later, Huarng [43] proposed a fuzzy linguistic summary as one of the data mining function to discover useful knowledge from database. In fact, fuzzy relation equations are easier to be understood and applied than decision tables or decision rules.
In view of this, many researchers have adopted fuzzy relation equations for time series analysis and forecasting. For instance, Wu and Hung [114] proposed a fuzzy identification procedure for ARCH and Bilinear models. Kumar and Wu [58] used fuzzy statistical techniques in change period’s detection of nonlinear time series. Chen and Hwang [23] proposed the two-factors time-variant fuzzy time series model and developed two algorithms for temperature prediction. Huarng [43] and Tseng et al. [101] proposed heuristic models by integrating problem-specific heuristic knowledge with Chen’s [23] model to improve forecasting.

Despite being published before the discovery of DNA, Darwin’s 1859 classic remains a robust description of evolution by natural selection. Inspired by this, the humanity and social science sustained growth stimulated natural selection that shaped the evolution of the biology. The mechanism of evolutionary change is adopted in this pioneering research, which the DNA molecule is used as a medium in forecasting. The process of PCR reflects the natural selection in our society. The strand of DNA templates that generated from historical data can be searched for best represent the future trend.

5.4 Modeling Time Series and Forecasting with DNA

In this study, we will make a conventional estimation as well as a fuzzy estimation (with membership function) of the coefficients for the investigated time series. The ARIMA model is one of the most adopted forecasting techniques for the time series data, which the key process is to find the order for the candidate models
ARIMA\((p, d, q)\). After the order decided, the appropriate coefficients are then estimated for obtaining the lowest value of Akaike information criteria (AIC). The final step is to forecast from the constructed model. For the fuzzy autoregressive process of ordered one, the \(l\)-steps prediction becomes \(X_n(l) = E(X_{n+l} \mid X_n, \ldots, X_1)\). In order to evaluate the forecasting performance, our result will be compared with a traditional method.

**Modeling Fuzzy Time Series**

To estimate the appropriate coefficients for further processing with DNA is demonstrated following:

Let \(\{X_t; t = 1, 2, \ldots, n\}\) be a time series which has followed by an AR(1) or Markov process; \(\{\Delta X_t = X_t - X_{t-1}; t = 2, 3, \ldots, n\}\) be the first order difference of the time series \(\{X_t\}\); \(T = \max |\Delta X_t|\) be the maximum first order difference \(\Delta X_t\).

We assign the variation of \(\Delta X_t\) into five period with respect to linguistic variables, say \(\text{plunge}=-1.0, \text{drop}=-0.5, \text{draw}=0, \text{soar}=0.5, \text{surge}=1.0\), where each linguistic variable denotes an element in fuzzy set and its corresponding membership. The memberships of fuzzy time series transformed by \(k_t = \frac{\Delta X_t}{T}\) are based on Table 5.1.

**Encoding Scheme**

Next will be a series of biochemical process. Assume we have time series data with \(n\) records, which exists \(n-1\) first order difference. A number of \(n-1\) test tubes are set to represent each first order difference \(\Delta X_t\). Two DNA sequences are designed to represent the “up” trend and the “down” trend, denote as DNA1 and DNA2, respectively. These two DNA sequences are identical besides its own special designed
Table 5.1: Memberships of fuzzy time series $k_t = \frac{\Delta X_t}{T}$ w.r.t linguistic variables

<table>
<thead>
<tr>
<th>Membership</th>
<th>$Membership$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_t \leq -0.5$</td>
<td>$-2(k_t + 0.5)$</td>
</tr>
<tr>
<td>$-0.5 &lt; k_t \leq 0$</td>
<td>$0$</td>
</tr>
<tr>
<td>$0 &lt; k_t \leq 0.5$</td>
<td>$0$</td>
</tr>
<tr>
<td>$0.5 &lt; k_t$</td>
<td>$0$</td>
</tr>
</tbody>
</table>

Figure 5.1: DNA encoding scheme. DNA1 and DNA2 are designed in represent “up” trend and “down” trend respectively. DNA1 and DNA2 have its own special design primers.

PCR primers denote as $UF$ (DNA1 forward primer), $UR$ (DNA1 reverse primer), $DF$ (DNA2 forward primer), and $DR$ (DNA2 reverse primer) as shown in Fig. 5.1.

Synthesizing short ssDNA is a routine process in current biological technology. The molecules can be made by an auto-programming machine called DNA synthesizer. Itineraries can then be produced from the encodings by linking them together in proper order. To accomplish this, we can take advantage of the fact that DNA hybridizes with its complementary sequence. Random itineraries can be made by mixing difference encodings.

The ratio of up trend primers ($UF, UR$) and down trend primers ($DF, DR$) for
Table 5.2: The ratio of primer for each test tube

<table>
<thead>
<tr>
<th>$\frac{\Delta X_t}{T}$</th>
<th>$[−1.0, −0.9]$</th>
<th>$[−0.9, −0.6]$</th>
<th>$[−0.6, −0.3]$</th>
<th>$[−0.3, −0.1]$</th>
<th>$[0.1, 0.3]$</th>
<th>$[0.3, 0.6]$</th>
<th>$[0.6, 0.9]$</th>
<th>$[0.9, 1.0]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{up} : P_{down}$</td>
<td>$9 : 1$</td>
<td>$8 : 2$</td>
<td>$7 : 3$</td>
<td>$6 : 4$</td>
<td>$5 : 5$</td>
<td>$4 : 6$</td>
<td>$3 : 7$</td>
<td>$2 : 8$</td>
</tr>
</tbody>
</table>

Figure 5.2: Each test tube is set up to contain the template of $DNA_1$ and $DNA_2$ accompanied with their own specific PCR primers. Besides, the up trend primers ($i.e., UF, UR$) and down trend primers ($i.e., DF, DR$) are inputted in ratio according to the $\Delta X_t$.

Polymerase Chain Reaction

As shown in Fig. 5.2, each test tube is set up to contain the template of $DNA_1$ and $DNA_2$ accompanied with their own specific PCR primers. Besides, the up trend primers ($i.e., UF, UR$) and down trend primers ($i.e., DF, DR$) are inputted in ratio according to Table 5.2 that represent the specific first order difference $\Delta X_t$. We expect to collect the amplified numbers of $DNA_1$ and $DNA_2$ after DNA amplification.
as illustrated in Fig. 5.2. To accomplish this, the PCR technique is used, which allows us to reproduce many copies of a specific sequence of DNA. PCR is an iterative process that cycles through a series of copying events using an enzyme called DNA polymerase. Polymerase will copy a section of ssDNA starting at the position of a primer, a short piece of DNA complimentary to one end of a section of the DNA that we are interested in. By selecting primers that flank the section of DNA we want to amplify, the polymerase preferentially amplifies the DNA template through specific set of primers, doubling the amount of DNA containing this sequence in each cycle of PCR reaction. After many iterations of PCR (we took 25 cycles in this research), the DNA we are working on is amplified exponentially. The free process of amplification between DNA1 and DNA2 represents the natural selection, which the superior amplifies more and the inferior amplifies less. After PCR, we end up with a specific sequence of test tube full of dsDNA of two different sequences, DNA1 and DNA2.

**DNA Quantification**

The test tube is now filled with DNA encoded itineraries. What interested us is the quantity of DNA1 and DNA2 in each test tube. Many molecular biology techniques can be utilized to estimate PCR-produced DNA populations, such as real-time PCR (quantitative PCR, or Q-PCR), gel electrophoresis, and fluorescence assay, etc. The detection and quantification of DNA amplification is very important both in research and in a clinical diagnostic setting. Real-time PCR, which is used in this research, has become a well-established procedure for quantifying levels of gene expression. Its power resides in the ability to detect, at every cycle of the PCR, the amount of PCR product (amplicon) using fluorescence.
Making Prediction

In the final stage, according to the quantity of DNA1 and DNA2 retrieved from the previous step, we define the ratio of up-and-down trend as the auto-correlation coefficient by

\[ U_n = \frac{\sum_{i=2}^{n} DNA1_i}{\sum_{i=2}^{n} DNA1_i + \sum_{i=2}^{n} DNA2_i} \] (5.4.1)

\[ D_n = \frac{\sum_{i=2}^{n} DNA2_i}{\sum_{i=2}^{n} DNA1_i + \sum_{i=2}^{n} DNA2_i} \] (5.4.2)

The first order difference can be represented as

\[ \Delta X_n = (U_n - D_n)T \] (5.4.3)

The one step forecasting value can be derived

\[ X_{n+1} = X_n + (U_n - D_n)T \] (5.4.4)

The forecasting with \( l \)-steps (\( l \geq 2 \)) can be expressed

\[ X_{n+1} = X_n + (U_{n+l-1} - D_{n+l-1})T \] (5.4.5)

where

\[ U_{n+l-1} = \frac{\sum_{i=2}^{n} DNA1_i + \sum_{i=n+1}^{n+l-1} DNA1_i}{\sum_{i=2}^{n} DNA1_i + \sum_{i=2}^{n} DNA2_i + \sum_{i=n+1}^{n+l-1} DNA1_i + \sum_{i=n+1}^{n+l-1} DNA2_i} \] (5.4.6)

\[ D_{n+l-1} = \frac{\sum_{i=2}^{n} DNA2_i + \sum_{i=n+1}^{n+l-1} DNA2_i}{\sum_{i=2}^{n} DNA1_i + \sum_{i=2}^{n} DNA2_i + \sum_{i=n+1}^{n+l-1} DNA1_i + \sum_{i=n+1}^{n+l-1} DNA2_i} \] (5.4.7)

In the fuzzy forecasting for one step, \((U - D)T\) is transferred into fuzzy number according to Table 5.1. Hence, the one step fuzzy forecasting becomes

\[ FX_{n+1} = X_n \oplus (\frac{m_i}{L_i} + \frac{m_{i+1}}{L_{i+1}})T \] (5.4.8)
where \((\frac{m_i}{L_i} + \frac{m_{i+1}}{L_{i+1}})T\) means the memberships of the variation with respect to the linguistic variables \(L_i\) and \(L_{i+1}\) based on the total variation \(T\), and \(\oplus\) stands for fuzzy addition.

The proposed bio-inspired forecasting algorithm is summarized as follows:

**Step 1.** For time series \(\{X_t\}\), decide the total variation of \(\{\Delta X_t\}\) and linguistic variables \(\{L_1, L_2, \ldots, L_5\}\) of \(\Delta X_t\).

**Step 2.** Design the memberships of fuzzy time series w.r.t the linguistic variables.

**Step 3.** Determine the ratio of primer.

**Step 4.** Design the DNA sequences and primers.

**Step 5.** Molecular reaction [PCR].

**Step 6.** DNA quantification [Real-time PCR].

**Step 7.** Calculate the forecasting value.

### 5.5 Mean Absolute Forecasting Accuracy: The Performance Evaluation

The Mean Absolute Forecasting Accuracy (MAFA) is defined for evaluating the forecasting performance in this research. To compare the performance between different forecasting methods, each linguistic variable needs to be assigned with an ordered rank. In this study, for instance, a **plunge** as \(-1.0\), **drop** as \(-0.5\), **draw** as \(0\), **soar** as \(0.5\) and **surge** as \(1.0\). By doing so, MAFA can be performed.
Definition 5.5.1. Suppose \( \{ RL_t, t = 1, \ldots, n \} \) and \( \{ FL_t, t = 1, \ldots, n \} \) denote the real and outputting linguistic variables respectively. The mean absolute forecasting accuracy can be defined as

\[
MAFA = 1 - \frac{\sum_{t=1}^{n} |FL_t - RL_t|}{n - 1}
\]

where \( r \) denotes the number of linguistic variables.

Example 5.5.1. Suppose that real linguistic variables of the time series are \{drop, draw, drop, surge, soar, drop, surge, drop, draw, plunge\}, then the corresponding values of linguistic variables are \{-0.5, 0, -0.5, 1.0, 0.5, -0.5, 1.0, -0.5, 0, -1.0\}. The outputting linguistic variables are \{drop, draw, plunge, surge, draw, draw, surge, drop, surge, soar\}, then the corresponding values of linguistic variables are \{-0.5, 0, -1.0, 1.0, 0, 0, 1.0, -0.5, 1.0, 0.5\}. By Definition 5.5.1, the MAFA can be obtained.

\[
MAFA = 1 - \frac{\sum_{t=1}^{10} |FL_t - RL_t|}{10 - 1} = 1 - \frac{1}{10} = 0.90
\]

5.6 Empirical Application

Forecasting the monthly exchange rates of JPY/USD and EUR/USD are taken as an example in this study.

Data Analysis

The historical exchange rates between January 2004 and December 2005 as shown in Fig. 5.3 were gathered from the Federal Reserve Bank of New York. According to the data retrieved, \( T = max|\Delta X_t| = 5.27 \) for JPY/USD, and \( T = max|\Delta X_t| = 0.034 \)
Fuzzy Model Construction

After fuzzified these data of monthly exchange rate, the method mentioned in Section 5.1 is applied for calculating the data’s corresponding memberships in linguistic variables $L_i$ as illustrated in Tables 5.3 and 5.4.

Forecasting and Performance

The memberships of JPY/USD and EUR/USD monthly variations between January 2004 and December 2005 can be observed from Tables 5.3 and 5.4. Finally, the result from DNA forecasting is shown at Tables 5.5 and 5.6. Due to the purpose of this research is to explore the qualitative trend of time series, the memberships were transformed based on fuzzy rule from the concentration of DNA as illustrated at Table 5.7, which was generated from Table 5.1. The corresponding linguistic variables were obtained to facilitate analysis, and the result was compared with the best
Table 5.3: The ratios of primer and memberships for JPY/USD

<table>
<thead>
<tr>
<th>Month</th>
<th>$\frac{\Delta N_t}{T}$</th>
<th>$P_{up} : P_{down}$</th>
<th>plunge</th>
<th>drop</th>
<th>draw</th>
<th>soar</th>
<th>surge</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004/1-2</td>
<td>0</td>
<td>5:5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/2-3</td>
<td>0.39</td>
<td>3:7</td>
<td>0</td>
<td>0</td>
<td>0.22</td>
<td>0.78</td>
<td>0</td>
</tr>
<tr>
<td>2004/3-4</td>
<td>-0.26</td>
<td>6:4</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/4-5</td>
<td>1.00</td>
<td>1:9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2004/5-6</td>
<td>-0.58</td>
<td>7:3</td>
<td>0.16</td>
<td>0.84</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/6-7</td>
<td>-0.02</td>
<td>5:5</td>
<td>0</td>
<td>0.04</td>
<td>0.96</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2005/9-10</td>
<td>0.71</td>
<td>2:8</td>
<td>0</td>
<td>0</td>
<td>0.59</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>2005/10-11</td>
<td>0.68</td>
<td>2:8</td>
<td>0</td>
<td>0</td>
<td>0.64</td>
<td>0.36</td>
<td></td>
</tr>
<tr>
<td>2005/11-12</td>
<td>0</td>
<td>5:5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.4: The ratios of primer and memberships for EUR/USD

<table>
<thead>
<tr>
<th>Month</th>
<th>$\frac{\Delta N_t}{T}$</th>
<th>$P_{up} : P_{down}$</th>
<th>plunge</th>
<th>drop</th>
<th>draw</th>
<th>soar</th>
<th>surge</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004/1-2</td>
<td>-0.06</td>
<td>5:5</td>
<td>0</td>
<td>0.12</td>
<td>0.88</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/2-3</td>
<td>0.73</td>
<td>2:8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.48</td>
<td>0.52</td>
</tr>
<tr>
<td>2004/3-4</td>
<td>0.54</td>
<td>3:7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.92</td>
<td>0.08</td>
</tr>
<tr>
<td>2004/4-5</td>
<td>-0.01</td>
<td>5:5</td>
<td>0</td>
<td>0.02</td>
<td>0.98</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/5-6</td>
<td>-0.28</td>
<td>6:4</td>
<td>0</td>
<td>0.56</td>
<td>0.44</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/6-7</td>
<td>-0.25</td>
<td>6:4</td>
<td>0</td>
<td>0.50</td>
<td>0.50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2005/9-10</td>
<td>0.48</td>
<td>3:7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.96</td>
<td>0.04</td>
</tr>
<tr>
<td>2005/10-11</td>
<td>0.47</td>
<td>3:7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.94</td>
<td>0.06</td>
</tr>
<tr>
<td>2005/11-12</td>
<td>-0.16</td>
<td>5:5</td>
<td>0</td>
<td>0.32</td>
<td>0.68</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 5.5: The ratios of two DNAs and memberships for JPY/USD after reaction

<table>
<thead>
<tr>
<th>Month</th>
<th>DNA1:DNA2</th>
<th>plunge</th>
<th>drop</th>
<th>draw</th>
<th>soar</th>
<th>surge</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004/1-2</td>
<td>52:48</td>
<td>0</td>
<td>0.08</td>
<td>0.92</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/2-3</td>
<td>25:75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2004/3-4</td>
<td>69:31</td>
<td>0</td>
<td>0.76</td>
<td>0.24</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/4-5</td>
<td>6:94</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.24</td>
<td>0.76</td>
</tr>
<tr>
<td>2004/5-6</td>
<td>58:42</td>
<td>0</td>
<td>0.32</td>
<td>0.68</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/6-7</td>
<td>54:46</td>
<td>0</td>
<td>0.16</td>
<td>0.84</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2005/9-10</td>
<td>52:48</td>
<td>0</td>
<td>0.08</td>
<td>0.92</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Average</td>
<td>41:59</td>
<td>0</td>
<td>0</td>
<td>0.36</td>
<td>0.64</td>
<td>0</td>
</tr>
<tr>
<td>2005/11</td>
<td>38:62</td>
<td>0</td>
<td>0</td>
<td>0.52</td>
<td>0.48</td>
<td>0</td>
</tr>
<tr>
<td>2005/12</td>
<td>45:55</td>
<td>0</td>
<td>0</td>
<td>0.20</td>
<td>0.80</td>
<td>0</td>
</tr>
<tr>
<td>2006/1</td>
<td>47:53</td>
<td>0</td>
<td>0</td>
<td>0.88</td>
<td>0.12</td>
<td>0</td>
</tr>
<tr>
<td>2006/2</td>
<td>42:58</td>
<td>0</td>
<td>0</td>
<td>0.68</td>
<td>0.32</td>
<td>0</td>
</tr>
</tbody>
</table>

ARIMA model (1, 0, 0) in Tables 5.8 and 5.9.

The MAFA defined in Definition 5.5.1 is applied for measuring the accuracy of forecasting methods. From Tables 5.8 and 5.9, we found the DNA model shown better performance comparing with the Best ARIMA model. The result of MAFA is shown at Table 5.10.

5.7 Concluding Remarks

A novel forecasting technique is proposed in this paper. The bio-inspired forecasting technique with DNA is so exciting because of the collaboration of chemists, biologists, mathematicians, and computer scientists to understand and simulate fundamental biological processes and algorithms taking place in molecules. Although bio-inspired forecasting might not replace conventional techniques in the near future, they still
Table 5.6: The ratios of two DNAs and memberships for EUR/USD after reaction

<table>
<thead>
<tr>
<th>Month</th>
<th>DNA1:DNA2</th>
<th>Plunge</th>
<th>Drop</th>
<th>Draw</th>
<th>Soar</th>
<th>Surge</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004/1-2</td>
<td>46:54</td>
<td>0</td>
<td>0</td>
<td>0.16</td>
<td>0.94</td>
<td>0</td>
</tr>
<tr>
<td>2004/2-3</td>
<td>13:87</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.52</td>
<td>0.48</td>
</tr>
<tr>
<td>2004/3-4</td>
<td>23:77</td>
<td>0</td>
<td>0</td>
<td>0.04</td>
<td>0.96</td>
<td>0</td>
</tr>
<tr>
<td>2004/4-5</td>
<td>51:49</td>
<td>0</td>
<td>0.04</td>
<td>0.96</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/5-6</td>
<td>70:30</td>
<td>0</td>
<td>0.80</td>
<td>0.20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2004/6-7</td>
<td>68:32</td>
<td>0</td>
<td>0.72</td>
<td>0.28</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Average</td>
<td>52:48</td>
<td>0</td>
<td>0.08</td>
<td>0.92</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2005/9-10</td>
<td>41:59</td>
<td>0</td>
<td>0</td>
<td>0.36</td>
<td>0.64</td>
<td>0</td>
</tr>
<tr>
<td>2005/11</td>
<td>50:50</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2005/12</td>
<td>45:55</td>
<td>0</td>
<td>0</td>
<td>0.20</td>
<td>0.80</td>
<td>0</td>
</tr>
<tr>
<td>2006/1</td>
<td>62:38</td>
<td>0</td>
<td>0.48</td>
<td>0.52</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2006/2</td>
<td>58:42</td>
<td>0</td>
<td>0.32</td>
<td>0.68</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.7: Memberships of DNA concentration $C = \frac{50 - DNA1\%}{50}$ w.r.t linguistic variables

<table>
<thead>
<tr>
<th>Membership</th>
<th>Plunge=-1</th>
<th>Drop=-0.5</th>
<th>Draw=0</th>
<th>Soar=0.5</th>
<th>Surge=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C \leq -0.5$</td>
<td>$-2(C + 0.5)$</td>
<td>$2(C + 1)$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$-0.5 &lt; C \leq 0$</td>
<td>0</td>
<td>$-2C$</td>
<td>$2(C + 0.5)$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$0 &lt; C \leq 0.5$</td>
<td>0</td>
<td>0</td>
<td>$-2(C - 0.5)$</td>
<td>$2C$</td>
<td>0</td>
</tr>
<tr>
<td>$0.5 &lt; C$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$-2(C - 1)$</td>
<td>$2(C - 0.5)$</td>
</tr>
</tbody>
</table>
Table 5.8: Forecasting performance comparison of JPY/USD

<table>
<thead>
<tr>
<th>Time</th>
<th>Real Value</th>
<th>Best ARIMA (1, 0, 0)</th>
<th>DNA Computing</th>
<th>Fuzzy DNA Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005/11</td>
<td>118.41</td>
<td>113.67</td>
<td>115.77</td>
<td>114.82 $\oplus$ (0.30 + 0.64) T + 0.64 ) 5.27</td>
</tr>
<tr>
<td>2005/12</td>
<td>118.43</td>
<td>112.75</td>
<td>117.14</td>
<td>115.77 $\oplus$ (0.30 + 0.64) T + 0.64 ) 5.27</td>
</tr>
<tr>
<td>2006/1</td>
<td>115.25</td>
<td>112.01</td>
<td>117.71</td>
<td>117.14 $\oplus$ (0.30 + 0.64) T + 0.64 ) 5.27</td>
</tr>
<tr>
<td>2006/2</td>
<td>111.41</td>
<td>118.05</td>
<td>118.05</td>
<td>117.71 $\oplus$ (0.30 + 0.64) T + 0.64 ) 5.27</td>
</tr>
<tr>
<td>2006/3</td>
<td>110.93</td>
<td>118.96</td>
<td>118.96</td>
<td>118.05 $\oplus$ (0.30 + 0.64) T + 0.64 ) 5.27</td>
</tr>
</tbody>
</table>

1 $X_{t+1} = 21.45 + 0.8X_t + \varepsilon_t$
2 $X_{t+1} = X_t + (U - D)T$
3 $FX_{n+1} = X_n \oplus (\frac{m_t}{L_t + \frac{m_{t+1}}{L_{t+1}}}) T$
Table 5.9: Forecasting performance comparison of EUR/USD

<table>
<thead>
<tr>
<th>Time</th>
<th>Real Value</th>
<th>Best ARIMA (1, 0, 0)</th>
<th>DNA Computing</th>
<th>Fuzzy DNA Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005/11</td>
<td>0.849</td>
<td>0.835</td>
<td>0.831</td>
<td>0.832 + (0.08 \delta_{\text{draw}} + 0.92 \delta_{\text{soar}})0.034</td>
</tr>
<tr>
<td>2005/12</td>
<td>0.843</td>
<td>0.837</td>
<td>0.831</td>
<td>0.831 + (0.48 \delta_{\text{draw}} + 0.52 \delta_{\text{soar}})0.034</td>
</tr>
<tr>
<td>2006/1</td>
<td>0.825</td>
<td>0.838</td>
<td>0.835</td>
<td>0.835 + (0.48 \delta_{\text{draw}} + 0.52 \delta_{\text{soar}})0.034</td>
</tr>
<tr>
<td>2006/2</td>
<td>0.839</td>
<td></td>
<td>0.827</td>
<td>0.827 + (0.48 \delta_{\text{draw}} + 0.52 \delta_{\text{soar}})0.034</td>
</tr>
<tr>
<td>2006/3</td>
<td>0.840</td>
<td></td>
<td>0.822</td>
<td>0.822 + (0.48 \delta_{\text{draw}} + 0.52 \delta_{\text{soar}})0.034</td>
</tr>
</tbody>
</table>

4 \( X_{t+1} = 0.203 + 0.759X_t + \varepsilon_t \)
5 \( X_{t+1} = X_t + (U - D)T \)
6 \( FX_{n+1} = X_n \oplus (\frac{m_t}{L_t} + \frac{m_{t+1}}{L_{t+1}})T \)
Table 5.10: Forecasting accuracy comparison between Best ARIMA and DNA computing

<table>
<thead>
<tr>
<th>Exchange Currency</th>
<th>Best ARIMA 1,0,1</th>
<th>DNA computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPY/USD</td>
<td>81.25%</td>
<td>81.25%</td>
</tr>
<tr>
<td>EUR/USD</td>
<td>75%</td>
<td>81.25%</td>
</tr>
</tbody>
</table>

have endless potentials for other applications. The bio-inspired forecasting technique has clear advantages over conventional techniques when searched DNA templates best represent the historical trend; it would literally resulting high forecasting accuracy theoretically.

In this research, we tried to make an appropriate process of constructing fuzzy time series model and use this model to forecast the exchange rate of JPY/USD and EUR/USD. Compare the DNA forecasting and DNA fuzzy forecasting model with traditional ARIMA model by the performance of forecasting accuracy, we can find that the bio-inspired model has better forecasting performance than that of traditional ARIMA model. We hope this method will provide a new forecasting technique for investors to make optimal decision with fuzzy information.

This experiment solved a forecasting problem, but there are two major shortcomings preventing a large scaling up of this computation. The complexity of the forecasting problem simply does not disappear when applying a different method of solution: it still increases exponentially. For our method, what scales exponentially is not the computing time, but rather the amount of DNA molecules. Unfortunately, this places some hard restrictions on the number of variables that can be solved. Another factor that places limits on this method is the nonlinear trend for each operation. Since these operations are not deterministic but stochastically driven (from
biochemistry), each step contains statistical errors, limiting the number of iterations one can do successively before the probability of producing an error becomes greater than producing the correct result.

In spite of the existed limitations, there are some progresses for further studies. For example:

1. To make a general rule for fuzzy order identification instead of the Markov relation. Bio-inspired forecasting with DNA molecules can be more accurately described as a collection of new computing paradigms rather than a single focus. Each of these different paradigms within molecular computing can be associated with different potential applications that may prove to place them at an advantage over conventional methods. Many of these models share certain features that lend them to categorization by these potential advantages. However, there exists enough similarities and congruencies that hybrid models will be possible, and that advances made in both “classic” and “natural” areas.

2. To extend our result to the multivariate fuzzy time series case. In fact, how to solve the nonstationary or seasonal factors for the time series are still the open questions.

3. In this research, we adopt five-ranking classification and transform the time series data into fuzzy numbers through membership functions. However, seven-ranking classification used in social sciences may be used in future studies for special situation, and it is yet to prove whether it will provide significant improvement on forecasting accuracy?

4. Future applications might make use of the error rates and instability of DNA
based computation methods as a means of simulating and predicting the emergent behavior of complex systems. This could pertain to weather forecasting, economics, and lead to more a scientific analysis of social science and humanities. Such a system might rely on inducing increased error rates and mutation through exposure to radiation and deliberately inefficient encoding schemes. Similarly, methods of DNA computing might serve as the most obvious medium for use of evolutionary programming for applications in design or expert systems. DNA might also serve as a medium to implement a true fuzzy logic system.
Chapter 6
Conclusion

“Adopt the pace of nature: her secret is patience.”
- Ralph Waldo Emerson, American essayist and poet.

This dissertation focuses on the developments of bio-soft computing methodologies in solving the engineering applications. In the first research, fixed-length DNA computing algorithm was proposed for group control optimization in the example of elevator dispatching problem. The DNA strands that represent floors and costs were in constant length. Varied costs that were differ from the DNA melting temperature. The strategy of linking DNA strands of different elevator was considered. An example of 6-story 2-elevator dispatching problem solved by the proposed algorithm was demonstrated. The thermodynamic property of DNA was used to find the local optimization. Yet the automation of operation is needed to solve the problem efficiently.

The same methodology from the first study was adopted in the second research but differ from solving strategy. The cable trench problem was solved by bio-soft computing with fixed length DNA. Due to the nature of the problem, the optimal
solution was searched round-trip based. This strategy also provided a more accurate result that benefited from the temperature controlled bio-techniques. An example of campus cable underlying project which is a 6-vertex, 8-edge, 4-weight cable trench problem was efficiently solved by the proposed algorithm and strategy. However, we have also shown that DNA computing is problem-depended.

In the third research, bio-inspired forecasting technique was developed. The new forecasting technique combines the mathematical, computational and biological sciences, and is capable of nonlinear time series forecasting. The currency exchange rates forecasting of JPY/USD and EUR/USD was demonstrated in the empirical study. A forecasting performance evaluation method, i.e., mean absolute forecasting accuracy, was also defined to rank the performance of linguistic forecasting. The proposed new technique has shown the better performance than the Best ARIMA model.

The researches contained in this dissertation have shown the application of bio-soft computing with DNA toward the engineering management problems is achievable. Not only my work extends previously reported research on the fixed-length DNA computing model to make it more practical, but a new forecasting technique through DNA molecules have also been proposed. DNA algorithms have been developed for group control optimization, cable trench problem, and forecasting, and applied to the real-world applications. The applications demonstrated in this dissertation include elevator dispatching, cable underlying, and currency exchange rates forecasting.

As limitations arose, the next bet is to automate the processes. DNA-based self-assemble appears to be a robust and readily programmable phenomenon, even as challenges emerge. Most important among them are the uncertain of the computational results that caused by DNA biochemistry reaction, and the exponential increase
in number of DNA molecules necessary to solve problems of interesting size. Despite these issues, presuming that algorithmic self-assembly of DNA can be made more reliable, it becomes important to understand the logical structure of self-assembly programs, and its relations with existing models of computation. In addition, new paradigms based on molecular evolution have emerged from molecular biology to inspire new directions in bio-soft computing. The programmable, algorithmic biochemical systems are our best bet to the next computational era.

Major directions for future researches beyond this dissertation include:

1. **DNA Computing Implementing Genetic Algorithm (GA):**
   As DNA computing and GA share similar properties, it motivates me to simulate GA by DNA computing. The evolutionary computation implemented by genetic materials *in vitro* is considered. Among all evolutionary computation techniques, GA has the most potential that may be implemented by DNA as it generally manipulating populations of bitstrings using both crossover and point-wise mutation. Besides the similarity of their features, the advantages of DNA computing can be taken: highly parallelism, massive information storage, and most importantly, the operational errors in DNA computing become naturally tolerable in GA then other deterministic algorithms.

2. **DNA Computing for Bioinformatics:**
   DNA computing approaching to bioinformatics is considered as it is naturally reasonable and offering quick solutions to difficult combinatorial problems. For example, combinatorial interaction of transcription factors (TFs) is important for gene regulation. The interest in computational discovery of regulatory elements is growing. There is also a growing awareness that regulatory elements
work in combination, and that this combinatorial behavior must be modeled for successful motif discovery. Targeting biological problems in a biological way is what motivates me to investigate on DNA computing based robust methodologies, and may provide an alternative that may efficiently solve such existing difficult bioinformatics problems.

The DNA computing has great promise in its ability to process enormous quantities of information efficiently. Since Adleman’s pioneering achievement, vast studies have been made in better understanding the capabilities of DNA computing systems and in overcoming technical difficulties. The term “DNA computing” is a bit narrow to the entire concept of information processing. Future bio-soft computing will not limited to DNA but generally to the entire biomolecular such as RNA and proteins \textit{in vitro} and \textit{in vivo}. Even though a general-purpose DNA computer capable of outperforming a conventional electronic computer is unlikely happened in the near future, the wide variety of potential for bio-soft computing systems make further researches desirable. We may take advantage of the biomolecular properties and target on the applications that best fit, for example, bioinformatics and biomedical applications.
Bibliography


[34] A. Fujino, T. Tobita, K. Segawa, K. Yoneda, and A. Togawa, *An elevator group control system with floor-attribute control method ad system optimization using*


Appendix A

MST with Minimum Path Length is NP-complete: A Proof

An informal proof of finding the MST that minimizes the path length between a particular set of vertices \( s \) and \( t \) is NP-complete by a reduction from 3SAT [32].

Outline of the reduction

A graph with two edge lengths is considered. The short edges will be formed into chains of many edges, so that the desired shortest MST path will be formed only of long edges. Say that a path is “good” if it is formed of long edges and is contained in some MST. The reduction will produce a problem in which the shortest MST path is good iff the original graph is 3-colorable.

Proof with a special “crossover gadget”

In Fig. A.1, the vertical and horizontal edges are long and the others represent chains of short edges. The only good paths are the ones going horizontally or vertically. It is impossible for a good path to turn or to use both the horizontal and vertical parts.

Now, given a planar graph for which we want to test 3-coloring, a different planar graph will be formed for which we want to test the existence of a good path, in the
Figure A.1: The original graph for proving MST with minimum path length is NP-complete.

following way. Assume the graph is embedded. First, find a set of arcs forming a tour of all the vertices, with each arc being strictly interior to a face (i.e. no arc can cross a vertex of the original graph). Assume the tour visits each vertex once, but the repeated vertices can be handled as described later.

Replace each vertex of the planar graph with a gadget as Fig. A.2, where the very left and right portions are incoming and outgoing arcs of the tour; the middle portion represents a set of three different paths, one labeled with each possible color the vertex may have. These three paths may wiggle around and cross over each other using the crossover gadget as the expanded view illustrated in Fig. A.3.

So (since the tour itself avoids the original graph’s edges) there is enough room to wiggle these colored paths along the edges of the original graph until the green path passes near the green path of each neighboring vertex (at which point we attach them by a chain of short edges so that only one of the two paths can be taken), the red
path passes near the red path of each neighboring vertex, etc. We should be careful doing this so that no path crosses itself, and no path crosses a differently colored path of a neighboring vertex, but this is not difficult.

If the tour passes through a vertex \( k \) times, just use \( k \) of these gadgets, where each one is responsible for handling some of the interactions with neighboring vertices, and with similar interactions among these \( k \) gadgets that force each one to use the same color path (e.g., the green path of one gadget interacts with any red and blue paths of other gadgets - it is not hard to do this without needing to cross over any other green paths).
List of Related Publications

**Refereed International Journals**


**Refereed International Conferences**


**Refereed Domestic Conferences**


**Book Chapter**