ADAPTING ARTIFICIAL IMMUNE ALGORITHMS FOR UNIVERSITY TIMETABLING

MUHAMMAD ROZI BIN MALIM

UNIVERSITI SAINS MALAYSIA

2009

ADAPTING ARTIFICIAL IMMUNE ALGORITHMS FOR UNIVERSITY TIMETABLING

by

MUHAMMAD ROZI BIN MALIM

Thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

JUNE 2009

ACKNOWLEDGEMENTS

I am very grateful and thankful to Allah S.W.T. for giving me the strength and chances to complete my research study. This thesis was made possible with the help and support of many people. Although I am solely responsible for the study and its findings, there are many external contributions which I must acknowledge.

I am much indebted to my academic supervisors, Associate Professor Dr Ahamad Tajudin Khader and Dr Adli Mustafa from School of Computer Science and School of Mathematical Sciences, Universiti Sains Malaysia, respectively, for their wise counselling, valuable advice, motivation, help and guidance throughout the duration of my doctorate study at Universiti Sains Malaysia, Penang. Without their support and cooperation, I could not have completed this study.

I am obliged to Universiti Teknologi MARA (UiTM) and Public Service Department of Malaysia (JPA) for providing the scholarship for my study. This also extends to Institute of Graduate Studies, University Science Malaysia, for their efficient handling of the many aspects of postgraduate programme.

Last, but not least to my beloved wife Faridah Abdul Halim, my daughter Nur Farihin, my son Muhammad Faiz, and my beloved parents for their love, patience, encouragement and continuous support which have made it possible for me to pursue this doctorate study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

1 $\mathcal{D}$ – the timetabling variable for *Department*; d_i represents *department i*.

2 $\mathcal{M}$ – the timetabling variable for *Academic-program*; m_i represents *program i*.

3 C – the timetabling variable for *Course*; c_i represents *course i*.

4 $\mathcal{G}$ – the timetabling variable for *Student-group*; g_i represents *student-group i*.

5 S – the timetabling variable for *Student*; s_i represents *student i*.

6 $\mathcal{P}$ – the timetabling variable for *Staff*; p_i represents *staff i*.

7 $\mathcal{E}$ – the timetabling variable for *Event*; e_i represents *event i*.

8 $\mathcal{T}$ – the timetabling variable for *Timeslot*; t_i represents *timeslot i*.

9 $\mathcal{R}$ – the timetabling variable for *Room*; r_i represents *room i*.

10 δ – the number of *Day* academic-programs.

11 α_1 – the number of *required* courses.

12 α_2 – the number of *Day required* courses.

13 α_3 – the number of *Day optional* courses plus *required* courses.

14 β_1 – the number of *events* of *required* courses.

15 β_2 – the number of *Day events* of *required* courses.

16 β_3 – the number of *Day events* of *optional* courses plus events of *required* courses.

17 $\mathcal{L}_*(*, *, *)$ – a *logic* function; a function of timetabling variables, always has a logic value of either '0' or '1'.

18 f_* - a *penalty* (weight) function that reflects the importance of a soft constraint.

19 $\boldsymbol{t}$ – test statistic for small samples hypothesis

20 υ – degrees of freedom for a $\boldsymbol{t}$-test

21 $\overline{X}_i$ – the sample mean

22 s_i^2 – the sample variance

23 **A** – department–academic-program matrix

24 **B** – department-staff matrix

25 **C** – student-conflict matrix

26 **D** – department-room matrix

27 **E** – academic-program–course matrix

28 **F** – academic-program–student-group matrix

29 **G** – course-student enrollment matrix

30 **H** – course-event allocation matrix

31 **L** – student-group–student allocation matrix

32 **M** – student-group–event allocation matrix
33 **N** – student-group–room preassignment matrix
34 **O** – staff-event preference matrix
35 **P** – staff-event preassignment matrix
36 **Q** – staff-event restriction matrix
37 **R** – staff-timeslot availability matrix
38 **S** – event-timeslot preassignment matrix
39 **T** – event-timeslot restriction matrix
40 **U** – event-room preassignment matrix
41 **V** – event-room restriction matrix
42 **W** – timeslot-room availability matrix
43 **X** – event-staff assignment matrix
44 **Y** – event-timeslot assignment matrix
45 **Z** – event-timeslot–room assignment matrix

LIST OF ABBREVIATION

1 ACS – Ant Colony System
2 AI – Artificial Intelligence
3 AIN / AINE / aiNET – Artificial Immune Network
4 AIRS – Artificial Immune Recognition System
5 AIS / ARTIS – Artificial Immune System
6 AISEC – Artificial Immune System for E-mail Classification
7 AN – Artificial Network
8 APC – Antigen Presenting Cell
9 ARB – Artificial Recognition Ball
10 B-cell (T-cell) – Immune cell matured in Bone Marrow (Thymus)
11 BCR – B-cell receptor
12 BNS – Binary Negative Selection
13 CBR – Case-Based Reasoning
14 CDAI – Cognitive Distributed Artificial Intelligence
15 CHIP – Constraint Handling in Prolog
16 CIFD – Computer Immune system for Fraud Detection
17 CLONALG – Clonal Algorithm

CLONCLAS – Clonal Classification Algorithm
CLP – Constraint Logic Programming
CPU – Central Processing Unit
CS – Computer Science
CSA – Clonal Selection Algorithm
CSAUT – Clonal Selection Algorithm for University Timetabling
CSP – Constraint Satisfaction Problem
CTP – Class Timetabling Problem
DARS – Distributed Autonomous Robotic System
DNA – Deoxyribonucleic Acid
DynamiCS – Dynamic Clonal Selection
EA – Evolutionary Algorithm
ECLiPSe – ECRC Logic Programming System
ECRC – European Computer Research Center
ETP – Examination Timetabling Problem
EXAMINE – Examination Scheduling System
FD – Finite Domain
FSSP – Flow Shop Scheduling Problem
FTI – Fault-Tolerant Implementation
FUVS – Fatih University Vocational School
GA – Genetic Algorithm
GGA – Grouping Genetic Algorithm
GP – Goal Programming
GUI – Graphical User Interface
HCMC – Ho Chi Minh City
HGGA – Hybrid Grouping Genetic Algorithm
HNIS – Hybrid Neuro-Immune System
IDARA – Distributed Autonomous Robotics Architecture
IDS – Intrusion Detection System
ILOG – International Logics Inc.
IP – Integer Programming
IMA – Institute of Applied Mathematics
INA – Immune Network Algorithm
INAUT – Immune Network Algorithm for University Timetabling
IS – Immune System
ISGA – Immune System based Genetic Algorithm

ISP – Immune System Paradigm

JSSP – Job Shop Scheduling Problem

LISYS – Lightweight Intrusion Detection System

m-aiNET – Modified optimization version of Artificial Immune Network

MA – Memetic Algorithm

MHC – Major-Histocompatibility Complex

MILA – Multilevel Immune Learning Algorithm

MMAS – MAX-MIN Ant System

MOEA – Multiobjective Evolutionary Algorithm

NAT – Network Affinity Threshold

NIDS – Network Intrusion Detection System

NP - Nondeterministic Polynomial time

NSA – Negative Selection Algorithm

NSAUT – Negative Selection Algorithm for University Timetabling

NSMutation – Negative Selection Mutation

opt-aiNET – Optimization version of Artificial Immune Network

OR – Operational Research

PCSA – Poly Clonal Selection Algorithm

PRAIS – Pattern Recognizing Artificial Immune System

PSTP – Preparation School Timetabling Problem

RDAI – Reactive Distributed Artificial Intelligence

RLAIS – Resource Limited Artificial Immune System

RNS – Real-Valued Negative Selection

RRNS – Randomized Real-Valued Negative Selection

SA – Simulated Annealing

SANSAD - Self-Adaptive Evolutionary Negative Selection for Anomaly Detection

SA-PCSA – Simulated Annealing Poly Clonal Selection Algorithm

SSGA – Steady State Genetic Algorithm

TCR – T-cell receptor

TEDI – Timetabling Tool for Educational Institutions

TS – Tabu Search

TSHH – Tabu Search Hyper-Heurictic

TSP – Travelling Salesman Problem

UPS – Uninterruptible Power Supply

UTP – University Timetabling Problem

UUTM – Unified University Timetabling Model

LIST OF APPENDICES

LIST OF PUBLICATIONS & SEMINARS

1 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2004a) Artificial Immune Systems: An Overview. *Statistics Colloquium*, Faculty of Information Technology & Quantitative Sciences, UiTM, 25th September 2004.

2 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2004b) Artificial Immune Systems for Scheduling Problems. *Proceedings of the National Seminar on Decision Science 2004 (SKSP2004)*, Faculty of Quantitative Sciences, UUM, 15-17 December 2004, Penang, pp: 63-70.

3 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2005a) University Examination Timetabling: A General Model. *Proceedings of the National Symposium on Mathematical Sciences XIII (SKSMXIII)*, Faculty of Quantitative Sciences UUM, 31st May – 2nd June 2005, Alor Star, pp: 714-724.

4 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2005b) University Course Timetabling: A General Model. *Proceedings of The 2nd International Conference on Research and Education in Mathematics (ICREM2)*, 25-26 May 2005, UPM, Serdang, pp: 834-853.

5 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2005c) PhD Research Proposal: Artificial Immune System Algorithms for University Timetabling. *Proceedings of the National Computer Science Postgraduate Colloquium 05 (NaCSPC'05)*, 27-28 June 2005, USM, Penang, AI01.

6 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2005d) A Clonal Selection Algorithm for University Examination Timetabling. *Proceedings of the First International Symposium on Bio-Inspired Computing (BIC'05)*, 5-7 September 2005, UTM, Johor Bahru, SC01.

7 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2005e) A Negative Selection Algorithm for University Examination Timetabling. *Proceedings of the International Conference on Quantitative Sciences and Its Applications (ICOQSIA2005)*, UUM, 6-8 December 2005, Penang, pp: 47.

8 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2005f) Artificial Immune Algorithms for University Examination Timetabling. *Proceedings of the First Malaysia Software Engineering Conference 2005 (MySEC'05)*, 12-13 December 2005, Faculty of Computer Sciences, USM, Penang, pp: 161-166.

9 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2005g) A Clonal Selection Algorithm for University Course Timetabling. *Seminar Capaian Maklumat & Pengurusan Pengetahuan 2005 (CAMP2005)*, 21st December 2005, UKM, Bangi, pp: 87-95.

10 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2006a) An Immune Network Algorithm for University Examination Timetabling. *Proceedings of the National Seminar on Science and Its Applications in Industry 2006 (SSASI2006),* 14-15 February 2006, KUTKM, Malacca, pp: B173-B178.

11 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2006b) An Immune-Based Approach To University Course Timetabling: Immune Network Algorithm. *Proceedings of the International Conference on Computing and Informatics 2006 (ICOCI'06)*, 06-08 June 2006, FIT UUM, KL. Ref. 119.

12 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2006c) An Immune-Based Approach To University Course Timetabling: Negative Selection Algorithm. *Proceedings of the 2nd IMT-GT 2006 Regional Conference on Mathematics, Statistics and Applications (RCMSA'06)*, USM, 13-15 June 2006, Penang, Volume 4, pp: 151-156.

13 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2006d) Artificial Immune Algorithms for University Course Timetabling. *Proceedings of the First International Conference on Mathematics and Statistics (ICOMS-1)*, 19-22 June 2006, UNISBA Bandung.

14 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2006e) Artificial Immune Algorithms for University Timetabling. *Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling (PATAT2006)*, 30th August – 1st September 2006, Czech Republic, pp: 234-245.

15 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2008a) University Timetabling: A Unified 0-1 Integer Programming Model. Submitted to *Asia-Pacific Journal of Operational Research (APJOR)*, 20th May 2008, Manuscript Number: M080521-a, Current Status: Under Review.

16 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2008b) Examination Timetabling Problems: An Immune Network Approach. Submitted to *The International Arab Journal of Information Technology (IAJIT)*, 28th May 2008, Current Status: Under Review.

17 Muhammad Rozi Malim, Ahamad Tajudin Khader, and Adli Mustafa (2008c) Class Timetabling Using Artificial Immune Algorithms. Submitted to *International Journal of Computational Intelligence Systems (IJCIS)*, 1st November 2008, Current Status: Under Review.

MENGADAPTASI ALGORITMA IMUN REKAAN BAGI PENJADUALAN UNIVERSITI

ABSTRAK

Penjadualan kelas dan peperiksaan di universiti adalah masalah pengoptimuman berkekangan tinggi. Pendekatan metaheuristik, dan hibridnya, telah berjaya menyelesaikan masalah tersebut. Tesis ini bertujuan menghasilkan algoritma-algoritma imun rekaan bagi penjadualan universiti (kuliah dan peperiksaan). Tiga algoritma dipertimbangkan; pemilihan berklon (CSA), rangkaian imun (INA), dan pemilihan negatif (NSA). Objektif utama adalah memperkenalkan algoritma tersebut sebagai kaedah pilihan terbaru bagi penjadualan universiti. Dengan kata lain, membuktikan bahawa algoritma imun rekaan boleh diadaptasikan bagi menyelesaikan masalah penjadualan kelas dan peperiksaan.

Satu model bersepadu (UUTM) dan tiga algoritma imun rekaan (CSAUT, INAUT, dan NSAUT) bagi penjadualan universiti telah dicadangkan. Algoritma-algortima ini telah diuji ke atas set-set data penjadualan piawai (kuliah dan peperiksaan). Masalah piawai tersebut dirumus sebagai pengaturcaraan integer 0-1 menggunakan model bersepadu. Keputusan-keputusan ujian menunjukkan bahawa kesemua algoritma adalah algortima pengoptimuman yang baik; berjaya menghasilkan jadual-jadual berkualiti. Pengoperasi utama adalah pengklonan dan mutasi. Ujian-ujian hipotesis berstatistik menunjukkan bahawa INAUT lebih efektif berbanding CSAUT dan NSAUT, manakala CSAUT dan NSAUT sama efektif. Algoritma-algoritma ini boleh menangani kekangan-kekangan tetap dan tidak tetap dengan baik, dan boleh diterima sebagai algoritma berevolusi baru bagi menyelesaikan masalah penjadualan. Keteguhan relatif yang diukur bagi semua algoritma dan data jelas menunjukkan bahawa jadual-jadual CSAUT adalah lebih teguh daripada yang dihasilkan oleh INAUT dan NSAUT; dengan kata lain, jadual-jadual CSAUT mempunyai keserupaan yang lebih besar berbanding INAUT dan NSAUT. Masa CPU yang direkodkan bagi semua algoritma menunjukkan bahawa INAUT mengambil masa paling lama bagi semua set data.

Perbandingan dengan keputusan-keputusan yang telah diterbitkan menunjukkan keberkesanan algoritma imun rekaan; kesemua algoritma mampu menghasilkan jadual-jadual berkualiti setanding dengan kaedah-kaedah lain. Ketiga-tiga algoritma telah menghasilkan keputusan yang kurang baik dalam hanya dua set data peperiksaan (daripada 12 set data). Populasi awal memainkan peranan yang penting dalam suatu algoritma pengoptimuman. Saiz populasi 10 boleh dikatakan kecil, dan seterusnya membataskan ruang pencarian. Saiz populasi yang kecil akan menyebabkan penumpuan pra-matang dan mengurangkan kebolehpercayaan pengoptimuman. Saiz populasi awal yang lebih baik dan besar akan menghasilkan keputusan yang lebih baik. Malangnya, kekangan-kekangan penjadualan yang kompleks telah membataskan saiz populasi.

ADAPTING ARTIFICIAL IMMUNE ALGORITHMS FOR UNIVERSITY TIMETABLING

ABSTRACT

University class and examination timetabling are highly constrained optimization problems. Metaheuristic approaches, and their hybrids, have successfully been applied to solve the problems. This thesis aims to develop artificial immune algorithms for university timetabling (class and examination). Three algorithms are considered; clonal selection (CSA), immune network (INA) and negative selection (NSA). The ultimate goal is to introduce the algorithms as new alternative approaches for university timetabling. In other words, to show that artificial immune algorithms can be adapted for solving class and examination timetabling problems.

A unified model (UUTM) and three artificial immune algorithms (CSAUT, INAUT and NSAUT) for university timetabling are proposed. The algorithms have been tested on benchmark datasets (class and examination). The benchmark problems have been formulated as 0-1 integer programming using the unified model. Experimental results have shown that all algorithms are good optimization algorithms; have successfully produced good quality timetables. The main operators are cloning and mutation. Statistical tests of hypotheses have shown that INAUT is more effective than CSAUT and NSAUT, while CSAUT and NSAUT are equally effective. The algorithms can handle the hard and soft constraints very well, and may be accepted as new members of evolutionary algorithms for solving timetabling problems. The relative robustness measured for all algorithms on all datasets (class and examination) have significantly shown that the CSAUT timetables are more robust than those produced by INAUT and NSAUT; i.e. CSAUT timetables have large similarity compared to INAUT and NSAUT. The CPU times recorded on all algorithms have shown that INAUT has acquired the longest times on all datasets.

Comparisons with published results have shown the effectiveness of the proposed artificial immune algorithms; all three algorithms are capable of producing good quality timetables as good as other methods. The three AIS algorithms have produced poor results in only two examination datasets (from 12 datasets). Initial population plays a crucial role in an optimization algorithm. The population size 10 is considerably small, and hence has limited the search space. A relatively small population size would results in premature convergence and decreases the optimization reliability. A better and larger size of initial population would produce better results. Unfortunately, the complexity of timetabling constraints has limited the population size.

CHAPTER ONE
INTRODUCTION

1.0 Introduction

Various facets of *biology* have always been the inspiration in developing computational models and problem solving methods. The rapid increase in research of the biological systems (human body and genetics) has enabled us to gain insight into the miraculous operation of our body. The *theory of evolution* has been used within the field of evolutionary computation, the *artificial neural network* was inspired by the way the brain functions, the *ant colony* algorithms are modeled on the behaviors exhibited by real ants, and more recently, the *concept of DNA* computing has arisen, inspired by the processes that govern life itself. The use of biologically inspired metaphors can result in new computer technologies and methods of problem solving, and computing can provide new techniques for exploring biological concepts from an alternative prospective.

The *immune system* (IS), a biological system, has recently drawn significant attention; and as a result, the *artificial immune system* (AIS) has emerged. In 1986 the theoretical immunologist, J.D. Farmer, first suggested a possible relationship between *immunology* and *computing* [Farmer et al., 1986]. Since then, the field has expanded rapidly, with numerous papers published applying AIS to a diverse set of topics ranging from computer security [Forrest et al., 1994] to robotics [Ishiguro et al., 1997].

1.1 Immunology

Immunology is a relatively new science. It is the study of the body's defenses against infection. The birth of immunology as an experimental science dates to Edward Jenner's successful smallpox vaccination in 1796. The worldwide vaccination acceptance led to mankind's greatest achievements in preventing disease, and smallpox is the first and only human disease that has been eradicated [Janeway Jr. et al., 2001]. During the 20^{th} century, the impact of immunology has moved beyond defense against infections.

There are four main causes of human death; injury, infection, degenerative disease and cancer. Of these, only the former two regularly kill their victims before they reach child bearing age, and as such are a potential source of lost genes. The immune system (IS) is an example of a mechanism which may help to ensure the survival of those genes, and has evolved over time in order to protect us from infectious organisms existing in the environment. The immune response broadly falls into two categories, the *innate* (non-specific) and the *adaptive* (specific) [Janeway Jr. et al, 2001]. The innate response is provided by a number of non-specific chemicals such as *lysozyme* which destroys the outer surface of many bacteria, non-specific chemical effectors such as *macrophages* and simple barrier mechanism such as the skin. On the other hand, the adaptive response is highly specific for particular *pathogens* (microorganisms), and it improves with each subsequent exposure to the pathogen. It is with this adaptive aspect of the IS that artificial models are generally concerned.

The adaptive response consists of two major phases [Hart, 2002]; a *recognition* followed by a *reaction* to eliminate the pathogens. Recognition requires the IS to distinguish between the body's own cells (*self*) and foreign pathogens (*nonself*). As far as computer scientists are concerned, the task essentially remains a recognition followed by an action, and it is the mechanisms by which the natural IS achieves these aims that make it so attractive to information processing.

1.2 Immune System Features and Relevance to Information Processing

There are two systems in human being that possess extraordinary capabilities of information processing such as learning and memory, and ability to recognize patterns and to make decisions about how to behave in an unfamiliar environment [Antoniou et al., 2001]; *nervous system* and *immune system*. The immune system is incredibly *robust* [de Castro and Timmis, 2002a]; it is a remarkably efficient and powerful information processing system, and contains several features that make it appealing

from a computational point of view. These features, the well-known terminology of information processing, are summarized below; adopted from Dasgupta (2000).

- *Recognition:* Recognizes and classifies different patterns and generates responses.
- *Feature extraction:* Features are extracted from pathogens by *antigen presenting cells* (APC) which present the features of the pathogens on their surface.
- *Diversity:* Utilizes a combinatoric process to generate a diverse set of pathogen recognizing molecules.
- *Learning:* Learns by experience the structure of specific antigens, following the first exposure of the system to a new antigen.
- *Memory:* When the system has been activated, a few lymphocytes become special *memory cells* which are then content-addressable.
- *Distributed detection:* The system is inherently distributed throughout the body.
- *Self-regulation:* There is no central organ coordinating the response, and therefore the mechanisms are self-regulatory, although not necessarily stable.
- *Threshold Mechanism:* A response and the subsequent proliferation of immune cells only takes place above a certain matching threshold.
- *Co-Stimulation:* Activation of immune cells is regulated through co-stimulation by a second signal from *helper T-Cells*.
- *Dynamic protection:* The processes governing generation of high-affinity immune cells dynamically balance exploration vs exploitation in adaptive immunity.
- *Probabilistic detection:* Detection of antigens is approximate; therefore a lymphocyte can bind with several different kinds of structurally related antigen.

Thus the IS contains a number of general mechanisms which can be adapted into computational systems for solving complex problems. It is unnecessary to replicate all of these features; rather they should be used as general guidelines in designing a system. Most AIS applications only implement some modified subset of these features. Note that several of these features are apparent in other biologically inspired systems.

1.3 Motivation

The potential application areas of the immune system (IS) metaphors are those seeking *robust* and *good-enough* solutions to problems occurred in dynamic environments [Hart, 2002]. These features are characteristic of a number of real-world problem domains; anomaly detection, pattern recognition, computer and network security, hardware fault tolerance, dynamic environments, dynamic learning, robotics, diagnosis and control, data analysis, optimization, and scheduling.

The task of producing *robust schedules* (large similarity) has a direct analogy with the task faced by the IS [Hart, 2002]; both operate in a dynamic and unpredictable environment. The IS must mount an efficient and immediate antibody response against invaders to survive. Similarly, to minimize costs, a useful scheduling system should be able to mount a response to environmental changes by rapidly altering schedules so that minimum disruption is caused. The antibodies produced by the IS do not have to perfectly match the invading pathogens, similarly the new schedules produced by the scheduling system do not have to be optimal, just *good-enough* for the scheduling to continue with the least interruption. Some or all of the characteristics of the IS may be adapted to implement a scheduling system. The AISs are relatively new techniques. Most of the scheduling problems considered by AIS researchers were *flow-shop* and *job-shop*. These should attract more attention from researchers to apply AIS approaches on other types of scheduling problems such as *timetabling*.

AIS approaches have been successfully applied to optimization and scheduling problems [Hart et al., 1998; Hart and Ross, 1999; Carlos et al., 2003; Doyen et al., 2003; Walker and Garrett, 2003; Coello et al., 2004]. Since timetabling is a special case of scheduling, and treated as optimization by the OR community, the AIS approaches may be adapted for timetabling problems to produce good quality (and robust) timetables. This is the main motivation behind the research study presented in this thesis.

1.4 Research Problem

Class and *examination* timetabling are known to be highly complex scheduling problems. These problems are always studied because of its variety and complexity. *Modeling* a timetabling problem is the first and crucial step before solving it. With a good *mathematical model*, the complexity of the problem may be reduced. Most of the models developed by previous researchers were *problem-based*, and the class and examination are always considered as two different problems. In an education institution, the class and examination timetables are constructed from one system. A *unified* or *generic* model that can be applied for both class and examination timetabling would reduce the administrative work. Both timetables may be constructed simultaneously.

The increase in the number of students (and courses) and complexity of program structures mean that timetables are becoming more complex and difficult to schedule. New timetables must be produced for every single semester to take account of staff, student and course changes, causing a large amount of administrative work. Small changes in timetabling data would ruin the feasibility of a timetable. AIS approaches may be applied to produce *robust timetables* (large similarity timetables). With a set of robust timetables, another feasible timetable (suite the changes) may be selected. In scheduling and AIS, the term *robustness* is synonym with the flow-shop and job-shop scheduling problems [Hart et al., 1998; Kawata et al., 2003] but not for timetabling.

Heuristic methods are often used to solve real-world timetabling problems. The most popular and well-studied heuristics are *metaheuristics* which include simulated annealing, tabu search, and evolutionary algorithms. However, many of these approaches *lack the robustness*. AIS schedules are *robust* and observed to be robust than schedules produced by a standard *genetic algorithm* [Hart et al., 1998]. However, no timetabling researchers have applied AIS algorithms. AIS and Timetabling are two separate disciplines, and there already exist a large number of good timetabling algorithms.

1.5 Objectives and Contributions of Thesis

The aim of this thesis is to develop a number of *AIS algorithms for university timetabling* (class and examination). The ultimate goal is *not* to show that the AIS algorithms are better than other well-established algorithms in timetabling, but to introduce the algorithms as new alternatives for solving timetabling problems.

The main objectives of this thesis are as follows:

1. Describe and propose a *unified model* for university timetabling problems (class and examination).
2. Develop *immune system-based algorithms* for university timetabling problems based on *three immunological principles* (immune network, negative selection, and clonal selection); i.e. *three* AIS algorithms are considered.
3. Implement the unified model and AIS algorithms on benchmark timetabling datasets (empirical testing and comparison with other approaches).

The main contributions of this thesis are:

1. *Unified model*: The model shall be employed to formulate various university timetabling problems. Then a timetabling algorithm may be applied to solve the problems and produce the desired timetables. Also, based on this model, a comparison of various timetabling algorithms can be carried out.
2. *AIS algorithms*: These algorithms shall be considered as new alternative approaches for solving university timetabling problems. The AIS algorithms may produce not only a set of good quality university timetables (comparable with other methods), but also a set of robust timetables (large similarity).

1.6 A Brief Overview of Methodology

This section briefly discusses the methodology employed to realize the objectives of this research. The detailed methodology is described in Chapter 4. For the first objective, a number of different university timetabling models and constraints

found in the literature are studied. Then a *unified model* for the university timetabling problems (UTPs) is proposed, and formulated as 0-1 *integer programming* (IP). This would show that the AIS algorithms can be applied to *all* UTPs, and not just for a number of benchmark problems considered in the implementation stage.

For the second objective, the approaches, models, and algorithms used in the literature on AIS and university timetabling are studied. This would generate a general idea on how to adapt the IS metaphors for solving UTPs. Then *three* immune system-based algorithms for university timetabling are developed. Finally, for the third objective, the three algorithms are implemented and compared (fitness and robustness) using the benchmark datasets (class and examination) available on the internet. The results are also compared with other published results to assess the effectiveness of the algorithms.

1.7 Contents of Thesis

Chapter 2 introduces some basic immunology for computer scientists. This is followed by a review of a number of different AIS models (and algorithms), which identifies the features of the natural IS each model contains and discusses the types of application to which each has been applied. Chapter 3 contains detailed descriptions of the UTPs. The timetabling approaches and models found in the literature are studied. All hard and soft constraints are gathered, and a structure of university timetabling variables is presented. The methodology (approaches and methods) employed to realize the objectives of this research are described in Chapter 4. A unified model for UTPs (class and examination) is proposed in Chapter 5. All common hard and soft constraints are considered, and mathematically formulated as 0-1 IP constraints. In Chapter 6, three AIS algorithms for UTPs are developed and described in details. These algorithms are implemented on a number of benchmark timetabling datasets (class and examination) available on the internet in Chapter 7. The thesis is concluded in Chapter 8.

CHAPTER TWO
BASIC IMMUNOLOGY AND ARTIFICIAL IMMUNE SYSTEMS

2.0 Introduction

This chapter begins with a brief discussion on *immunology*. It is a huge topic, and only the relevant features are covered. For a more detailed overview, please refer to an introductory text such as Janeway Jr. et al. (2001). This is followed by an overview of the immunological principles used in artificial immune systems (AIS). Basic components for AIS are discussed in Section 2.3. Most of the AIS algorithms found in the literature are gathered in Section 2.4; AIS implementations are reviewed and compared. AIS applications to various domains are discussed in Section 2.5.

2.1 Basic Immunology (for Computer Scientists)

The true roots of immunology date from 1796 when an English physician, Edward Jenner, discovered a method of smallpox vaccination. He noted that dairy workers who contracted cowpox from milking infected cows were thereafter resistant to smallpox. Jenner injected a young boy with material from a milkmaid who had an active case of cowpox. After the boy recovered from his own resulting cowpox, Jenner inoculated him with smallpox; the boy was *immune*. After Jenner published the results of this and other cases in 1798, the practice of *Jennerian vaccination* spread rapidly.

When Edward Jenner introduced the smallpox vaccination in 1796, he knew nothing of the infectious agents that cause disease. It was not until late in the 19th century that Robert Koch proved that infectious diseases are caused by *microorganisms*, each one responsible for a particular disease, or *pathology* [Janeway Jr. et al., 2001]. There are four broad categories of disease-causing microorganisms, or *pathogens*: viruses, bacteria, pathogenic fungi, and parasites. In 1890, Emil von Behring and Shibasaburo Kitasato discovered that the serum of vaccinated individuals contained substances called *antibodies* that specifically bound to the relevant pathogen.

During the 1880s, a Russian microbiologist, Elie Metchnikoff discovered that many microorganisms could be engulfed and digested by *phagocytic cells*, called *macrophages*. These cells are immediately available to combat a wide range of pathogens without prior exposure. Antibodies, by contrast, are produced only after infection, and are specific for the infecting pathogen. The antibodies present in a person directly reflect the infections to which he or she has been exposed. It became clear that specific antibodies can be induced against a vast range of substances, known as *antigens* (*gen*eration of *anti*bodies). By the early 1900s, immunology had become an established medical field with its own journals, first Germany in 1909 and then USA in 1916.

2.1.1 The Components of the Immune System

The immune system (IS) is a complex system that includes *organs*, *cells*, and *molecules* [Timmis, 2001]. The IS organs are divided into *primary lymphoid organs* (thymus and bone marrow) and *secondary lymphoid organs* (Figure 2.1).

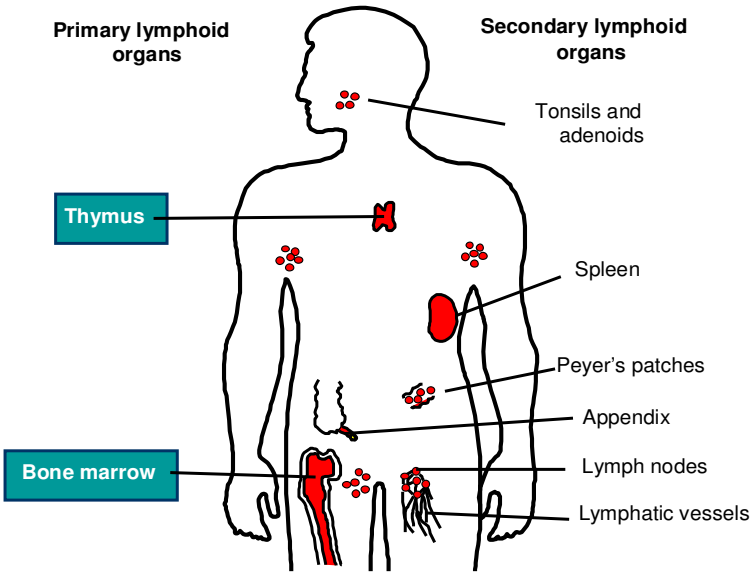

Figure 2.1: Immune System Organs [Timmis, 2001]

The cells of the IS originate in the *bone marrow*. They then migrate to guard the peripheral tissues, circulating in the blood and in a specialized system of vessels called

the *lymphatic system*. The main purpose of the system is to recognize all cells within the body and categorize those cells as *self* or *nonself*. The nonself cells are further categorized to induce an appropriate type of defensive mechanism. The system learns through evolution to distinguish between foreign antigens and the body's cells.

The architecture of the immune system is *multilayered*, with defenses provided at many levels [Timmis, 2001]. In Figure 2.2, the outermost layer, the *skin*, is the first barrier to infection. A second barrier is *physiological*, where conditions such as pH and temperature provide inappropriate conditions for some pathogens. Once pathogens have entered the body, they are handled by the *innate* and *adaptive* immune systems.

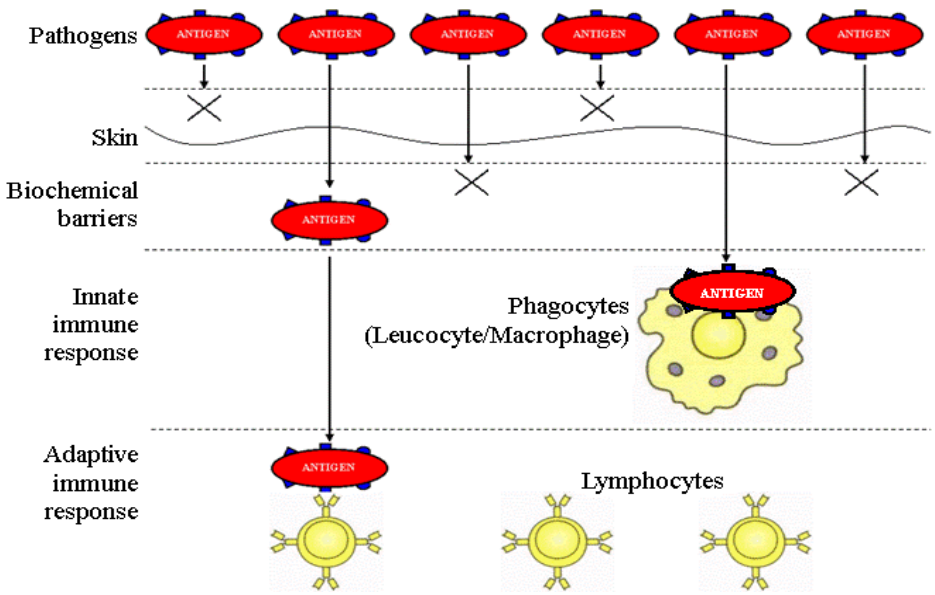

Figure 2.2: Multiple Layers of the Immune System [Timmis, 2001]

2.1.2 Innate and Adaptive Immune Systems

The innate and adaptive immune systems work together (Figure 2.3). The *innate system* is the first line of defenses that composed primarily of circulating *complement system* and *phagocytes*; include many white blood cells (*leukocytes*) and macrophages. *Complement* is a system of blood (plasma) proteins that interacts with pathogens to mark them for destruction by phagocytes [Janeway Jr. et al., 2001]. The innate response

makes a crucial contribution to the activation of adaptive immunity. The *adaptive system* is the most sophisticated and involves many different types of cells and molecules; composed mainly of white blood cells called *lymphocytes*. It is called *adaptive* because it is responsible for immunity that is adaptively acquired during the lifetime of the organism. It can be viewed as a *distributed detection system* [Somayaji et al., 1997].

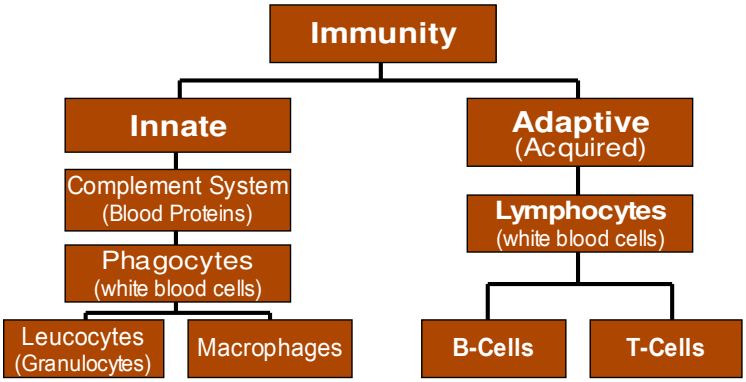

Figure 2.3: The Innate and Adaptive Immune Systems

2.1.3 Lymphocytes: B-cells and T-cells

Lymphocytes are the basic building blocks of the adaptive system. There are two major classes of lymphocytes; *B*- and *T-cells*. B-cells are generated and matured in the *b*one marrow, while T-cells are generated in the bone marrow and matured in the *t*hymus. Before maturation, both cells are exposed to the self proteins. If they bind to a self protein they are censored or deleted. If not, they mature and leave the bone marrow or thymus. These cells generate specific *antigen receptors* on their surface and circulate through the blood and lymph systems. The B-cells are capable of fine-tuning the cell receptors (*diversity*). Both B- and T-cells confer resistance against future infections (*memory*). The function of the B-cell is to attack extracellular pathogens by releasing antibodies which recognize and bind to target antigens. T-cells function by interacting with other cells, and divides into *helper* and *killer* T-cells. Helper T-cells activate B-cells, promoting their growth and differentiation into an antibody-secreting state. Activated B-cells cut antigens into smaller parts (*peptides*) and present them to killer T-cells.

A critical difference between B- and T-cells is how each *sees* its antigen. B-cells recognize their antigen in its native form. In contrast, T-cells recognize their antigen in a processed form, as a peptide fragment presented by a MHC (*Major-Histocompatibility Complex*) molecule. MHC molecules enable the IS to detect intracellular pathogens (e.g. viruses) that reside inside cells. Intracellular pathogens are *invisible* to lymphocytes. Lymphocytes can only bind on the surface of cells. MHC molecules bind to peptides within a cell and transport them to the surface, displaying the contents to lymphocytes.

2.1.4 Immune Recognition: B-Cell and T-Cell Receptors

The detection or immune recognition is based on part of the pathogens known as *antigens* and on part of the immune cells known as *antibodies*. The process is mainly handled by *B-cell receptors* (BCRs) and *T-cell receptors* (TCRs).

(a) B-Cell Receptors (BCRs)

BCRs, or antibodies, are proteins found on the surface of B-cells. There are several types of BCRs, but an individual B-cell can only produce one. BCRs bind antigens, triggering proliferation of B-cells and production of antigen-specific immunoglobulin. This also results in the production of memory (*plasma*) cells, which will proliferate if the body is ever exposed to the antigen again. B-cell recognition of antigen is not the only element necessary for B-cell activation. *Naive* B-cells that have not been exposed to antigen can be activated in a T-cell dependent or independent manner.

The antibodies have a distinct molecular structure, a flexible Y-shaped (Figure 2.4), and recognize the shape of particular antigen via a mechanism often likened to a *lock* and *key* [Hart, 2002]. The portion of the antigen (*epitope*) acts as the lock, and the portion of the antibody (*paratope*) acts as the key. Antibodies are made up of *four* protein chains [Janeway Jr. et al., 2001]. There are two types of chain; a larger called *heavy* (H) and a smaller called *light* (L). Each chain has both *variable* (V) and *constant* (C) regions.

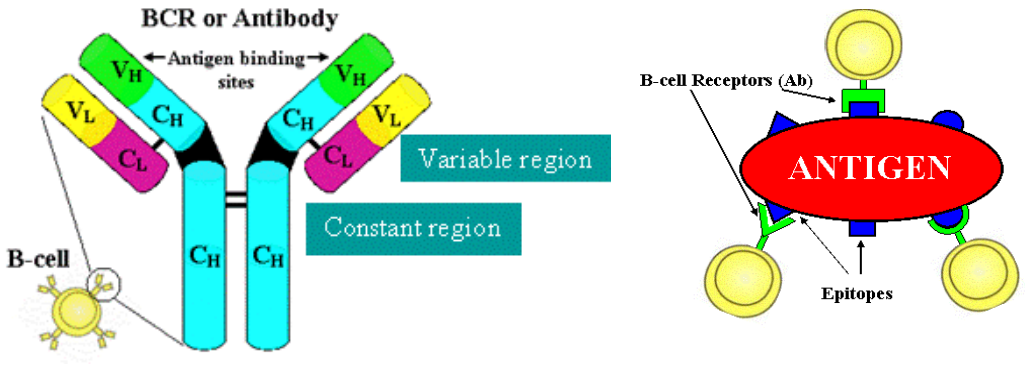

Figure 2.4: B-cell Antigen Receptors (BCRs) [adapted from Timmis (2001)]

(b) T-cell Receptors (TCRs)

The T-cell receptor is a molecule found on the surface of T-lymphocytes that is responsible for recognizing antigens bound to major MHC molecules. It consists of an *alpha* and *beta* chain in 95% of T-cells [Janeway Jr. et al., 2001], see Figure 2.5.

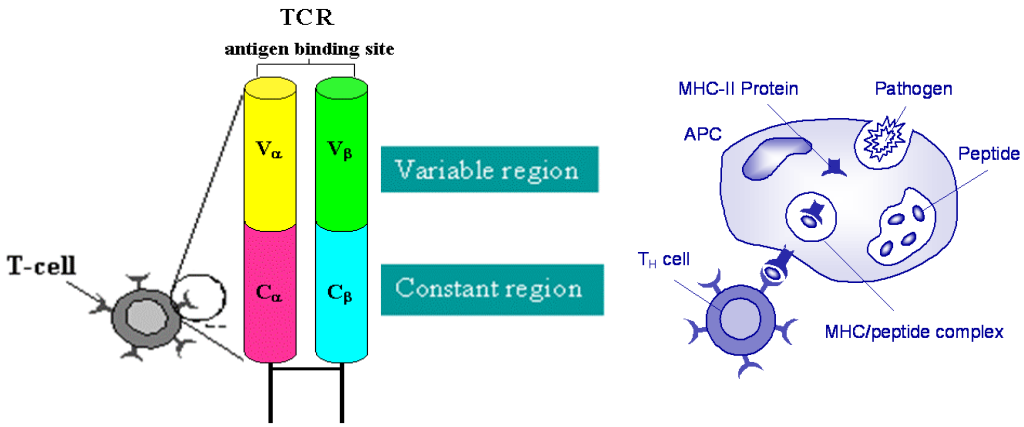

Figure 2.5: T-cell Antigen Receptors (TCRs) [adapted from Timmis (2001)]

Engagement of TCR with antigen and MHC results in activation of T-lymphocyte through a series of biochemical events mediated by associated enzymes, co-receptors, and specialized molecules. Each chain of the TCR is a member of the immunoglobulin super-family and possesses one variable (V) region and one constant (C) region.

(c) Activation Mechanism of B-cells and T-cells

A B-cell is triggered when it encounters its matching antigen. When a B-cell ingests a pathogen, it attaches parts of the pathogen's proteins to a MHC protein. This complex is moved to the outside of the cell membrane, where it can be recognized by a T-cell. If the structures match, the T-cell activates B-cell, which produces antibodies against the presented antigen. Most antigens are T-dependent; T-cell help is required for maximal antibody production. With a T-dependent antigen, the first signal comes from antigen cross linking the BCR and the second signal comes from co-stimulation by a T-cell. When a B-cell processes and presents the same antigen to the T-cell, the T-cell secretes *cytokines* that activate B-cell. These cytokines trigger B-cell proliferation and differentiation into plasma cells. Then the plasma cells are released into the blood, and antibodies are locked onto matching antigens. After reacting against a pathogen most of the B-cells die, but others *adapted* (memory) *B-cells* are kept alive and will be used again when the same pathogen invades the organism.

2.2 The Immunological Principles

Three immunological principles are primarily used in AIS methods [Dasgupta et al., 2003b]. These include the *clonal selection principle*, the *immune network theory*, and the *negative selection mechanism*.

2.2.1 Clonal Selection Principle

The immune response is specific to a certain antigen. When an antigen is detected, those antibodies that best recognize this antigen will proliferate by *cloning*. This process is called *clonal selection principle* [de Castro and Von Zuben, 1999]. The *clonal selection theory* is used to explain how the IS *fights* against an antigen. When a bacterium invades our organism, it starts multiplying and damaging our cells. One form the IS found to cope with this replicating antigen was by replicating the cells successful in recognizing and fighting this disease-causing element. Those cells reproduce themselves

asexually in a way proportional to their degree of recognition (*affinity*); the better the recognition, the higher the number of clones. During the cell division process, the cells suffer a *mutation* that allows them to become more adapted to the antigen recognized.

In the *original* clonal selection theory [Burnet, 1959], memory would be provided by expanding the size of an antigen-specific clone, and a random mutation would be allowed to enhance affinity. The developing self-reactive lymphocytes are removed before they can mature (*clonal deletion*). The IS also practices molecular selection of receptors. Instead of the expected clonal deletion, occasionally B-cells had undergone *receptor editing*; B-cells had deleted their self-reactive receptors and developed entirely new receptors (*somatic hypermutation*). Any high affinity clone developed by editing would be expected to be expanded, but a few low affinity cells are also allowed to enter the repertoire, maintaining the population diversity. Clonal selection is related to *Darwin's theory* of natural selection but applied to the cell populations within the IS [Forrest et al., 1993]. The analogy with natural selection is that the fittest clones are the ones that recognize the antigen, survive and grow, whereas clones that do not recognize the antigen die and are replaced by others. The clones that grow also turn on a mutation mechanism. This results in cells that have very high affinity matches with the antigen.

When an individual is exposed to an antigen, B-cells respond by producing antibodies [de Castro and Von Zuben, 2000]. By binding to these antibodies and a second signal from helper T-cell, the antigen stimulates B-cell to proliferate and mature into non-dividing plasma cells, see Figure 2.6. While plasma cells are the most active antibody secretors, large B-cells also secrete antibodies at a lower rate. B-cells can also differentiate into long-lived memory B-cells. Memory cells circulate through the blood, lymph and tissues, and when exposed to a second antigenic stimulus, commence to differentiate into large lymphocytes capable of producing high affinity antibodies.

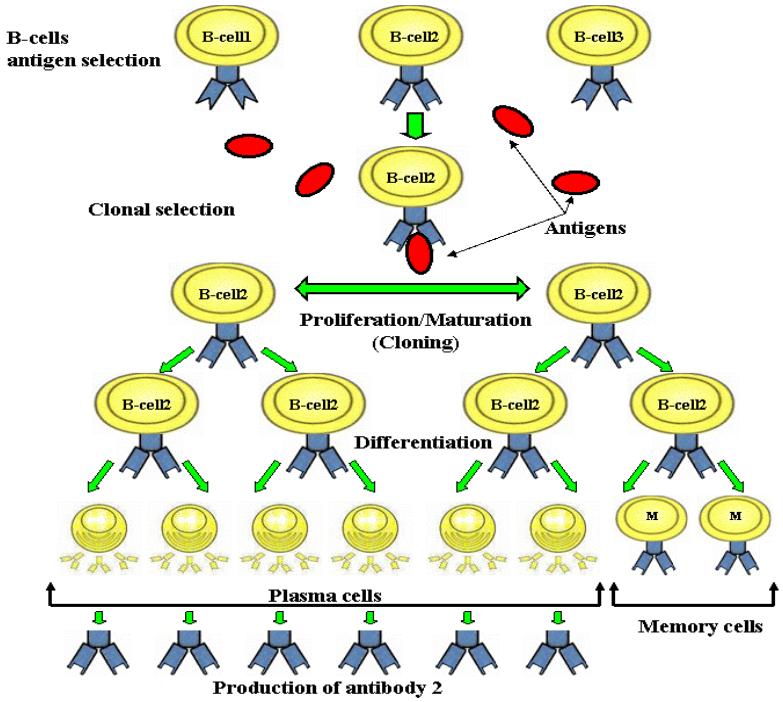

Figure 2.6: The Clonal Selection Principle [de Castro and Von Zuben, 2000]

The main features of Clonal Selection Principle are:

(i) *Proliferation and differentiation*: The antigen stimulates the B-cell to proliferate, differentiate and mature into plasma cells.

(ii) *Recognition*: The better the recognition, the higher the number of clones.

(iii) *Affinity*: The higher the affinity of the parent cell, the lower the mutation.

(iv) *Clonal deletion*: Developing self-reactive B-cells are removed before maturity.

(v) *Receptor editing*: Self-reactive B-cells delete their self-reactive receptors and develop entirely new receptors.

(vi) *Somatic hypermutation*: New cells are copies of their parents (clones) subjected to a mutation mechanism with high rates.

2.2.2 Immune Network Theory

Jerne (1974) formalized what is to date known as the *immune network theory*. He introduced the idea of *immune networks* to explain how the IS maintains its memory of an antigen. Jerne suggested that IS maintains an *idiotypic network* of interconnected

B-cells for recognition. These cells stimulate and suppress each other in certain ways that lead to the stabilization of the network. B-cells carry on their surface highly specific antibodies. B-cells and antibodies of a given specificity are said to have a certain *idiotype*. Between B-cells of different idiotype thus emerges a functional network of mutual stimulation and inhibition (idiotypic network).

Figure 2.7 illustrates the Jerne's immune network. The network is supposed to contribute to the functionality of the IS as *immunological memory* [Hart, 2002]. At a given moment, the random network has certain architecture. Knowledge of this architecture is crucial for describing the population dynamics of the interacting B-cells and antibodies. Each antibody has a specific antigen determinant, the *idiotope*. This gives a possibility that antibodies can recognize other antibodies as well as antigens, resulting in a large, self-regulating and mutually reinforcing network of antibodies.

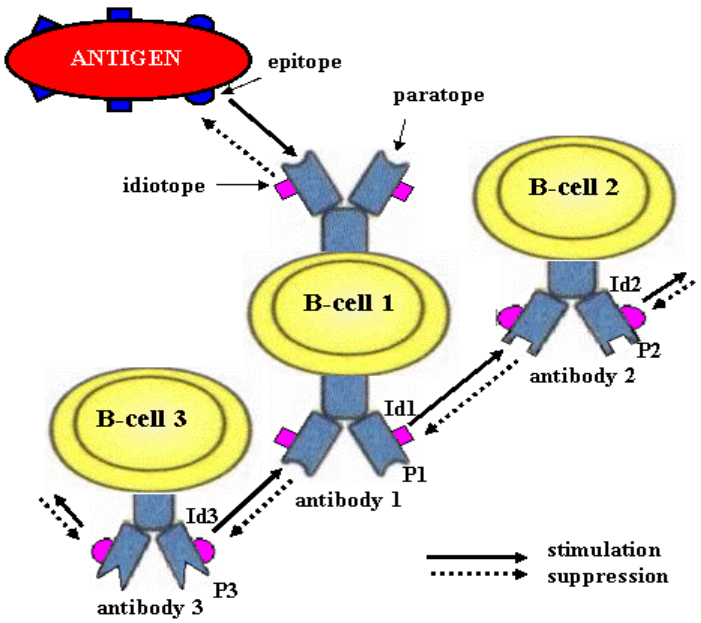

Figure 2.7: The Immune Network Interactions [Hart, 2002]

In Figure 2.7, the idiotope of B-cell1 (Id1) stimulates B-cell2, and become connected via the paratope of B-cell2 (P2). Thus, Id1 is acting as an antigen from the

viewpoint of B-cell2 and causes B-cell2 to suppress the antibodies produced by B-cell1. On the other hand, Id3 acts as an antigen from the viewpoint of B-cell1, and is recognized by P1, and thus Id3 stimulates B-cell1 to produce antibodies. Hence, a large chain of suppression and stimulation can be set up between B-cells, resulting in a self-organizing and self-regulatory network. The network varies continuously according to the changes in environment. This is known as the *metadynamics* of the system and is realized by incorporating new cells into the network and removing useless ones. The network remains stable due to the suppression mechanisms.

When the network theory was originally proposed by Jerne (1974), it did not aim at explaining cell signaling, neither the mechanisms of interaction between antibodies and cells, nor to deal with the effector mechanisms that may become operative as a result of this interaction [de Castro and Von Zuben, 1999]. Instead, it hypothesized a novel viewpoint of lymphocyte activities, antibody production, pre-immune repertoire selection, tolerance and self/nonself discrimination, memory and the evolution of the IS. It was suggested that the system is composed of a regulated network of molecules and cells that recognize one another even in the absence of antigens.

The three remarkable characteristics of the immune networks are:

(i) *Structure*: Network structure describes the types of interaction among molecules and cells, without reference to the functional consequences that they might have.

(ii) *Dynamics*: Immune dynamics accounts for the interactions among immune components. A unique property of the IS that goes beyond the network dynamics is the continuous production of novel antibodies.

(iii) *Metadynamics*: Metadynamics represents the immune mechanism for learning, and thus the source of the network's ontogenic plasticity in each individual.

2.2.3 Negative Selection Mechanism

Negative selection describes the process where a *lymphocyte-antigen* interaction results in the death of that lymphocyte [de Castro and Von Zuden, 1999]. The *negative T-cell selection* occurs within the thymus from the interactions of immature self-reactive T-cells with self-MHC, and results in cell's death to purge autoreactive cells from the repertoire. T-cells with TCRs that do not exhibit significant interactions with self-MHC are also lost from the repertoire.

In Figure 2.8, a minority of cells will recognize the antigen, and be activated by clonal selection. If the affinity between the TCR and the antigen is low or a self-reactive cell is detected, the lymphocyte might suffer *apoptosis* (programmed cell death), *anergy* (freezing), or a *receptor editing* process - a *negative selection* process. The purpose of negative selection is to provide *tolerance* for self cells [Dasgupta et al., 2003b]. It deals with the IS's ability to detect unknown antigens while not reacting to the self. Only 5% of developing T-cells are exported as matured T-cells. Those cells that react against self cells are destroyed. The matured T-cells then circulate throughout the body to perform immunological functions and protect the body against foreign antigens.

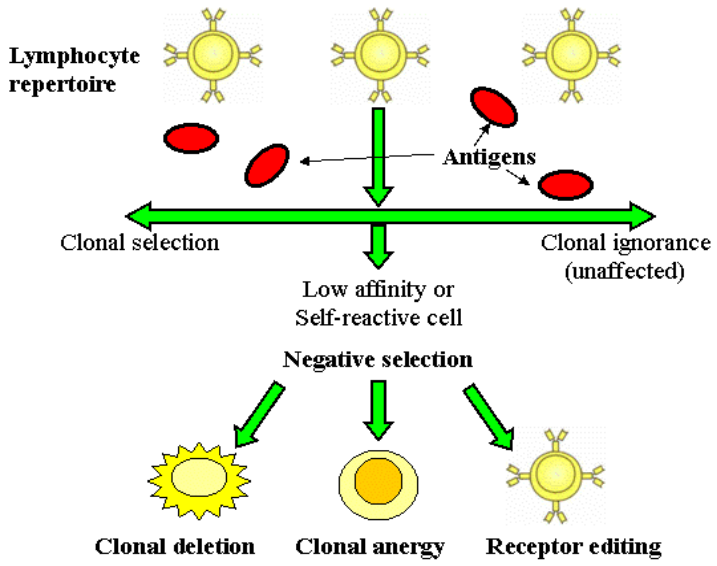

Figure 2.8: The Negative Selection process [de Castro and Von Zuben, 1999]

The IS makes use of the *gene libraries* in thymus and bone marrow [Kim and Bentley, 2002]. When an antibody is generated, the gene segments of different gene libraries are randomly selected. Hence, a vast number of antibodies can be generated. A large number of molecules on antibodies allow huge variability of the receptors. A single antibody will recognize more than one type of antigens (*diversity*).

The main features of Negative Selection Mechanism are:

(i) *Self tolerance*: Ability to detect unknown antigens while not reacting to the self.

(ii) *Gene libraries*: Make use of gene libraries in the thymus and bone marrow; a vast number of antibodies can be generated from gene segments in the gene libraries.

(iii) *Diversity*: Ability to detect a vast number of antigens with a small number of cells.

2.3 Artificial Immune Systems

Artificial immune systems (AIS) are adaptive computational systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to complex problem domains [de Castro and Timmis, 2002a]. To design AIS, it is necessary to choose an appropriate *shape-space* for the components of the system, one or more *affinity measure*, and an *immune algorithm* [de Castro, 2002].

2.3.1 Shape-Space

Shape-space is a formalism used to create *artificial* representations for the components of the IS. The *shape* of an immune cell corresponds to all the features required to quantify interactions between the cell and the environment, and also with other elements of the system. There are four main types of shape-spaces: *Euclidean*, *Hamming*, *Integer*, and *Symbolic*. In *Euclidean*, the elements are represented *as real-valued* vectors. In *Hamming*, the elements are attribute strings of a finite alphabet. In *Integer*, cells are represented as integers. *Symbolic* uses different types of attributes to represent a single element; for example, an integer value and a string such as *color*.

2.3.2 Affinity Measure

There are two main types of interactions that can be performed by an element of AIS. One is the interaction with the environment. For example, an AIS can be used as a pattern recognition tool, thus the *artificial immune cells* are used to recognize a set of *artificial antigens* (patterns). The degree of recognition is measured via a function that quantifies the strength of the match. Assuming two cells interact to the extent their *shapes* are similar, and then a similarity measure can be used according to the shape-space adopted. The second type is the interaction between immune cells.

2.3.3 Immune Algorithms

There are a number of different immune algorithms that can be applied to many domains. These algorithms were inspired by works on theoretical immunology and several processes that occur within the IS. They can be classified as *population*-based and *network*-based algorithms. In population-based, the elements of the system are not connected, they only interact directly with the environment. The interactions of the elements can only be performed indirectly, via, for example, a reproductive operator. In network-based, by contrast, some or all elements are interconnected. There are two levels of interaction; with the environment and with other elements in the system.

2.4 Artificial Immune Algorithms

Here, *three* types of artificial immune algorithms are discussed; each inspired by the respective immunological principle; *clonal selection algorithm*, *immune network algorithm*, and *negative selection algorithm*. The immune network algorithms are network-based, while clonal selection and negative selection are population-based.

2.4.1 Clonal Selection Algorithm

Clonal selection algorithm (CSA) was designed based on the clonal selection principle proposed by de Castro and Von Zuben (1999). The principle defines the basic

features of an immune response to an antigenic stimulus. It establishes the idea that only those cells that recognize the antigens are selected to proliferate. The selected cells are subject to an affinity maturation, which improves their affinity to the selective antigens. de Castro and Von Zuben (1999) discussed a computational implementation of CSA. The algorithm (Figure 2.9) was shown to be capable of solving complex *machine-learning* tasks, like pattern recognition and optimization.

1. *Generate a set (P) of candidate solutions, composed of the subset of memory cells (M) added to the remaining (Pr) population (P = Pr + M).*
2. *Determine the n best individuals of population P based on an affinity measure.*
3. *Clone these n best individuals, giving rise to a temporary population of clones (C); the clone size is an increasing function of the affinity measure of the antigen.*
4. *Submit the population of clones to a hypermutation scheme, where the hypermutation is proportional to the affinity of the antibody. A matured antibody population is generated (C*).*
5. *Reselect the improved individuals from C* to compose the memory set. Some members of the P set can be replaced by other improved members of C*.*
6. *Replace low affinity antibodies of the population, maintaining its diversity.*

Figure 2.9: Clonal Selection Algorithm [de Castro and Von Zuben, 1999]

de Castro (2002) proposed a standard CSA, named *CLONALG* (Figure 2.10). CLONALG is a type of *evolutionary algorithm* (EA). It has all the steps involved in an EA; reproduction and genetic variation, affinity (fitness) evaluation, and selection.

1. *Initialization: randomly initialize a population of attribute string (immune cells)*
2. *Population loop: for each antigen, do:*
   - *2.1 Selection: select those cells whose affinities with the antigen are greater.*
   - *2.2 Reproduction and genetic variation: generate copies of the immune cells; the better each cell recognizes the antigen, the more copies are produced. Mutate (perform variations) each cell inversely proportional to their affinity; the higher the affinity, the smaller the mutation rate.*
   - *2.3 Affinity evaluation: evaluate the affinity of each mutated cell with antigen.*
3. *Cycle: repeat Step 2 until a given convergence criterion is met.*

Figure 2.10: CLONALG [de Castro, 2002]

Kim and Bentley (2002) investigated the behavior of *DynamiCS*, a dynamic CSA; designed to have the properties of self-adaptation. DynamiCS was created to tackle the

difficulties of anomaly detection in changing environments. Doyen et al. (2003) proposed *Clonal Selection and Affinity Maturation Algorithm* (Figure 2.11) for scheduling problems. The possible schedules are represented by integer-valued strings of length *n*. The *n* elements are the jobs which will be sequenced, so the strings are composed of permutations of *n* elements. Those strings are accepted as antibodies of the AIS.

```
Create a population of z antibodies (z – size of antibody population): x=0
  For each generation, do: x = x + 1
    For each antibody, do
      Decode the antibody
      Determine the makespan (affinity) of antibody
      Calculate the selection probability (rate of cloning)
      Cloning (generate copies of antibodies)
      For each generated clone, do
        Inverse mutation (generate a new string)
        Decode the new string
        Calculate the makespan of the new string
        If makespan (new string) < makespan (clone,) then clone = new string
        else do pairwise interchange mutation (generate a new string)
        Decode the new string
        Calculate the makespan of the new string
        If makespan (new string) < makespan (clone), then clone = new string
        else clone = clone
        Antibody = clone
  If x = M (M = 25% of the generation steps)
    Eliminate worst N number of antibodies in the population
    Create new antibodies of N number (N=25% of the population size)
    Change the new created ones with the eliminated ones
  End if
While stopping criterion = false.
```

Figure 2.11: Clonal Selection and Affinity Maturation Algorithm [Doyen et al., 2003]

White and Garrett (2003) examined the CLONALG and suggested that it is suitable for pattern recognition. One potentially negative feature of CLONALG is that it fails to capitalize on the information generated by each cloned population. Once a candidate memory cell has been selected, the remaining mutated clones are discarded, even though it may contain a number of high affinity candidates. By preserving a larger proportion of the matured population, the algorithm could build from a stronger base of high affinity matches and should theoretically reach an optimal solution in fewer generations. However, this also introduces a new danger of becoming stuck on a local minimum as the population becomes increasingly narrow in its focus. The replacement

phase that introduces new randomly generated antibodies into the population should prevent this. The amended CLONALG, *CLONCLAS*, is given in Figure 2.12.

1. *Randomly generate an initial population of antibodies Ab. This is composed of two subsets Ab_m (memory population) and Ab_r (reservoir population).*
2. *Create a set of antigenic patterns Ag.*
3. *Select an antigen Ag_i from population Ag.*
4. *For G generations*
   a) *Carry out step 4-7 of CLONALG.*
   b) *Replace the antibodies in Ab_r with the same number of antibodies from the sorted population C_i*.*
   c) *Remove those antibodies with low affinity in the population Ab_r and replace with new randomly generated members.*
5. *Return to step 3 until all antigens have been presented.*

Figure 2.12: CLONCLAS [White and Garrett, 2003]

Coello et al. (2004) introduced a new AIS approach to solve *job-shop scheduling problems* (JSSP). The approach uses concepts from clonal selection theory (extending ideas from CLONALG), and adopts a permutation representation that allows repetitions. Garain et al. (2006) presented an application of a CSA for recognition of handwritten Indic numerals. In particular, a two-phase CSA implementing a retraining scheme was proposed. Suliman et al. (2006) presented the application of AIS for online voltage stability evaluation that could be used as early warning system to the power system operator. The key features of the proposed method are the implementation of clonal selection principle that has the capability in performing pattern recognition task.

Zhu et al. (2007) proposed an improved CSA for *Traveling Salesman Problems* (TSP). In addition to receptor editing with clonal selection, a self-crossover operator is also implemented to improve performance. Gao et al. (2008b) proposed a *Simulated Annealing PolyCSA* (SA-PCSA) which combines the PCSA and the traditional SA for TSP. By using PCSA, the solution space can be explored more efficiently. Moreover, by using SA the probability of local minimum can be reduced because of the introduction of jump probability which can be adjusted by controlling the temperature.