Aisimam - An Artificial immune system based intelligent multiagent model

Srividhya Sathyanath
AISIMAM – AN ARTIFICIAL IMMUNE SYSTEM BASED INTELLIGENT MULTIAGENT MODEL

By

SRIVIDHYA SATHYANATH

Thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

Approved by

Thesis Advisor
Dr. Ferat Sahin

Thesis Committee
Dr. Raghuveer. M. Rao

Thesis Committee
Dr. Vincent Amuso

Department Head
Dr. Robert. J. Bowman

ROCHESTER INSTITUTE OF TECHNOLOGY
KATE GLEASON COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING, ROCHESTER, NEW YORK
NOVEMBER 2002
AISIMAM – AN ARTIFICIAL IMMUNE SYSTEM BASED INTELLIGENT
MULTIAGENT MODEL

I, Srividhya Sathyanath hereby grant the permission to the Wallace Library of the
Rochester Institute of Technology to reproduce my thesis in whole or in part. Any
reproduction will not be for commercial use or profit.

Date: 04/12/02

Author

11/21/02 Dept of Electrical Engineering
Acknowledgments

I would like to express my sincere thanks and gratitude to my graduate advisor, Dr. Ferat Sahin for his relentless help, support, and motivation. Firstly, I would like to thank him for giving me an opportunity to work as a research assistant during my course of study. Secondly, I would like to thank him for providing me with financial support throughout my course at RIT. I am truly grateful to him for his understanding and encouragement during my hard times. I would like to thank him for sharing his knowledge and experiences as a friend and a guide that made my research work very comfortable. I am sure that he would truly remain an inspiration for all future generations.

The Electrical Engineering Department at Rochester Institute of Technology has been a vital cornerstone for me in my career. I believe my life at R.I.T has given me the knowledge, skills and aspirations to scale different horizons. My special thanks and appreciation to the former Head of the Department, Dr. Raman Unnikrishnan for his fatherly advice and guidance. My deepest thanks to Dr. Robert Bowman, for having me as a part of the RIT culture.

My heart-felt gratitude also goes to all the professors in the Electrical Engineering department. My special thanks to Dr. Raghuveer. M. Rao and Dr. Vincent Amuso for being my thesis committee and for guiding me to complete my thesis work. I would like to thank Mr. Kenneth Snyder and Mr. James Stefano for providing me with additional privileges to use the computer labs without which this thesis would not have been possible. I would also like to thank the staff, Mrs. Florence Layton, Mrs. Patti Vicari and Mrs. Jill Lewis whose help with a smiling face made my stay comfortable.
My sincere love and thanks to my family for imparting me the right knowledge to be a good human being. All my successes are yours and you all will truly remain a cherished part of my life forever. This thesis is dedicated to all of you.
Accomplishments

Publications arising from this graduate thesis


Results of the Thesis

- Selected as the Finalist for the Conference Best Student Paper Award, awarded by the IEEE Systems, Man and Cybernetics Conference held at Arizona, 2001.

- Nominated for the Conference Best Student Paper Award, awarded by the IEEE Systems, Man and Cybernetics Conference held at Hammamet, Tunisia, 2002.

AN ARTIFICIAL IMMUNE SYSTEM BASED INTELLIGENT MULTIAGENT SYSTEM MODEL (AISIMAM)

By

SRVIDHYA SATHYANATH

Master of science in Electrical Engineering

Abstract

The goal of this thesis is to develop a biological model for multiagent systems. This thesis explores artificial immune systems, a novel evolutionary paradigm based on the immunological principles. Artificial Immune systems (AIS) are found to be powerful to solve complex computational tasks.

The main focus of the thesis is to develop a generic mathematical model that uses the principles of the human immune system in multiagent systems (MAS). The components and properties of the human immune system are studied. On understanding the concepts of AIS, a literature survey of multiagent systems is performed to understand and compare the multiagent concepts and AIS concepts. An analogy between the immune system parameters and the agent theory was derived. Then, an intelligent multiagent model named AISIMAM is derived. It exploits several properties and features of the immune system in multiagent systems. In other words, the intelligence of the immune systems to kill the antigen and the characteristics of the agents are combined in the model. The model is expressed in terms of mathematical expressions.

The model is applied to a specific application namely the mine detection and defusion. The simulations are done in MATLAB that runs on a PC. The experimental
results of *AISIMAM* applied to the mine detection problem are discussed. The results are successful and shows that *AISIMAM* could be an alternative solution to agent based problems.

Artificial Immune System is also applied to a pattern recognition problem. The problem experimented is a color image classification problem useful in a real time industrial application. The images are those of wooden components that need to be classified according to the color and type of wood. To solve the classification task, a simple negative selection and genetic algorithm based *AIS* algorithm was developed and simulated. The results are compared with the radial basis function approach applied to the same set of input images.
# Table of Contents

Acknowledgments ........................................................................................................ iii  
Accomplishments ........................................................................................................ v  
Abstract ......................................................................................................................... vi  
Table of Contents ......................................................................................................... viii  
List of Figures ............................................................................................................... xv  
List of Tables .............................................................................................................. xvii

1. Introduction .................................................................................................................. 1-1  
   1.1 Motivation ............................................................................................................... 1-1  
   1.2 Objectives and Contributions .............................................................................. 1-2  
      1.2.1 Tasks involved .............................................................................................. 1-2  
      1.2.2 Contributions .............................................................................................. 1-2  
   1.3 Outline of the Thesis ............................................................................................. 1-4  

2. The Human Immune System .................................................................................... 2-1  
   2.1 Introduction .......................................................................................................... 2-1  
   2.2 The Human Immune System .............................................................................. 2-1  
   2.3 Importance of Immune System .......................................................................... 2-1  
   2.4 Anatomy of the Human Immune System .......................................................... 2-2  
   2.5 Components of the Immune System .................................................................. 2-3  
      2.5.1 External Components (Non-self Cells – Antigens) ..................................... 2-4  
      2.5.2 Internal Component (Self-cells – Lymphocytes) ........................................ 2-4
2.5.3 Phagocytes ................................................................. 2-6
2.5.4 T Cells ................................................................. 2-6
2.6 Types of Immune System .................................................. 2-8
2.7 Types of Immunity ......................................................... 2-9
  2.7.1 Innate Immunity ................................................... 2-9
  2.7.2 Acquired Immunity ................................................ 2-9
2.8 Properties of the Immune System ......................................... 2-10
  2.8.1 Affinity Maturation and Receptor Editing .................... 2-11
  2.8.2 Clonal Selection and Expansion Principle .................... 2-11
  2.8.3 Immune Memory .................................................. 2-14
  2.8.4 Reinforcement Learning ........................................ 2-16
2.9 Jeme’s Idiotypic Network Theory ....................................... 2-17
  2.9.1 Need for the Immune Network .................................. 2-19
2.10 Self Organization ........................................................ 2-19
2.11 Self/non-self discrimination – Positive/Negative selection .......... 2-20
  2.11.1 Positive Selection .............................................. 2-20
  2.11.2 Negative Selection ............................................. 2-20
2.12 Types of Immune Responses ............................................. 2-22
  2.12.1 Humoral Immune Response .................................... 2-22
  2.12.2 Cell Mediated Response ....................................... 2-22
### 3. Artificial Immune Systems and their Applications

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Introduction to Artificial Immune Systems</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2</td>
<td>Definitions of Artificial Immune Systems</td>
<td>3-1</td>
</tr>
<tr>
<td>3.3</td>
<td>Artificial Immune Systems and Their Applications</td>
<td>3-2</td>
</tr>
<tr>
<td>3.4</td>
<td>Immune System Models</td>
<td>3-3</td>
</tr>
<tr>
<td>3.5</td>
<td>Negative Selection Algorithm</td>
<td>3-5</td>
</tr>
<tr>
<td>3.6</td>
<td>Affinity Maturation</td>
<td>3-7</td>
</tr>
<tr>
<td>3.7</td>
<td>Immune Memory</td>
<td>3-8</td>
</tr>
<tr>
<td>3.8</td>
<td>Need for a Global Model</td>
<td>3-11</td>
</tr>
</tbody>
</table>

### 4. Multiagent Systems

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Introduction to Multiagent Theory</td>
<td>4-1</td>
</tr>
<tr>
<td>4.2</td>
<td>Agent Definitions</td>
<td>4-2</td>
</tr>
<tr>
<td>4.3</td>
<td>Structure of Intelligent Agents</td>
<td>4-4</td>
</tr>
<tr>
<td>4.4</td>
<td>Multiagent Systems</td>
<td>4-5</td>
</tr>
<tr>
<td>4.5</td>
<td>Characteristics of Agents</td>
<td>4-6</td>
</tr>
<tr>
<td>4.6</td>
<td>Types of Environments</td>
<td>4-7</td>
</tr>
<tr>
<td>4.7</td>
<td>Central Issues in MAS</td>
<td>4-8</td>
</tr>
<tr>
<td>4.8</td>
<td>Performance Measures for Multiagent Systems</td>
<td>4-9</td>
</tr>
<tr>
<td>4.9</td>
<td>Examples of Agents</td>
<td>4-9</td>
</tr>
</tbody>
</table>
4.10 Multiagent Systems Applications ............................................. 4-10

5. AISIMAM - Artificial Immune System based Intelligent Multi Agent Model .5-1

5.1 Introduction ........................................................................... 5-1

5.2 AISIMAM - Artificial Immune System Based Intelligent Multiagent Model .5-1

5.3 Comparison of AIS and Multiagent Systems ................................ 5-2

5.3.1 Pattern Recognition ............................................................ 5-5

5.3.2 Binding Process ................................................................. 5-5

5.3.3 Activation Process .............................................................. 5-6

5.3.4 Post Activation Process ...................................................... 5-6

5.3.5 Post Processing ................................................................. 5-6

5.3.6 Agent Network ................................................................. 5-7

5.4 AISIMAM – Operational Scheme and Mathematical Representation .......... 5-10

5.4.1 AISIMAM - Parameter Definitions ...................................... 5-10

5.4.2 AISIMAM - Algorithm ......................................................... 5-11

5.4.3 Need for a Mathematical Representation ............................. 5-14

5.5 New Aspect of the Work ......................................................... 5-16

6. Applications of Artificial Immune Systems ........................................ 6-1

6.1 Introduction to Mine Detection Problem ..................................... 6-1

6.1.1 Importance of Mine Detection ............................................ 6-1

6.1.2 Types of Mines and Mine Detection Methods ....................... 6-2
6.1.3 Robots in Mine Detection ........................................6-3
6.1.4 Agent Theories in Mine Detection ..............................6-5
6.1.5 Motivation for the Mine Detection Problem ..................6-5
6.2 Application of AISIMAM to a Mine Detection Problem ......6-6
6.2.1 Parameter Definitions ...........................................6-6
6.2.2 Mine Defusion ....................................................6-8
6.3 Pseudo Code For The Mine Detection Problem ................6-10
6.3.1 Simulation Details ...............................................6-11
6.3.2 Experimental Results and Analysis ............................6-14
6.4 Implementation of Binding Period Resolves Freezing .......6-19
6.5 An AIS based Color Image Classification Problem in a Real Time Industrial Application ........................................6-20
6.6 Introduction to Classification ......................................6-20
6.7 A Pattern Recognition System ....................................6-21
6.8 Class Descriptions ..................................................6-22
6.9 Classification Theory ...............................................6-23
6.10 Genetic Algorithms ................................................6-24
6.11 Description of Genetic Algorithm ...............................6-25
6.12 Crossover ................................................................6-26
6.12.1 Mutation ..........................................................6-28
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.12.2</td>
<td>Elitism</td>
<td>6-28</td>
</tr>
<tr>
<td>6.13</td>
<td>An AIS Approach to a Color Image Classification Problem</td>
<td>6-28</td>
</tr>
<tr>
<td>6.14</td>
<td>Color Image Classification Problem</td>
<td>6-29</td>
</tr>
<tr>
<td>6.15</td>
<td>Need for Color Image Classification</td>
<td>6-30</td>
</tr>
<tr>
<td>6.16</td>
<td>An AIS based Image Classifier</td>
<td>6-33</td>
</tr>
<tr>
<td>6.16.1</td>
<td>Training using Matching Algorithm and Negative Selection</td>
<td>6-35</td>
</tr>
<tr>
<td>6.16.2</td>
<td>AIS based Genetic Algorithm</td>
<td>6-38</td>
</tr>
<tr>
<td>6.17</td>
<td>Threshold Setting</td>
<td>6-41</td>
</tr>
<tr>
<td>6.18</td>
<td>Classification Results</td>
<td>6-44</td>
</tr>
<tr>
<td>6.19</td>
<td>Summary of Results</td>
<td>6-46</td>
</tr>
<tr>
<td>7.</td>
<td>Future Work</td>
<td>7-1</td>
</tr>
<tr>
<td>7.1</td>
<td>Introduction</td>
<td>7-1</td>
</tr>
<tr>
<td>7.2</td>
<td>Improvements in AISIMAM</td>
<td>7-1</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Immune Memory</td>
<td>7-1</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Reinforcement Learning in Immune System</td>
<td>7-2</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Self-organization</td>
<td>7-3</td>
</tr>
<tr>
<td>7.2.4</td>
<td>Immune Network</td>
<td>7-4</td>
</tr>
<tr>
<td>7.3</td>
<td>Issues in Mine Detection Problem</td>
<td>7-4</td>
</tr>
<tr>
<td>7.4</td>
<td>AIS based Image Classification Problem</td>
<td>7-5</td>
</tr>
<tr>
<td>8.</td>
<td>Conclusions</td>
<td>8-1</td>
</tr>
</tbody>
</table>
9. References ................................................................. 9-1
10. Appendix A ............................................................... 10-1
List of Figures

Figure 2-1 Anatomy of the human immune system [11] ........................................... 2-3
Figure 2-2 Structure of the antigen, B cell, lock and key mechanism ........................ 2-5
Figure 2-3 Structure of the antigen with epitopes and B cell with receptors ................ 2-6
Figure 2-4 Types of immunity and their respective components [11] .......................... 2-10
Figure 2-5 Affinity maturation [11] ........................................................................... 2-11
Figure 2-6 Clonal selection and expansion ................................................................ 2-14
Figure 2-7 Primary and Secondary responses [11] ..................................................... 2-15
Figure 2-8 Representation of cross reactive response ................................................. 2-16
Figure 2-9 Jerne's immune network .......................................................................... 2-18
Figure 2-10 Self/Non-self discrimination .................................................................... 2-21
Figure 2-11 Humoral and Cell mediated response ...................................................... 2-23
Figure 2-12 Representation of the human immune system ......................................... 2-24
Figure 3-1 Shape space model .................................................................................. 3-4
Figure 3-2 Negative selection algorithm [15] ............................................................ 3-6
Figure 3-3 Flowchart representation of the human immune system ............................ 3-10
Figure 4-1 An Agent in its environment [60] .............................................................. 4-3
Figure 5-1 Representation of AISIMAM ..................................................................... 5-4
Figure 5-2 Processes in AISIMAM ............................................................................ 5-5
Figure 5-3 Flowchart representation of AISIMAM ..................................................... 5-9
Figure 6-1 Flowchart representation of the mine detection problem ......................... 6-9
Figure 6-2 Initial locations of the mines, robots and the locations after 2 iterations ....... 6-13
Figure 6-3 Four robots have circled one mine and the environment is updated .......... 6-13
Figure 6-4 Average rate of convergence vs. variation in comm circle ....................... 6-15
Figure 6-5 Average rate of convergence vs. variation in sensory and comm circle ...... 6-15
Figure 6-6 Average rate of convergence vs. sen & comm. circles & mines/robots ...... 6-16
Figure 6-7 Average rate of convergence vs. variation in robots and mines ............... 6-17
Figure 6-8 Average rate of convergence vs. variation in robots and mines ............... 6-18
Figure 6-9 Rate of convergence vs. variation in sensory and comm. circle ............... 6-18
Figure 6-10 A Pattern classification system [58] ....................................................... 6-22
Figure 6-11 Example of one point crossover ................................................................. 6-26
Figure 6-12 Example of one point crossover ................................................................. 6-27
Figure 6-13 Example of two-point crossover ................................................................. 6-27
Figure 6-14 Uniform crossover ...................................................................................... 6-28
Figure 6-15 A color image classification system ............................................................. 6-30
Figure 6-16 A color and species classification system ..................................................... 6-31
Figure 6-17 Structure of the images ................................................................................. 6-32
Figure 6-18 Eight classes of wood used to train the AIS .................................................... 6-34
Figure 6-19 Binary representation of an image ............................................................... 6-36
Figure 6-20 An example of the matching algorithm ......................................................... 6-37
Figure 6-21 Initial population and four-bit are crossed-over in the eight bits ................ 6-41
Figure 6-22 Flowchart representation of the AIS based image classification algorithm 6-43
List of Tables

Table 3-1 Application of artificial immune systems and their relation to the properties of the immune system..........................................................3-3
Table 5-1 Analogy of immune system features and AISIMAM ..............................5-8
Table 6-1 An example of information vector of mines and robots ............................6-12
Table 6-2 An Example of the environment vector ..................................................6-12
Table 6-3 Fitness values in six iterations and the corresponding images ....................6-41
Table 6-4 Negative selection based and image classification algorithm ....................6-43
Table 6-5 AIS and genetic algorithm based image classifier ..................................6-43
Table 6-6 Percentage of correct classification using the Euclidean distance method (Average and Minimum methods) ..................................................6-46
Table 6-7 Results of AIS and genetic algorithm based image classification algorithm 6-46
1. Introduction

1.1 Motivation

Biological systems have been an inspiration to scientists and engineers to solve complex computational and information processing problems. Imitating biological systems to solve complex engineering problems has been an area of vast research [1, 2].

Biological systems generally outperform advanced machines and are effective in solving many problems. These systems have remarkable capabilities and unique mechanisms with which they process information and find solutions [3]. For instance, the brain processes the data at a speed million times greater than the fastest computer [4]. Similarly, for face recognition, the human brain takes just four to five cycles compared to billions of cycles in a computer system (all the neurons are updated/fixed in one step/cycle). Storing a face requires mega bytes in a computer as opposed to effective storage in brain [4]. In addition to efficient information processing capabilities, the biological systems also possess learning mechanisms [5, 6].

There has been a growing interest in imitating the biological systems to find effective problem solving techniques. Genetic algorithms derived from the principles of genetics, neural networks derived from brain - nervous systems or neurology and cellular engineering based on cell biology are some of the biologically motivated evolutionary algorithms [1, 2].

This thesis concentrates on a new biologically motivated approach called artificial immune systems, based on human immune systems. Over the last few years there has been an increasing interest in the area of artificial immune systems that imitate the natural
immune system. The natural immune system has been studied to develop tools for solving complex problems due to their properties such as significant information processing capabilities, learning mechanisms and memory [1, 2, 5].

1.2 Objectives and Contributions

The objective of this research is to develop an intelligent multiagent model using immune system concepts. The research focuses on the investigation of the biological concepts of the immune system, the computational abilities, and the application of immune system concepts to multiagent systems.

1.2.1 Tasks involved

The tasks involved in this research are stated below.

- A literature survey is performed to understand the principles and features of the human immune system.
- The computational aspects and applications of the immune system known as the artificial immune systems are studied.
- Genetic algorithms are studied to improve the AIS based classification algorithm.
- Multiagent systems are studied in the context of the understanding the resemblance of artificial immune systems.

1.2.2 Contributions

Literature survey shows that algorithms are developed based on the properties of the immune system [1, 2, 3, 5, 6]. However, they work for only specific applications. Similarly, existing models do not generalize the description of the properties used in the immune system. In that aspect, the major contributions of this research include the following.
a) Artificial Immune System is found to be a distributed and sophisticated information processing system, therefore application of AIS to multiagent systems is researched. This led to the development of a mathematical model of artificial immune systems based on intelligent multiagent model (AISIMAM). Limited literature of application of AIS on agent based systems reveals that the mathematical model could be a valuable addition to the literature.

b) Incorporation of all the properties responsible for humoral response of the immune system into AISIMAM.

c) The model is generic because the features and principles are defined by generic functions. Generalized function descriptions can be useful in applying the model to diverse applications.

d) The application of AIS to defense is a recent thought by the defense community (Gene Peresich of Boeing systems, 2001). Application of AISIMAM to a mine detection problem is experimented towards that end. The results suggest that the proposed model could be an alternative solution to a mine detection problem.

e) Mine detection problem is simulated. The motivation for choosing this application is that micro mobile robots, with sensing, communication and decision-making capabilities can be built for mine detection in the future. Such custom built robots can replace every agent in the simulation to solve a mine detection problem. The algorithm is found to be robust and effective for changes in different parameters such as the population of the mines and robots and the area to be explored. The results are successful and prove that the model can be an alternative solution to other agent-based applications.
f) An AIS and genetic algorithm based classifier is also developed for a color image classification problem. The algorithm uses negative selection principle or the self-nonself discrimination property of the immune system. The classification results are compared using the other classification procedure namely the radial basis function networks (RBFN) [8, 9] using the same data. The biological based classification procedure showed promising results. It is inferred from the experiment that successful results are expected on inclusion of some more properties of the immune system.

1.3 Outline of the Thesis

The following chapters describe the literature survey, the problem definition for the AIS based color image classification algorithm, the proposed AIS based intelligent multiagent model, the mine detection problem and the simulation results.

Chapter 2 contains a literature review of the human immune system covering the components, properties and functions of the immune system. This essentially provides a background for the computational aspects of immunology that has evolved into artificial immune systems (AIS), also called immunological computations [1].

Chapter 3 briefly summarizes the computational models developed with the properties of the immune system. In spite of the short span of the emergence of artificial immune system, it is applied to many application areas. Some references are provided for the applications of AIS. The goal of this chapter is to state the existing models and the need for a global model.
Chapter 4 investigates agent definitions, fundamental concepts on multiagent systems, characteristics of the agents defining their behavior and different kinds of environments. Some applications of the multiagent systems are also stated.

Chapter 5 describes the AIS based intelligent multiagent model. A comparison of the AIS concepts and multiagent concepts are provided. A conceptual framework stating the processes involved in the model is explained. Later sections provide the mathematical expressions involved in the model.

Chapter 6 describes two applications of AIS. The first problem is the application of AISIMAM to a mine detection and defuse problem. The chapter begins with an introduction to mine detection problem, importance of the mine detection problem, different types of mines and existing methods of mine detection. Later sections explain the agent concepts in mine detection and application of robots as a possible solution for defusion of mines. Application of AISIMAM to mine detection and the simulation results are explained.

The second problem is an AIS approach to a pattern recognition problem. This is an AIS based color image classification problem useful in a real time industrial application. Genetic algorithms are also used to improve the solution. This chapter begins with the literature review on classifier theory, need for the image classification task and a discussion of genetic algorithms. Later sections of the chapter deal with the AIS and genetic algorithm approach to a color image classification with the simulation results.

Chapter 7 discusses the scope for future work in improving the AISIMAM, the mine detection application and the AIS based color image classification problem.
Chapter 8 discusses the conclusions derived out of this research. Chapter 9 lists the references used for this thesis. Appendix A consists of a CD-ROM that contains the source code for the color image classification problem and mine detection problem simulated in MATLAB 6.1.
2. The Human Immune System

2.1 Introduction

This chapter provides a literature review on the human immune system. This provides the background for the development of the proposed artificial immune systems based intelligent multiagent model.

2.2 The Human Immune System

Immunology is the study of immune system and the study of defense mechanisms that confer resistance against diseases [10].

The human immune system is a complex system of cells and molecules distributed throughout the body that can provide us with a natural defense mechanism. The main goal of the human immune system is to protect the internal components of the human body. By defense, it is meant that the immune system protects our body from infectious agents such as viruses, bacteria, fungi, and other parasites [3].

This chapter explains how our immune system attains immunity and protects our body from infectious diseases. The following topics discuss the importance of the human immune system, components of the immune system, properties and functioning of the immune system.

2.3 Importance of Immune System

The natural immune system is a very complex system with several mechanisms for defending against infectious agents entering our system. These defense mechanisms have the ability to detect foreign substances. In immunology, these foreign substances are called antigens. The immune system reacts to the antigens by producing antibodies that attack these antigens. In addition to reacting to the foreign agents that comes from 11/21/02 Dept of Electrical Engineering
outside and inside the living system, it also plays an important role to maintain its own system against dynamically changing environments [11].

The immune system is a sophisticated information processor. It learns and remembers the type of antigen that it has fought. In other words, it learns to recognize patterns and possesses memory [5, 6, 12]. The immune system contains thousands of different cells interconnected in a network that communicate with each other. The immune system is a distributed system with no central controller and the control is distributed throughout our body via its constituent cells and molecules [13]. The human immune system recognizes the antigenic patterns, communicates, learns and memorizes these patterns and eventually kills the antigens. Performing such complex tasks are some of the properties of the immune system that have inspired researchers to imitate these properties in various applications apart from the field of medicine.

The following paragraphs describe the anatomy of the immune system and some of the important properties and the principles of the immune system.

2.4 Anatomy of the Human Immune System

The immune system contains tissues and organs called the lymphoid organs throughout the body. Figure 2-1 shows the anatomy of the human immune system.

The following are the main functions performed by the lymphoid organs [11].

- **Bone Marrow**: It is the soft tissue in the longest bone of the human body that generates the B cells.
• **Thymus**: Some of the cells from the bone marrow transfer to *thymus*, gets multiplied and transforms into *T* cells that perform different functions from the *B* cells.

• **Lymph nodes**: All the lymphatic vessels converge at this point and each lymph node stores *B* and *T* cells.

• **Tonsils and Adenoids**: These are special lymph nodes with immune cells that protect the respiratory system from the foreign agents.

### 2.5 Components of the Immune System

The basic components of the immune system are the white blood cells, called *self-cells* or *lymphocytes* in immunological terms. These specialized cells are classified into two types namely, the *B* lymphocytes and *T* lymphocytes. The external components to the immune system are antigens also called the *non-self cells*, as they are foreign substances...
to the body [10, 5]. Detailed explanation of the structure and functioning of these two components are described in the following sections.

2.5.1 External Components (Non-self Cells – Antigens)

*Antigens* are foreign elements to the human system that cause disturbances to the normal functioning of the body. Therefore they are called *non-self* cells [10]. Although it is external to the human body, it can be treated as part of the immune system as it triggers the human system and invokes an immune response.

Antigens typically have several different types of *receptors* on their surface. Receptors are portions of the antigenic cell. These receptors are called *epitopes* [11]. Epitopes on antigens have different shapes that can be recognized by several different antibodies. Antigens have multiple receptors and therefore can be called *poly-specific*.

2.5.2 Internal Component (Self-cells – Lymphocytes)

In immunological terms, the main component of the immune system is the *lymphocytes* or white blood cells. These specialized cells are classified into two types. They are *B-lymphocytes* produced by the bone marrows and the *T* cells that develop in bone marrow and mature in *thymus*. *B* cells and *T* cells together are defined as the *self*-cells, representing that they belong to our own body [12, 3]. The following section deals with the structure of the *B* cell and the antibodies.

2.5.2.1 *B* Cells and Antibodies

The major responsibility of the *B* cells is the secretion of the antibodies (*Ab*) as a response to antigens (*Ag*). Roughly there are $10^7$ *B* lymphocytes in the human body. Each of these *B* cells or lymphocytes has distinct chemical structures and generates
antibodies that are Y shaped receptor molecules on the surface of the B cell [14]. Figure 2-2 is a graphical representation of the structure of the antigen and the B cells.

Each B cell is known to have a single type of antibody because of which it is characterized as mono-specific [12]. Antibodies have regions called idiotopes and paratopes. The paratopes and idiotopes are collectively called idiotypes. Paratopes vary their shape to identify the antigenic pattern and therefore they are also called V region representing the variable regions. This flexibility in varying the shape of the antibody according to the antigenic shape provides the ability for the B cells to kill the antigens and protect our system [11, 14, 15].

The role of the B cells is to generate antibodies of complementary match to recognize and bind the antigen. Complementary match means the generation of a structure that fits well with the antigenic epitope thereby recognizing the antigen. This complementary match acts like a lock and key mechanism where the idiotypes of the B cells lock themselves with the antigenic epitopes as a key.

![Figure 2-2 Structure of the antigen, B cell, lock and key mechanism](image-url)
Figure 2-3 illustrates the $B$ cell with a single receptor type and the antigen with multiple epitopes. Since each $B$ cell is known to have a single type of antibody or monospecific, and since the antigens have different epitopes or poly specific, it requires multiple $B$ cell interactions, cooperation and coordination to bind and kill the antigen. The following paragraph briefly explains the $T$ cells and *phagocytes* in the immune system.

![Diagram of antigen and B cell with receptors](image)

**Figure 2-3 Structure of the antigen with epitopes and B cell with receptors.**

### 2.5.3 Phagocytes

These are white blood cells capable of digesting microorganisms and antigenic particles. Some of them can also present the antigens to the lymphocytes, thus being called as *antigen-presenting* cells (APC). They are helpful at the beginning of the immune response. Some of the important phagocytes are the *macrophages* and *monocytes* [11].

### 2.5.4 T Cells

$T$ cells are blood cells that are generated in the bone marrows and travel to the thymus and mature. Since they mature in thymus these cells are called $T$ cells [11]. Functions of the $T$ cells include regulation of other cells and directly attacking the antigens.

$B$ cells have direct contact with the antigen when it interacts with them. They have receptors on their surface, which can recognize antigens invading the human body.
But T cells can bind to the antigen only after it is processed and presented by the other cells. That is, the macrophages roam around the body, digesting the antigens and fragmenting into antigenic peptides [11]. These fragments are joined to form major histo compatibility complex molecules (MHC) that are presented on the surface of the cells. T cell receptors can recognize the antigen associated with the MHC molecules. Once the T cells recognize and interact with an antigen MHC molecule complex, the cell secretes various growth factors known collectively as cytokines. They activate the B cells and T cells to kill the antigen. This is the behavior exhibited by the immune system to protect our body against the antigens. However this process is divided amongst the T cells. According to the operation they perform, T cells can be categorized into three [11]. They are

- **Helper T cells** *(Th)*: These activate the B cells, other T cells, and other necessary cells to initiate the immune response.

- **Killer T cells** *(cytotoxic cells)*: They eliminate viruses or foreign substances, by injecting noxious chemicals on the other cells, perforating the surface and destroy them.

- **Suppressor T cells**: These cells take care of *intra self cell* reactions. By suppressing the actions of the self-cells on the other self-cells, auto-immuno diseases are avoided.

### 2.6 Types of Immune System

There are two types of immune system, depending on the way it reacts to the antigen; the *vertebrate* and *invertebrate immune* systems [12, 14].
- The vertebrate immune system involves lymphocytes that are antigen specific. That is, they have distinct receptors to interact with different antigens.

- The invertebrate immune systems are systems that consist mainly of *phagocytes*, which have a non-specific immune response. That is, they have no distinct receptors for specific antigens and hence the cells try to kill any antigen that enters the body [3].

The vertebrate immune system is more complex than the invertebrate immune system and the two properties that characterize the difference are memory and specificity [3, 12]. Some properties of the vertebrate immune systems are

- Use *feature extraction* to determine the unique nature of the antigen.
- *Learn* to recognize new patterns of the antigens.
- Work as a distributed *pattern recognizer*.
- Use content addressable *memory* to retrieve known patterns/antigens.
- Use specific proliferation and self-replication for quick recognition and response.
- Eliminate or neutralize the effect of antigens in a systematic pattern.

In essence, with these properties, the immune system has the ability to identify and kill the antigen in an efficient manner and protect our body.

### 2.7 Types of Immunity

There are two types of immunity in the human system namely *innate immunity* and *acquired immunity*. Innate and adaptive immunity are explained in the following sections.
2.7.1 Innate Immunity

Innate immunity is the natural resistance of the body to the foreign antigens. This natural resistance is an inherent property of the human immune system. This is not directed towards any specific invaders that enter the body, but against any antigen that enters the body. This is a non-specific approach and doesn’t get modified by repeated exposure. There are a number of external and internal lines of defense in innate immunity [14]. Some examples are lysozymes in tears and skin as a barrier. Invertebrate immune system uses a non-specific immune response with the help of phagocytes to attack the antigens.

2.7.2 Acquired Immunity

In contrast to the innate immunity, acquired immunity is directed towards specific invaders and is modified by exposure to such foreign antigens. Acquired immunity is also called adaptive immunity [11]. The acquired immune system uses a specific immune response to antigens. A vertebrate immune system employs acquired immunity. Adaptive immunity has the advantages of immunological memory.

Once the antigen is eliminated, the lymphocytes change into memory cells. These cells will recognize the same foreign element rapidly when it invades the body again and kill it before it causes any harm to the body [11, 16]. Immunological memory will be explained in section 2.8.3. Figure 2-4 gives a picture on the types of immunity. The following sections explain the important properties of the immune system.
2.8 Properties of the Immune System

As explained in the earlier sections, the main function of the immune system is to kill the antigen. It is interesting to note that this common goal of the system is handled by the individual components of the immune system. At the same time, they also work collectively to perform the common task. Some properties of the immune system are clonal selection and expansion, immune memory, reinforcement learning, self-organization, affinity maturation, positive and negative selection (Self/Non-self discrimination) and Jerne’s idioptypic network [1, 2, 3, 5, 13, 17]. The following section explains these properties in detail.

2.8.1 Affinity Maturation and Receptor Editing

When the antigen attacks the body, the receptors on the B cells frequently suffer a process called hyper mutation that edits the receptors of the B cells, and therefore the antibody response improves after repeated immunization [18]. In the process of receptor editing, paratopes match with the epitopes of the antigen to recognize the pattern of the antigen.
Because paratopes change their shapes according to the shape of the epitope, this recognition or matching is achieved and the antigen is killed. The amount of matching and the strength of the bond depend on the chemical structure of the lymphocytes [15]. This property is called affinity maturation. This is also defined as dynamic protection [1]. Dynamic protection increases the coverage provided by the immune system over time. Figure 2-5 shows affinity maturation versus the immune response. It can be seen that the affinity grows higher with the growth of the pattern of the antibody.

![Figure 2-5 Affinity maturation [11]](image)

**2.8.2 Clonal Selection and Expansion Principle**

Millions of varieties of lymphocytes are supposed to be produced by the immune system. In contrast to the invertebrate immune systems the vertebrate systems are supposed to have a specific immune response that depends on the antigen [16]. The specificity of the immune response is determined by the clonal selection principle. Clonal selection is followed by the clonal expansion.
2.8.2.1 Clonal Selection

In the first stage, once the B cell is activated by an antigenic stimulus, B cells proliferate and secrete its receptor molecules. Not all the B cells get cloned. *Clone* is a single cell or cells that are exact reproduction of the same cell. The immune system clones only the cells that are found useful to create the antibodies to kill the antigen. This process is described as *clonal selection*, because the antigen selects the B cells to be cloned indirectly.

2.8.2.2 Clonal Expansion

The second stage is called *clonal expansion*. Once the useful B cells are chosen according to the pattern of the antigen, they are reproduced. Reproduction multiplies the number of clones. Therefore, the number of B cells that produce the specific antibodies for a particular antigen type is also multiplied. This process is defined as clonal expansion. During reproduction, the clones suffer a process called hyper mutation. Hyper mutation is the process that alters the shape of the clones with relation to their parent cells. This process increases the affinity between the antigen and the newly created clone. Higher the affinity, higher is the ability to recognize the antigen. Therefore, the highest affinity clone or the fittest clone becomes the best that recognizes the antigen. Also, for best recognition, the receptor population should be enough to recognize the foreign shape. Therefore, by increasing the number of clones, a better receptor population is generated [11, 12, 16]. The main features of the clonal selection theory as given by Burnet are given below [19].

- The new cells are copies of their parents (clones) subjected to a mutation mechanism with high rates, which is called hyper mutation.
- Elimination of newly differentiated lymphocytes carrying self-reactive receptors leading to autonomous diseases.

- Proliferation and differentiation on contact of mature cells with antigens.

Clonal selection and expansion operate on both the T and B cells. Clonal expansion and hyper mutation allow generation of higher affinity immune cells. Some of the clones on completing the job of binding the antigen differentiate into memory cells and the rest of the clones become plasma cells that produces cells with higher affinities [20, 21]. The memory and plasma cells are explained in detail in the following paragraphs. Figure 2-6 shows the antigen, B cells with antibody receptors, selected cells for cloning, cloned B cells, memory cells and plasma cells.

![Figure 2-6 Clonal selection and expansion](image_url)

Figure 2-6 Clonal selection and expansion
It can be seen that B3 cell is selected and cloned due to clonal expansion. Finally, the cloned cells transform into memory cells and plasma cells.

2.8.3 Immune Memory

The ability of the immune system to remember the already entered or attacked antigens is called immune memory [3]. There are two cases of immune memory.

2.8.3.1 Primary and Secondary Responses

When the immune system encounters an antigen for the first time, it invokes the antibodies to kill the infection. This is called primary response, where a large number of antibodies are produced. Some of them remain in the system even after the infection is cleared. These remaining antibody cells effectively remember the killed antigen. That is, the immune system is prepared for that antigen for the next time. This is called the secondary response. The secondary response makes the system more rapid and accurate towards the incoming antigens [3, 11]. In comparison to the primary response, secondary response is characterized by a shorter lag \( H \), and a longer persistence of antibody synthesis [11]. Figure 2-7 represents the primary and secondary responses with respect to time. The lag time (or time for recognition) for antigen \( Ag_1 \) is longer during the primary response whereas the lag is shorter during the secondary response. At the same time, recognition of antigen \( Ag_2 \) takes longer time, since it is the primary response for \( Ag_2 \). This lag is represented by \( H \) in Figure 2-7.
2.8.3.2 Associative Memory

Memory in immune system is also associative. In the previous case, the secondary response towards the same antigen was discussed. Suppose, there are two antigens A and C, that attacks the system in a time interval. The B cells adapt to the antigen A that is the primary response and offers a faster secondary response to C. This holds good when the antigens A and C are similar even though they are not identical to each other. Here the antigen A originally establishes the memory and the immune system reacts to C by the property of associative memory. In immunology, this is also called immunological cross-reaction or cross-reactive response as shown in Figure 2-8.

Figure 2-7 Primary and Secondary responses [11]
Associative memory is found very useful in artificial intelligence and neural networks [22] because of the following reasons.

- The data stored is recovered through the reading of the same or related data.
- They are usually robust not only to noise in the data but also to failures in the components of the memory between different antibodies [14].

It can be seen that the antigen A and C are different. But since they are similar, the memory cells can associate antigen C with A and still provide a response that can attack antigen C because they are similar.

2.8.4 Reinforcement Learning

In the normal course of the immune system, every B cell would be expected to encounter a given antigen repeatedly. In the initial exposure to an antigenic stimulus, immune response is adaptive. A small set of clones, each producing antibodies of different affinity due to affinity maturation, handles the immune response. The effectiveness of the response to secondary encounters would be enhanced, by storing some of the high affinity antibody producing cells from the first infection, so as to form a large initial clone for the subsequent encounters [5, 6]. This strategy helps to improve the immune system.
response in terms of speed and accuracy, rather than starting the detection every time [3,11]. This procedure is called reinforcement learning where the system is continuously improving its capability to perform the task [15]. This process of learning in immune system involves memory of the pattern of the antigen and the corresponding antibody generation. Researchers in immunology still debate the cause of immune memory. However there are three reasons quoted in the literature for the existence of immune memory [23].

- The clones that differentiate into memory cells have longer life
- Memory cells are restimulated for the immune response
- Due to Jerne’s idiotypic hypothesis, there is a co-stimulation between the B cells even in the absence of antigen that mimics the process of antigenic stimulus.

The following section explains immune network in detail.

2.9 Jerne’s Idiotypic Network Theory

Jerne’s idiotypic network hypothesis [24], [25], [26], [27] as proposed by the immunologist deals with the interaction of the antibodies. Jerne’s immune network is a network of B cells that communicate the shape of the antigenic epitope through idiotopes and paratopes. This communication alters the structure of the receptors according to the antigenic pattern. This shape transformation is an important role of information transfer and communication between the B cells. Figure 2-9 shows the immune network representation. In Figure 2-9, the respective idiotopes and paratopes of the B cells B1, B2 and B3 are labeled as \( Id_1, Id_2, Id_3, P_1, P_2 \) and \( P_3 \). On an antigenic stimulus \( Ag \), idiotope, \( Id_1 \) of antibody \( (Ab_1) \) stimulates the B cell \( B_2 \), with the antibody \( (Ab_2) \) through the
paratope $P_2$. Viewed from the point of $Ab_2$, idiotope $Id_1$ of the $Ab_1$ works simultaneously as antigen. As a result, $Ab_2$ suppresses $B_1$. On the other hand, $Ab_1$ stimulates $Ab_3$ since the $Id_1$ of the $Ab_1$ works as an antigen for the $Ab_3$.

Figure 2-9 Jerne's immune network

These mutual interaction chains between different species of antibodies form the large-scale closed chain loop, which works as a self and non-self recognizer [28]. Besides the information transfer between the $B$ cells, the immune network is also necessary for the collaborative activity of killing the antigen. The reason is that the antigens have multiple epitopes and the $B$ cells are mono-specific, with a single type of receptor [14]. Therefore in order to kill the antigen the $B$ cells with different types of receptors group through the immune network to bind the antigenic epitopes and eventually kill them. The immune network is the major reason for self-organization ability of the $B$ cells.
2.9.1 Need for the Immune Network

As stated in the earlier sections, each B cell has antibodies that are Y shaped receptor molecules on its surface whose role is to recognize and bind to the antigen. Contrary to the antigen that has different types of epitopes, each B cell has single type of antibody (mono specific). Since antigens have different types of epitopes, for the antigen to be killed, a complementary pattern of all the epitopes is necessary to bind the antigen. This is possible only when many B cells of different types work together [29]. When B cells communicate with each other, they collectively provide different antibodies of different shapes corresponding to the epitopes of the antigen. An antigenic stimulus initiates the process of network stimulation. Each B cell understands the other cell by means of the shapes of the paratopes and the idiotopes. That is, the V region of the paratope changes according to the shape of the antigen and results in idiotopes of different shapes. This shape is communicated to the next B cell whose V region is changed accordingly. This process works in a chain and creates antibodies of different shapes that eventually bind and kill the antigens.

2.10 Self Organization

There is no central organ that controls the functions of the immune system since the mechanisms of immune response are self-regulatory in nature. The regulation of immune responses can be either local or global depending on the route and property of the antigenic pattern [1]. This means that, the B cells organize themselves to work independently or in groups.
2.11 Self/non-self discrimination – Positive/Negative selection

The immune system needs to be able to distinguish from the molecules of *internal cells* (self) and foreign or *external* (non-self) molecules, which are a priori indistinguishable [30, 31]. In the immune system, the distinction of the cells becomes important because an immune response may be triggered to the self-cells causing *autoimmune diseases*. The ability of the immune system by which it differentiates and understands between its own and the foreign cells is called *self/non-self discrimination* [32].

The shape transformation plays an important role for such discrimination. Additionally, an encounter between a lymphocyte receptor and an antigenic epitope does not necessarily result in activation of the lymphocyte. In that respect, the response to the stimulus may take two turns called the *positive selection and negative selection* as explained in the following sections.

2.11.1 Positive Selection

Positive selection is the process whereby a lymphocyte antigen interaction results in the growth of that lymphocyte when a smaller percentage of cells mature into *immuno-competent* cells. This process is responsible for the controlling the survival of the immune cells.

2.11.2 Negative Selection

Negative selection of a lymphocyte describes the process whereby self-cell’s interaction results in the death of that lymphocyte [30]. This permits the control of those lymphocytes bearing non-self receptors that would fight against self-cells. In this process, the *B* and *T* cells are purged out. This prevents self-specific lymphocytes from becoming auto aggressive.
Forest et al developed a negative selection algorithm [31] on the basis of self/non self-discrimination of the immune system [32]. Negative selection uses the property of self/non-self distinction to detect the foreign antigens. In the biological system, this is achieved in part by $T$ and $B$ cells that have receptors on their surface that can detect foreign antigens. The process of receptor generation is random [1]. Then they undergo a process of filtering in the thymus where the cells that react against self-cells are destroyed and only those do not bind to self-cells are allowed to leave the thymus. These matured $T$ and $B$ cells circulate through out the body to protect against foreign antigens.

Figure 2-10 shows the self / non-self discrimination of the immune system. It can be seen that self-cells that recognize the antigen are selected and undergo clonal expansion.
In this process, the self-cell that cannot recognize the antigen is ignored and the self-cell that recognizes the self-cells are deleted in the process.

2.12 Types of Immune Responses

The immune responses described in the above sections can be summarized into two categories. They are

- **Humoral immunity** mediated by the B cells and
- **Cellular immunity** mediated by the T cells

Both follow a similar kind of defense except for a difference as explained in the following discussion.

2.12.1 Humoral Immune Response

In humoral response, a specific type of B cell proliferates and a part of these B cells produce plasma cells. These plasma cells produce antibodies that react with the antigen and eventually kill it. The remaining fraction of the B cells become dormant and circulates in the bloodstream and remembers the pattern of the antigen. These cells later become memory cells whose lifetime is longer than the other B cells.

2.12.2 Cell Mediated Response

Cell mediated response is associated with T cells. A specific type of T cell becomes cytotoxic, and kills the antigen directly. The corresponding long-lived T cells carry the memory of the specific pattern of the antigen. The helper T cells play an important role in both kinds of immune responses. By this process, the human system is supposed to have gained or acquired immunity to a specific antigen. Because of the presence of memory cells the secondary response is quicker and stronger. This is the primary principle of vaccination [21]. Figure 2-11 shows the humoral and cellular immunity.
2.13 Overall Functioning of the Immune System

The overall functioning of the immune system is illustrated in Figure 2-12. The immune system recognizes the antigens and the antigenic patterns are identified. On identification of an antigenic pattern, the $B$ cells communicate the information to each other simultaneously by means of paratopes and idiotopes in the network. Paratopes match with the epitopes of the antigen to recognize the antigen. Paratopes also change their shape to strengthen the bond between the epitope and the paratope. However, the binding stays only for a short time called the *tolerization period* within which a number of receptors should bind the antigen. When this process of binding within a short period happens, the $B$ cells gets activated and performs a set of actions to kill the antigen. On
activation, every $B$ cell responds by changing the shape of the receptor according to the antigenic epitope. $B$ cells that have higher affinity towards the antigen are the ones that recognize the antigen.

![Diagram of the human immune system](image)

**Figure 2-12 Representation of the human immune system**

The useful cells undergo multiplication by clonal expansion and produce high affinity cells or clones. Since the antigen has multiple epitopes and the $B$ cells are mono-specific, the $B$ cells work together to kill the antigen through immune network. Part of the clones differentiate into plasma cells that create higher affinity cells and the rest turn out to be memory cells that remember the antigen that was destroyed. Thus the human system attains immunity against the antigens.
3. Artificial Immune Systems and their Applications

3.1 Introduction to Artificial Immune Systems
The natural immune system is a subject of great interest because it possesses powerful information processing capabilities. The immune system performs complex computational tasks in an efficient manner. Hofmeyer gives a good reason for why the immune system interests engineers and computer scientists [17]. He states that computer systems, in particular networks, have in complexity. Since the complexity of such systems approaches the complexity of biological systems, it is natural to turn to biology-based solutions.

The natural immune system is a sophisticated system and engineers hope that it can provide knowledge for experimentation and find solutions to difficult problems. The computational aspects of the immune system are pattern recognition, distributed detection, sophisticated information processing, autonomous decentralized control, adaptive learning, in built memory and self organization [1].

This chapter briefly discusses AIS definitions, applications of AIS, some of the AIS based models derived out of the properties and functioning of the immune system.

3.2 Definitions of Artificial Immune Systems
An artificial immune system (AIS) can be defined as follows.

An artificial immune system (AIS) is defined as an evolvable computational system obtained by imitating the biology of the human immune system in terms of its features and properties that helps the human body to attain immunity.

There are several other definitions of artificial immune systems in the literature. Timmis states that "artificial immune systems are data manipulation, classification,
reasoning and representation methodologies that follow a plausible biological paradigm: the human immune system” [34]. On the other hand, Hajela and Yoo state that an artificial immune system is a computational system based upon metaphors of the natural immune system [35]. Dasgupta defines “artificial immune systems are composed of intelligent methodologies, inspired by the natural immune system, for the solution of real-world problems” [1].

Artificial immune system provides a lot of scope in algorithm development that explores various mechanisms of the immune system and their relation to information processing and solving complex engineering problems. This is achieved by modeling the parameters and concepts of the immune system.

3.3 Artificial Immune Systems and Their Applications

Artificial immune systems have been employed to solve several problems. Since the number of applications solved by artificial immune system is quite high, only some of the application areas are listed. Applications of AIS are not limited to the following engineering fields such as optimization [35], computer security [12, 15, 36, 37], neural network approaches [38], data mining [39], robotics [13, 40], image segmentation and inspection [41], image classification [42], autonomous agents [14, 43, 44] and evolvable hardware [45]. A detailed collection of the AIS applications can be found at [46]. Table 3-1 lists the applications and their relation to the properties the immune system.
a lattice at \( r \) in a \( d \) dimensional space, then the anti-idiotype or the epitope of the antigen is located at \(-r\) on the same \( d \) dimensional space. Therefore, the strength of the bond at the lattice \( r, -r \) is maximized. This model hence provides a measure for the affinity between the clones [47].

![Figure 3-1 Shape space model](image)

The cellular automata model uses the property of clonal selection to describe the population dynamics of the \( B \) cells. The populations of the \( B \) cells are formulated in two ways namely the discrete approach and the continuum approach.

In the discrete approach, the population is represented by a discrete variable that takes only binary values corresponding to low and high populations. This discrete variable is called the automaton and a system consisting of such mutually interacting automata are called cellular automata. In this case, the dynamic equations are represented as maps in discrete time. Also, the interactions are allowed to have only discrete values. The discrete model can be formulated either by threshold automata or Boolean automata [48, 49].

The discrete immune network model (aiNET) is an artificial immune network model demonstrating the immune concepts useful for data analysis. The model assumes
a set of unlabeled patterns, where every pattern is defined by a variable that characterizes the molecular configuration as a point in space. This space is called the shape space. The shape space sets the features needed to determine the interactions between two antibodies or the interaction between an antibody and an antigen. These interactions are defined by a connectivity graph. The aiNet model as described by De Castro [15] is as follows.

The evolutionary artificial immune network, named aiNet, is an edge-weighted graph, not necessarily fully connected, composed of a set of nodes, called cells, and sets of node pairs called edges with a number assigned called weight, or connection strength, specified to each connected edge [15].

There are other models such as the differential equation models [22] and stochastic equation models [50] that use the properties of the immune system such as learning, diversity, pattern recognition and immune network. Some of the models incorporate other biological paradigms such as the genetic algorithm for hyper mutation of antibodies [42, 51]. There are comparisons of the nervous system and the immune system and development of neural network model [38] based on the comparison. There are also similarities derived for the immune system based on learning classifier systems [37, 52, 53, 54].

The following section describes the negative selection algorithm, mathematical representation of affinity maturation and immune memory. These derivations are used in the proposed AISIMAM explained in Chapter 5.

3.5 Negative Selection Algorithm

Figure 3-2 shows the negative selection algorithm with the censoring and monitoring phase. As shown by the flowchart, in the censoring phase, the randomly generated strings are matched with the self-strings, and if a match is found they are
rejected. If there is no match, detectors are valid and they are used to match with the non-self cells to detect the non-self cells in the monitoring phase.

After Forrest et al., 1994

Censoring phase

Monitoring phase

Figure 3-2 Negative selection algorithm [15]

Negative selection algorithm defined by Dasgupta works on the principle of generating detectors randomly and eliminating the ones that detect self-cells, so that the remaining T cells can detect any non-self [1]. The algorithm as summarized by Dasgupta [1] is as follows.

- Define self as a collection of strings S of finite length L that needs to be monitored.
- Generate a set of detectors R each of which fails to match any string in S.
- Monitor S for changes by continually matching the detectors in R against S. If any detector matches, then a change is known to have occurred, because the detectors are designed not to match any of the original strings in S.
3.6 Affinity Maturation

When the antigen attacks the body, the receptors on the B cells frequently suffer a process called hyper mutation that edits the receptors of the B cells, and therefore the antibody response improves after repeated immunization [18]. When the binding gets stronger the affinity between the receptor and the epitope also increases. Let \( r \) and \( e \) represent the receptor of the antibody and epitope of the antigen. Let \( B(r,e) \) represent the binding between the receptor and the epitope and \( A_f(r,e) \) represent the affinity between the receptor and the epitope. Here, affinity is the complementary match between the receptor of the antibody and the epitope of the antigen [12]. Affinity is directly proportional to the binding shown in Eq. (3.1)

\[
A_f(r,e) \propto B(r,e) \quad (3.1)
\]

Once the receptor and the epitope match, the B cells are activated and trigger a set of actions that kill the antigen. This activation happens only if the binding with \( N_r \) receptors is higher than a threshold \( T_{hr} \) [17] as shown in Eq. (3.2). If \( A_l \) represents the activation of the lymphocytes,

\[
A_l = 0 \text{ if } B(N_r,e) < T_{hr} \\
A_l = 1 \text{ if } B(N_r,e) > T_{hr} \quad (3.2)
\]

But the bonds between the receptor and the epitope are not long lasting. It exists only for a certain amount of time \( \tau \) called the tolerization period [17]. Thus, for the lymphocytes to get activated they should bind a number of receptors within this short period of time. If the time taken for this binding is given by \( t_b \), then the binding time should be less than the tolerization period given by Eq. (3.3)

\[
A_l = t_b \leq \tau \quad (3.3)
\]
B cell activation is the result of the antigenic recognition.

3.7 Immune Memory

The immune memory is explained in Section 2.8.3. During the primary response, \( (R_p) \) where a large number of antibodies are produced. Some of the remaining antibody cells effectively remember the killed antigen for the next time. This is called the secondary response, \( (R_s) \). This makes the system more rapid and accurate towards the incoming antigens [3, 11]. In other words, the primary response of the immune memory \( R_p \) is slower than the secondary response \( R_s \). If \( T_p \) and \( T_s \) are the times taken for the primary and secondary responses respectively, then \( T_p \gg T_s \).

The efficiency of the primary and secondary responses depends on the response times and the number of lymphocytes that it bonded during the response time [17]. If the numbers of lymphocytes that bind the antigen during primary and secondary responses are \( N_p \) and \( N_s \), then \( N_p \ll N_s \). And if the efficiency of the primary and secondary responses are represented by \( \eta_p \) & \( \eta_s \), then efficiency can be represented as a function of the number of lymphocytes that bind the antigen and the time duration of the primary and secondary responses as given in Eq. (3.4)

\[
\begin{align*}
\eta_p &= f(N_p, T_p) \\
\eta_s &= f(N_s, T_s)
\end{align*}
\] (3.4)

Cloning the B cells increase the efficiency of the memory response [17]. The efficiency of the memory increases exponentially as the number of clones increases. The efficiency of the memory \( M_j \) where \( j = 1,2,..., k \) can be formulated as an exponential function of the clones given by Eq. (3.5).
\[ \eta(M_j) = e^{c_i} \] (3.5)

Figure 3-3 shows the flowchart representation of the immune algorithm.
Initialization of Ag and Ab

Antigens have epitopes of different shapes on its surface

B cells generate antibodies that are receptors on its surface called the paratopes and idiotopes

N

Ag

Y

B cell N/w
Ab–Ab interaction
Leading to suppression

B cell N/w
Ag–Ab interaction – leading to n/w stimulation

Pattern recognition by the receptors on the B cells – Use Self/Non-self discrimination

Antibody generation

Affinity maturation (Af)
Affinity between Ag - Ab

Af > Tr

Y

N

Clone the B cells of higher affinity - Reduces the size of the total population of B cells, but increases the size of the useful population of B cells - During reproduction they undergo hyper mutation which alters the shape of the clones which increases the affinity

Fraction of the clones differentiate into plasma cells which generates antibodies of higher affinity

Fraction of the clones differentiates into memory cells and circulates in the blood carrying a memory encounter of the antigen.

Antibodies interact with the antigen and eventually kill it

Figure 3-3 Flowchart representation of the human immune system
3.8 Need for a Global Model

In the previous section, the shape space model, the cellular automata model and aiNet models were explained. The idea is to show how models are developed using the immunological concepts rather than concentrating on a specific model.

This thesis proposes to develop a novel AIS based model that combines several components and properties of the immune system. The aim of the model is to make global definitions of the immune system concepts in such a way that it can be applied to several applications by defining the parameters of the model.

The proposed model uses AIS principles in multiagent systems due to the strong similarities found between both the systems. The model represents the immune cells and molecules, and formulates the interactions between them. It also follows the intelligent agent definitions and multiagent systems. Immune system concepts such as negative selection algorithm, clonal selection and expansion, immune network, immune memory, affinity maturation are used in the model. The next chapter explores multiagent systems. Then the proposed AIS based multiagent system is explained.
4. Multiagent Systems

4.1 Introduction to Multiagent Theory

For the last several thousand years, philosophers have examined learning, remembering, and reasoning. In 1943, Warren McCulloch and Walter Pitts started the first work on Artificial Intelligence (AI). There are several definitions of AI found in the literature. These definitions vary in two main dimensions. They are grouped into two categories namely thought processes or reasoning and behavioral processes. In each category, “success” is measured either in terms of a human’s performance or in terms of “ideal” intelligence (rationality) [55]. According to Russel and Norvig, AI definitions can be of four types. They are

- Systems that think like humans
- Systems that act like humans
- Systems that think rationally and
- Systems that act rationally [55]

As research improved over time, the sub fields of AI emerged. One such field is Distributed Artificial Intelligence (DAI). Bond and Gasser define distributed artificial intelligence as “the sub field of AI concerned with concurrency in AI computations” [56]. DAI is concerned with systems that consist of multiple independent entities that interact in a domain. DAI is divided into two namely; distributed problem solving and multiagent systems.

- Distributed problem solving (DPS) focuses on the information management aspects of systems with several branches working together towards a common goal.
• Multiagent Systems (MAS) deals with the behavior management in collections of several independent entities, or agents.

DPS deals with how to solve a problem using a number of modules that divide labor and share knowledge about the problem. In MAS, autonomous agents coordinate their skills, knowledge, goals, and plans to accomplish a global problem. In addition to accomplishing the global goal, the agents in MAS may work toward achieving their own individual goals. The main difference between DPS and MAS is that MAS can be an open system where there is no global control. In other words, MAS can be either a centralized or decentralized system, where DPS can only be a centralized system [55].

4.2 Agent Definitions

There is no universal definition for an agent found in the literature. In fact, there is a good deal of controversy and debate on agent definitions. A few definitions of agents are stated here [56].

• Wooldridge defines an agent as a computer system, that is situated in some environment, and that is capable of taking actions autonomously in this environment in order to meet its design objectives [57].

• Distributed Artificial Intelligence researchers focused on agents as computational entities that interact with each other to solve distributed problems [58].

• Nwana and Ndumu defines an agent as a component of software and/or hardware which is capable of acting in order to accomplish tasks on behalf of its user [59].
Bond and Gasser define agents that operate robustly in rapidly changing, unpredictable or open environments, and where there is a significant possibility that actions can fail are known as *intelligent agents* or sometimes called *autonomous agents* [56].

The agent takes sensory input from the environment and produces actions that affect the environment. Figure 4-1 shows an agent in its environment taking an input from the sensor and produces an action to the environment. The interaction is usually an on-going one.

![Figure 4-1 An Agent in its environment [60]](image)

The *Effector/Medium/Sensor* paradigm (EMS) explains this. This paradigm provides an appropriate abstraction of a human agent acting and interacting with its environment and other agents [60].

For an agent to be intelligent and to perform a task in the way the humans do, it needs to have some properties possessed by human beings. Some of the features of the agents that make them intelligent as stated by Wooldridge [58] are

- **Reactivity:** Intelligent agents are able to perceive their environment, and respond to changes that occur in a timely fashion in order to satisfy their design objectives.
• **Pro-activeness**: Intelligent agents are able to exhibit goal directed behaviors by taking the initiative in order to satisfy their design objectives.

• **Social ability**: Intelligent agents are capable of interacting with other agents in order to satisfy their design objectives.

### 4.3 Structure of Intelligent Agents

The goal of multiagent systems is to develop the agent methodologies [55]. An intelligent agent methodology includes agent architecture to perform the defined task and an agent program to process and execute the task.

The agent program can be a mapping function that implements the perceived concepts to actions. It can be an algorithm or program to perform a particular task. The agent program runs on a computing device. This computing device is called the architecture in agent terminology. According to the architecture, the agents can either be *hardware* agents or *software* agents. The hardware agents could be plain computers or specific hardware built to perform a task. For instance, they could be customized robots, devices for processing camera images, or specially designed filters to filter an audio or video input. On the other hand, software agents are software programs that can exist in local or global domains to perform a task [55].

The constituents of an agent in simple terms can be a program and architecture. The agent program can also be classified into three different categories according to how the program is implemented. They are **simple reflex agents**, **goal-based agents** and **utility-based agents** [55]. Simple reflex agents are limited in their actions because their actions are determined only by the current state of perception. Goal-based agent programs have the information not only about the current state, but also about the
information about results of possible actions in order to choose actions that achieve the goal. Problem solving agent is an example of goal-based agent. Utility-based agent programs operate according to the concept of utility value associated with the goal. In other words, the agent can have several goals that it should aim for without certainty to reach it. The utility provides a measure of likelihood of success and weight for the importance of goals. Utility-agents thus provide a measure to decide in the event of conflict or to prioritize the goals.

Having defined agents and some features of intelligent agents, concept of multiagents are defined in the following section.

4.4 Multiagent Systems

Agents can exist alone or in a society of agents called multiagents (MAS). Multiagents are population of agents, that is, more than one agent can change the environment to accomplish the task. Each agent in a MAS has a list of goals or tasks. Similarly, a MAS has a list of global goals that it will strive to achieve where each agent contributes some effort toward reaching these global goals [61]. The multiagent environment is usually open, decentralized, and consists of autonomous agents [62]. In addition, Huhns and Singh believe that agents are better “developed” in a multiagent system than in isolation [61]. Moreover, Huhns and Stephens believe that multiagent systems are the best way to build distributed computational systems [62].

In multiagent systems, agents have independent access to the environment [60]. At the same time, they have to organize themselves and interact with each other to perform a common task. In the literature, these types of problems are called as self-
organizing problems. The following section describes some characteristics of agents and the environments.

4.5 Characteristics of Agents

The agent’s contribution to their MAS is controlled by their behavior such as cooperation, altruism, friendliness, and benevolence. In addition, agent’s interactions with each other are determined by their characteristics such as the autonomy, sociability, friendliness, level of cognition and mobility [63].

Autonomy in agents is a measure of self-sufficiency. The agents that operate on their own are independent agents, and if they are restricted by external influences then they are called controlled agents. Sociability is a behavioral measure of an agent to think about itself or about others. An altruistic agent acts regardful of other’s benefits, and is unselfish. In contrast, an egoistic agent acts with excessive thoughts of self and is self-loving.

Agents could be friendly and be cooperative or compete with each other. Agents are classified into reactive and deliberative according to their level of cognition. The first ones sense and react in a timely manner for an environmental change and the latter ones reasons out before making actions. Mobility determines if the agents are stationary or itinerant. Stationary agents do not move and itinerant agents are mobile.

Other characteristics of the agents that deal with the agent’s adaptability, rationality and locality can be referred in the literature [63, 64]. The following section discusses different kinds of environments in which agents operate.
4.6 Types of Environments

The key problem facing an agent is that of deciding which of its actions it should perform in order to best satisfy its design objectives. Agent architectures are really software architectures for decision-making systems that are embedded in an environment. The complexity of the decision making process can be affected by a number of different environmental properties [63, 55]. Russell and Norvig suggest the following environmental properties [55].

An *accessible* environment is one in which the agent can obtain complete, accurate, up to date information about the state of the environment. The more accessible an environment is the simpler it is to build agents to operate on it. Complex environments like the physical world are defined as *inaccessible* environments.

*Deterministic* environment and *non-deterministic* environment deal with the certainty of the any action and responds to such an action. The physical world can be regarded as non-deterministic. This creates greater problems to the designer.

An *episodic* environment deals with the performance of agents in discrete episodes without any links and a *non-episodic* environment deals with linked actions between the past and current data respectively. An example of the episodic environment is a mail sorting system where the agent decides its action according to the current episode [55].

A *static* environment remains unchanged unless there is an action performed by the agent. A *dynamic* environment has other processes operating on it and changes beyond the agent’s control. The physical world is a highly dynamic environment.
4.7 Central Issues in MAS

There are several issues involved in a multiagent system. They are interaction, cooperation, organization, action and behavior [64]. Every issue has its own components. The components for interaction are the objectives or intentions of agents, the relationships of these agents to their resources and the skills to achieve their goals. The cases that arise from the three components of interaction are independence, collaboration, obstruction, coordinated collaboration, individual competition, collective competition, individual and collective conflicts over resources. Similar issues related to cooperation are grouping and multiplication, communication, specialization, collaboration by sharing tasks and resources, coordination of actions, conflict resolution by arbitration and negotiation.

The internal organization of an agent is defined as the *architecture*. Among several kinds of architectures, some are found to be possible for implementation. They are modular architectures, blackboard based architectures, subsumption based architectures, production rules, classifiers and dynamic system architectures.

There are several ways in which the actions are modeled in MAS. Action can be modeled as a transformation of a global state, response to influences, computing processes that use tools such as petri nets and finite state automata. Actions can also be modeled as a physical displacement in physical and geometrical models, and as a command in cybernetic model [64]. Other techniques and specific models can be found in the literature [55 - 64].
4.8 Performance Measures for Multiagent Systems

An agent in a MAS determines solutions using the central issues stated in Section 4.7. Some of the parameters to be determined are [55] given as follows.

Completeness determines if the chosen method can find the solution to the given problem. Time complexity is used to measure the required duration to find the solution. Space complexity is a measure of the space requirement of space to reach the solution. For example, space could be memory required in a computer system, or area in a real time problem. Optimality provides a measure of finding out if the arrived solution is the best or optimal solution for the given task. Computational complexity determines the time for the convergence of the solution.

4.9 Examples of Agents

At this point it is worth noting some of the examples of agents. Any control system can be viewed as an agent. A simple example of such a system is a simple thermostat. Thermostats have a sensor for detecting room temperature. This sensor is directly embedded into the environment and it senses the temperature to give two output signals. If the temperature is too low then the heating in ON and if the temperature is high then the heating is OFF. According to the two states of the signals given out by the sensor the agents take either of the decision. More complex environment structures however have got more complex decision-making structures [63].

Most software daemons, which monitor a software environment and perform actions to modify it, can be viewed as agents. An example is the X-windows program xbiff. This utility continually monitors a user’s incoming email, and indicates via a GUI icon whether or not there are unread messages. Unlike the previous example, in the
physical world xbiff has a software environment [63]. Agents can be cooperative on the web and help each other search for information that will greatly reduce the traffic on the internet thereby benefiting everyone. The task for the agent here is to pickup information from different information resources for a given query. The following situation might also occur on the web. Agent x asks agent z about a piece of information, but agent z does not know the answer. Later, agent y asks agent z the same question, but agent z still does not know the answer. If agent z is a cooperative agent then it will inform agent y that agent x was asking the same exact question and agent y should go and ask agent x because agent x most likely found the answer by now. This will save agent y a lot of time and reduce the traffic on the net. This example demonstrates the power of cooperation between agents [63, 55].

When a space probe makes its flight from earth to outer space, a crew is usually required to make decisions on continuously tracking the progress. Sometimes it becomes practically impossible and expensive when decisions are to be made quickly. For these reasons, organization like NASA is adding autonomous agents aboard their space probes thus solving the cost and speed problem. Deep Space1 (DS1) is carrying autonomous agents on board. [55, 63].

4.10 Multiagent Systems Applications

Multiagent systems have been found useful in several applications. Some of the application areas of MAS are controls [65, 66], communication [67], robotics [68, 69], e-commerce [70] and computers networks and security [71].

This chapter has provided an introduction to multiagent system with agent definitions, agent characteristics, agent types, types of environments, examples of agents
and applications of agents. The following chapter discusses artificial immune system architecture for multiagent systems.
5. AISIMAM – An Artificial Immune System based Intelligent Multi Agent Model

5.1 Introduction

This chapter deals with the mathematical derivations of an Artificial Immune System based Intelligent Multiagent Model named AISIMAM. Initial sections of the chapter deal with the how AISIMAM is derived using MAS and AIS concepts. Later section covers the mathematical derivations followed by the need for such a mathematical representation. To verify the model, an application is simulated. The problem experimented is mine detection and defusion. The results show that AISIMAM is successful in solving a multiagent problem. Chapter 6 covers the mine detection application and the results.

5.2 AISIMAM - Artificial Immune System Based Intelligent Multiagent Model

The backbone of AISIMAM involves imitating the artificial immune system in terms of features and functions of multiagent systems.

The motivation for this research comes from the fact that artificial immune systems have found solutions for several applications [35 - 46]. In the same context, agent based solutions have also been developed in different application domains [65 -71]. The reason for developing the AISIMAM is due to the similarities observed between the architecture of immune system and multiagent systems. The distinct similarities between the multiagent systems and the immune systems are

- Both are distributed or decentralized systems
- Both have multiple autonomous entities
- Both have individual and global goals
- Both systems learn from their experience
5. AISIMAM – An Artificial Immune System based Intelligent Multi Agent Model

5.1 Introduction

This chapter deals with the mathematical derivations of an Artificial Immune System based Intelligent Multiagent Model named AISIMAM. Initial sections of the chapter deal with the how AISIMAM is derived using MAS and AIS concepts. Later section covers the mathematical derivations followed by the need for such a mathematical representation. To verify the model, an application is simulated. The problem experimented is mine detection and defusion. The results show that AISIMAM is successful in solving a multiagent problem. Chapter 6 covers the mine detection application and the results.

5.2 AISIMAM - Artificial Immune System Based Intelligent Multiagent Model

The backbone of AISIMAM involves imitating the artificial immune system in terms of features and functions of multiagent systems.

The motivation for this research comes from the fact that artificial immune systems have found solutions for several applications [35 - 46]. In the same context, agent based solutions have also been developed in different application domains [65 -71]. The reason for developing the AISIMAM is due to the similarities observed between the architecture of immune system and multiagent systems. The distinct similarities between the multiagent systems and the immune systems are

- Both are distributed or decentralized systems
- Both have multiple autonomous entities
- Both have individual and global goals
- Both systems learn from their experience
- Both are adaptable
- Both sense the changes in the environment and act accordingly
- Both systems communicate and coordinate
- Both possess knowledge with which they make intelligent decisions.

Therefore, immune system based multiagent architecture is derivable. The following section describes architecture of AIS based intelligent multiagent system with necessary comparisons and explanations.

5.3 Comparison of AIS and Multiagent Systems

AISIMAM imitates the humoral response in a vertebrate immune system. Since the model assumes vertebrate immune system, AISIMAM acquires immunity instead of innate immunity.

The model defines the non-self cells (antigens) and self-cells (B & T cells) as two agents with different characteristics and goals. Therefore, the two types of agents in AISIMAM are antigens modeled as non-self agents (NAGs) and lymphocytes or self-cells modeled as self-agents (SAGs).

The environment is defined as a matrix in which both the NAGs and the SAGs operate. The environment can be any one of the types of environment explained in Chapter 4, depending on the application. There is an information vector for each non-self agent. This could represent a disturbance in a process, malfunction or a virus in a computer network depending on the application. The information vectors correspond to the epitopes of the antigen. Similarly, each self-agent has an information vector that defines the self-goals. The information vectors correspond to the receptors of the lymphocytes. The information vector can contain a single datum or multiple data. For
example, the information could be a location information, identification number, text information, or all of them depending upon the application. This information is considered as idiotopes and the paratopes. However, the model does not distinguish between the paratopes and idiotopes. Instead, the target will be to perform the end goal with the available information by each self-agent. The end goal could be destroying the non-self agent as the antigen is killed in the immune system, or it can be to identify the best action sets of each self-agent to react to the non-self agent’s action vector. This is however problem dependent.

The information vectors and the characteristics of the self and the non-self agents differ from each other. This is similar to the structures of the epitopes of the antigen and the paratopes of the lymphocytes. In other words, the agents perform individual actions or goals determined by the action generator function and the global goal is the coordinated actions of the individual SAGs. The individual action of the agent corresponds to the receptor shape change in a B cell and the coordinated actions correspond to a group of B cells killing the antigen.

The SAGs are assumed to have sensory capability to identify the NAG within a region called sensory neighborhood. They also possess the capability to communicate the NAG information to the other SAGs within a region called communication neighborhood. The model assumes that the communication neighborhood is greater than the sensory neighborhood. This is in comparison with the capability of the B cells to recognize the antigenic pattern within a particular neighborhood. In immune system, the communication circle is analogous to the communication between B cells connected in the immune network (Jerne’s Network). In other words, every B cell communicates the
information to B cells that are within the communication neighborhood in the immune network. Similarly, the communication is achieved through the agents network.

The overall representation of the proposed model is shown in Figure 5-1. In Figure 5-1, the SAGs sense the NAG within the sensory circle and communicate to the other SAGS that are within the communication neighborhood. The agent network achieves the communication between SAGS.

![Figure 5-1 Representation of AISIMAM](image)

The following section describes the different stages of processing involved in the agent model. The agent model describes five stages of processing namely pattern recognition, binding process, activation process, post activation process and post processing. Figure 5-2 shows the stages along with the operation performed in each stage of processing.
5.3.1 Pattern Recognition

In pattern recognition, SAGs recognize the presence of the antigen by the *stimulation function* and identify the NAGs by an *identifier function*. This is analogous to the B cells recognizing the antigenic presence and then identifying the antigenic pattern by the shape of the epitopes.

![Diagram of pattern recognition processes in AISIMAM](image)

**Figure 5-2 Processes in AISIMAM**

5.3.2 Binding Process

The model defines an *affinity function* that calculates the affinity value between the actions of the self and the non-self agents. This process is defined as the *binding process*. In the immune system, the affinity is proportional to the binding between the B cell receptors and the epitopes. The affinity calculation in the agent model is the affinity between the epitope of the antigen and the receptor of the antibody. Binding and affinity
are modeled together as binding process in AISIMAM. For instance, the affinity function could be a distance metric such as the Euclidean distance.

5.3.3 Activation Process
The immune system selects antibodies whose affinities are greater than the affinity threshold. Similarly, in the activation process of the agent model, the affinity values that are greater then a set activation threshold are chosen. The actions whose affinity values that are greater than the activation threshold are called mature actions. Mature actions are the actions that are closer to the desired goal. Additionally, in the immune system, a B cell gets activated only if the binding between the epitope and paratope occurs within the tolerization period. The agent model defines the binding period as the time taken by the number of agents to bind the NAG. The model defines this time as a sum of recognition time and grouping time. Recognition time is the time taken by every agent to recognize the NAG and is the same for every agent. The grouping time is the time taken by the other agents to react to the identified NAG and this time differs from agent to agent.

5.3.4 Post Activation Process
The post activation process involves cloning. Here, the agents are reproduced with the mature action. In the immune system, during cloning B cells undergo receptor editing or hyper mutation. Hyper mutation in AISIMAM is the process of generating new actions. Hyper mutation exists as the action generator function.

5.3.5 Post Processing
In post processing, a part of these cloned agents differentiate into memory agents containing the matured action obtained as a result of a particular NAG. The rest of the
clones become plasma agents that create higher affinity actions. Plasma agents could possess the action generator function to generate actions. Post processing involves the primary and secondary response of immune memory, which is also included in the model. Once the end goal is reached, memory agents remember the actions performed to reach the goal.

In the immune system, the efficiency of the primary and secondary responses depends on the response time and the number of lymphocytes that it bonded within the response time [17]. The agent model defines the efficiency of the primary and secondary responses as a function of the actions required to reach the end goal within the binding period.

5.3.6 Agent Network

AISIMAM assumes that all the self-agents work in an agent network similar to Jerne's network. The process of information transfer and communication between the agents is an analogy of the agent network to the immune network. The nature of the agent network is application dependent. Suppression in the agent network is determined by the suppression function. In the immune system, in the absence of the antigenic stimulus, the B cells perform suppression. In AISIMAM, in the absence of a NAG stimulus, suppression is performed. Suppression function is specified by the application. The model assumes the existence of agents network conceptually. The mathematical representation of the agent network will be included as future work.

Table 5-1 gives the analogy of immune system and AISIMAM parameters.
Table 5-1 Analogy of immune system features and AISIMAM

<table>
<thead>
<tr>
<th>Immune System</th>
<th>AISIMAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antigen</td>
<td>NAGs</td>
</tr>
<tr>
<td>B cells</td>
<td>SAGs</td>
</tr>
<tr>
<td>Antigen epitopes</td>
<td>NAGs information vector</td>
</tr>
<tr>
<td>B cells receptors</td>
<td>SAGs information vector</td>
</tr>
<tr>
<td>Pattern recognition</td>
<td>Identifier function</td>
</tr>
<tr>
<td>Immune Network</td>
<td>Agents communication network</td>
</tr>
<tr>
<td>Affinity Maturation</td>
<td>Affinity function</td>
</tr>
<tr>
<td>Affinity threshold</td>
<td>Activation threshold</td>
</tr>
<tr>
<td>Antigenic Stimulus</td>
<td>Stimulation function</td>
</tr>
<tr>
<td>Clonal selection and expansion</td>
<td>Cloned agents</td>
</tr>
<tr>
<td>Immune memory</td>
<td>Memory agents</td>
</tr>
<tr>
<td>Plasma operation</td>
<td>Plasma agents (Action generator function)</td>
</tr>
<tr>
<td>Tolerization period</td>
<td>Binding period</td>
</tr>
<tr>
<td>Suppression</td>
<td>Suppression function</td>
</tr>
</tbody>
</table>

Figure 5-3 shows the flowchart representation of AISIMAM. The flowchart explains the process of every SAG on identification of a NAG stimulus. Depending on the application, NAGs and SAGs can be defined and the processes can be applied.

The following section describes AISIMAM by providing mathematical and algorithmic derivations.
Non-self agent (NAG); Non-self agent has information vector (Antigens have epitopes of different shapes on its surface)

Every SAG in the network does the following

Is there a NAG? 

Identify NAG by SAG
SAG identifies the NAG by the identifier function
B cell N/w - Ag - Ab interaction leading to n/w

Action generator function generates new action vectors according to NAG
Shape transformation causes higher affinity cells

Calculate affinities between the NAG action and each SAG action vector by the affinity function - Affinity maturation (Affinity between Ag - Ab)

Is Affinity > Activation Threshold?

Yes 

Clone the agents with the mature action
(Clone the B cells of higher affinity - During reproduction they undergo hyper mutation that alters the shape of the clones that increases the affinity)

Part of the clones generate actions that have higher affinity values - action generator
(Fraction of the clones differentiate into plasma cells which generates antibodies of higher affinity)

Remaining fraction of the clones turn out to be memory agents with the best action with a memory of the identified NAG.
(Fraction of the clones differentiates into memory cells)

Reach the end goal defined by the application (Antibodies interact with the antigen and eventually kill it)

Self-agents (SAGs) have information vector representing a single type of action (SAG)
(B cells with paratopes and idiotopes on its surface)

Perform suppression
B cell N/w - Ab - Ab interaction leading to suppression

Figure 5-3 Flowchart representation of AISIMAM
5.4 AISIMAM – Operational Scheme and Mathematical Representation

This section starts with the definitions of parameters. Then, the mathematical derivations for the proposed model are presented. Finally, the necessity of such a mathematical model is explored.

5.4.1 AISIMAM - Parameter Definitions

This section defines all the parameters used in the model.

- Define the agents namely the self (SAG) and represent them by $S_i$, where $i = 1, \ldots N$.

- Define the non-self agents (NAG) $N_j$ where $j = 1 \ldots M$.

- Define the problem domain or the environment by $E$, where

$$E = S_i \cup N_j \quad \forall \ i \text{ and } j$$

(5.1)

- Define $\forall \ S_i \in E, \exists$ an information vector of $q$ elements given by $B^i$

$$B^i = [b_1, b_2 \ldots b_q]$$

(5.2)

- Define $\forall \ N_j \in E, \exists$ an information vector of $r$ elements given by $A^j$

$$A^j = [a_1, a_2 \ldots a_r]$$

(5.3)

- Suppression function between adjacent agents is represented by $P_{i,j}$ for $S_i$ and $S_j$

- Stimulation function between SAG $S_i$ and NAG, $N_j$ is denoted by $M_{j,i}$

- New action vectors for a SAG $S_i$ is represented by $U^i$

- Define $T_a$ to be the activation threshold

- Define $N_s$ as the sensory neighborhood and $N_c$ to be the communication neighborhood.
- Define primary response of the immune memory by $R_p$ and secondary response by $R_s$.

- Represent $T_p$ as the time taken for the primary response and $T_s$ as the time taken for the secondary response.

- Represent the number of SAGs (lymphocytes) that bind the NAGs (antigens) during the primary and secondary responses as $N_p$ and $N_s$.

- Define the efficiency of the primary and secondary responses to be $\eta_p$, $\eta_s$ respectively.

- Define the binding period to be $t_b$.

- Let $A_i$ be the activation level determined by the number of mature actions within the binding period.

### 5.4.2 AISIMAM - Algorithm

This section represents mathematical derivation of the proposed model, AISIMAM.

Initialize all the parameters defined above

For each $S_i$

Calculate $M_{j,i} = f_1(A^j, B^i)$, where $B^i$ is the information vector of $S_i$, and $A^j$ is the information vector $\forall N_j$ in the sensory neighborhood $N_s$.

$$f_1(A^j, B^i) = \begin{cases} 0 & \text{if no } A^j \text{ in } N_s, \quad j = 1, 2, ..., M \\ \neq 0 & \text{if } \exists A^j \text{ in } N_s \end{cases}$$ (5.4)

If $M_{j,i} \neq 0$
The information about the $NAG$ is transmitted to the other $SAG$s through the immune network.

For each $NAG$ $N_j$, within the sensory circle $N_s$, where $j = 1 \cdots e$, and $e \leq M$

**Pattern Recognition and Identification**

Identify the $NAG$ using the identifier function $I$ that is given by

$$ I_j = f_2(A_j) $$ (5.5)

Generate possible new actions $U^1_j \ldots U^k_j$ using action generator function that is a function of $I_j$

$$ U^i_j = f_3(I_j) \text{ where } i = 1 \ldots k $$ (5.6)

**Binding Process**

Find the affinity for all possible vectors $U^i_j$ by the affinity function

$$ Af^i_j = f_4(U^i_j), \forall j = 1 \ldots k $$ (5.7)

**Activation Process**

Choose mature actions whose affinity is greater than activation threshold $T_a$ and store in the action set $Y$

$$ Y = \{ U^i_j | Af^i_j > T_a \} $$ where $j = 1 \cdots p$ (5.8)

The activation of the mature actions within the binding period $t_b$ is given by

$$ A_l = f_5(Y) *[u(t) - u(t - t_b)] $$ (5.9)

where $u(t)$ is the unit step response.
\[ f_s(Y) = \begin{cases} 0 & \text{if no mature action in } Y \\ \neq 0 & \text{if } \exists \text{ mature actions in } Y \end{cases} \]  

(5.10)

If a best action needs to be chosen, the threshold should be very high such that there is only one mature action left in \( Y (p = 1) \).

**Post activation processing - Cloning**

If \( A_i \neq 0 \) (There is an activation)

In this case, agents are reproduced with mature action set \( Y \) in SAGs. Self-agent \( S_i \) is cloned with mature action set \( Y \) to generate \( q \) SAG clones.

\[ S_c \text{ where } c = N + 1, ..., N + q \]  

(5.11)

where \( N \) is the number of SAGs in the system.

**Post processing - Memory**

Choose \( s \) number of memory agents \( S_m \) from the cloned agents

\[ S_m = S_c \text{, where } m = N + 1, ..., N + s, \text{ where } s < q \]  

(5.12)

**Memory Response**

The efficiency of the primary and secondary responses \( \eta_p \) and \( \eta_s \) are given by

\[ \eta_p = f(N_p, T_p) \]  

\[ \eta_s = f(N_s, T_s) \]  

(5.13)

Here, \( T_p \gg T_s \) and \( N_p \ll N_s \), where \( N_p \) and \( N_s \) are the number of actions required to kill the NAG in the primary response. \( T_p \) and \( T_s \) are the time taken for the primary and secondary responses respectively.

**Plasma Response**
Rest of the clones are defined as plasma agents $S_p$, where $p = N + s + 1, ..., N + q$. Here $q-s$ agents are added into the system.

Else $S_c = 0$ (There is no cloning, $A_p = 0$)

End If

End For

Else (There is no NAG stimulus)

Perform suppression by the suppression function

$$P_{i,j} = f_s(B', B^j)$$

where $i, j$ are of $S_i$ and $S_j$  \hspace{1cm} (5.14)

End If

End For

The following section deals with the need for the mathematical representation of the derived model.

5.4.3 Need for a Mathematical Representation

The goal of AISIMAM is to provide a mathematical representation of the immune system. Several immune modeling such as the immune network model [12], negative selection algorithm [1], mathematical modeling of the clonal selection and immune memory [21, 72], agent-based immune systems [73] exist in the literature. AISIMAM differs from the other models in the context of mathematical functions defined for the entire process. In order to prove the usefulness of the representation, two applications namely bar code recognition and mine detection are compared.

In the case of barcode recognition, assume that the non-self agents $N_j$ or antigens are the characters to be recognized. The $B$ cells are the software agents $S_i$ whose
information vector contains the corresponding ASCII characters. Each agent has a defined group of characters. Environment $E$ has the information about the recognized and the unrecognized characters. If the agent can recognize the character, recognition is achieved. Otherwise the agents can communicate through the environment to find if the unrecognized character falls into its category. The stimulus $M$ is defined by the recognition of the start bit pattern of the barcode that defines the start of the recognition process. The identifier function $I$ is a character recognition function. The affinity function $Af$ can be defined as the matching function between the recognized character and the character in the agent’s information vector. Affinity threshold $T_a$ can be set to 1 that chooses the best match. In this case cloning is not possible. Thus the agents are not reproduced. In this application, sensory and communication neighborhood is zero, since the agents are not in a space. Here, the self-agents are software agents.

In the case of mine detection application, non-self agents are the mines and the mobile robots are the self-agents. In this case, both the self and non-self agents are hardware agents. The sensory and communication neighborhoods are defined by any chosen distance metric. The identifier function $I$ becomes finding the mine by the identifier and the location of the mine. The affinity function $Af$ is the Euclidean distance. Affinity threshold $T_a$ can be set to a predefined value.

As can be seen above, the model can be applied to different applications by changing the functions. Therefore, the generalized functions provide a global representation for several agent based applications.
5.5 New Aspect of the Work

Literature survey shows that there are several applications on artificial immune systems and multiagent systems independently. Some of the recent work also addresses some of the properties of AIS to agent systems to solve a particular task [73, 74]. AISIMAM is a generic model that provides to define the SAGs and NAGs in terms of functions to be determined by the applications. Individual goals and a global goal for the agents can also be defined by the functions. The model is flexible and unique because the parameters of the model can be changed by the formulated functions depending on the application.

The following chapter discusses two applications of artificial immune systems.
6. Applications of Artificial Immune Systems

This chapter deals with two experimented applications of artificial immune system. The first experiment involves the application of the proposed model AISIMAM to a mine detection problem. The chapter introduces the mine detection problem, the importance of mine detection, different types of mines and existing detection techniques. Later section explains the application of AISIMAM to mine detection problem, results and the conclusions.

The second problem involves the application of the negative selection property of the immune system to a pattern recognition problem. The problem experimented is a color image classification problem. These sections introduce the classification theory, the color image classification system and the AIS based algorithm for classification and the experimental results.

6.1 Introduction to Mine Detection Problem

This section gives a brief introduction to the mine detection problem. Different types of mines and several methods of mine detection are also stated. The discussion also shows the importance of mine detection. Several algorithms are also being developed for effective mine detection. Later sections investigate artificial immune system based agent methodology applied to the mine detection problem.

6.1.1 Importance of Mine Detection

Mines are a significant barrier to a nation's economy and a threat to the social and economic development all over the world. In defense terminologies, mines are called *ordnances* [75]. Humanitarian efforts and advanced technologies are used for mine
detection and defusion. The task of disposing the mines and making it threat free, still remains a challenge to the scientists and engineers and to the department defense [76].

The reason for this as stated by Dick Davis from the department of defense is as follows [76].

Because mines are relatively inexpensive and effective for military missions, they have been attractive to the worldwide community. The use of mines is a growing problem as more mines are put in the ground faster than they can be removed.

According to the demining research at the University of Western Australia, researchers think it is better to think in terms of which areas are known to be affected by mines that are populated by the civilians, and which areas are believed to be affected but of no importance to the civilians, rather than thinking about the number of mines. United Nations states that the mines affect 70 countries and it is estimated that there are 10,000,000 mines still on land that needs 500 years of demining to clear. The locations of the mines are widespread under ocean and different parts of land [77].

The problem of mine detection becomes a political, military and humanitarian problem. There are several mine detection technologies that are invented. These reflect the effort and amount of money each country spends for this process.

6.1.2 Types of Mines and Mine Detection Methods

Mines can be practically made out of any substance [78, 79]. Normally, they are stuffed with materials like metal, wood, plastic etc. Normal classifications of mines include antipersonnel mines, blast mines, fragmentation mines, antitank vehicles and butterfly mines. The explosive used in explosive are mainly phosperous containing compounds. Usually chemicals like Tri-nitro-toluene (TNT), RDX, tetryl, PETN are used [80]. Most 11/21/02

Dept of Electrical Engineering
of the mine detection methods rely on odors released by the chemicals for detection. Since in a minefield the concentration of the odors may depend on a lot of factors such as wind direction, concentration of the mines deployed, the rate of false alarms that could be caused by this method is quite high.

Some of the available methods of mine detection are acoustic imaging, thermal imaging, photo acoustic spectroscopy, ion mobility spectrometry, impulse radar or microwave and surface penetrating radar, antibodies and genetically engineered sensors [81], metal detectors [82], odor sensors [83], specially trained dogs [84], honeybees with radio tags [85].

There are also several instruments and novel techniques being attempted to solve this problem. Microprocessors and digital signal processors are concentrated the most [82]. A detailed collection of web resources related to mine detection problem can be found in [86].

The following sections deal with the robots in mine detection and the agent theory applied to the mine detection problem. This provides the motivation for the application of AISIMAM and suggests an alternative solution to the mine detection problem.

6.1.3 Robots in Mine Detection
The problems posed by sensors in the mine detection process are getting narrower due to advancement in technology [87]. The trend of employing robots for mine detection becomes feasible with the availability of sensors with a good degree of accuracy and communication methods. Some of the main aspects during the deployment of robots for the detection process are their ability to distinguish between mines and non-mines, the time spent due to false alarms, their ability to operate under varied types of terrain and
climatic conditions, their ability to detect a variety of mines, their cost effectiveness in construction [87].

Several algorithms have been attempted for the mine detection purposes. Usually, robots are employed in mine fields where the location of the mines are unknown. A multitude of heuristic and random strategies have been attempted for this purpose. Most of these strategies depend on the minefield and the construction of the robots themselves. These strategies are designed to include the issues such as low cost, simple operation for the robots built, durability of the robots, quicker detection of the mines.

Some examples of mobile robots used in mine detection are lightweight robots like Pemex-BE, legged robots COMET-II, vehicle-mounted mine detectors (VMMD), suspended robots etc [78, 87]. Before employing any robot for the demining process, the terrain should be prepared artificially to the needs of the mechanical aspects of the robots. Some times this process requires removing of vegetation and other obstacles in the field, either by mechanical flails or human hands. When robots are used for mine detection the robots should also be instructed about the demining process.

Robots can be built with separate detection and removal strategies. Many of the robots are programmed to explode the mine when it detects it, but this could physically damage the robots. Mechanisms could be devised to carry the detected robot to a monitoring place where it can be safely exploded, but this demands special abilities for the robots for the removal process. Since mine detection is a tedious and sensitive problem, effective strategy algorithm and robust robot architecture should be designed when robots are employed for the process [87].
Efforts of scientists from Japan claim the world’s first landmine robot with an appearance of a spider [88]. It is a six-legged robot with precision sensors fixed on each leg to sense the mines. However, the promise of technology remains an unfulfilled dream because the actual requirement is a ‘no mine’ detector that can provide 100 percent reliability [77].

6.1.4 Agent Theories in Mine Detection

Research in many parts of the world has focused on the utilization of heuristic techniques for robots in the mine detection process. Researches have experimentally found that using multiple robots could be more effective in the detection process against a single robot in terms of cost effectiveness and time consumed [87]. Robots can be employed with central or distributed intelligence. In centralized intelligence, a robot could be partially or totally dependent on a central agency for feedback. This creates a master-slave relationship between the controlling agency and the robot entity. The central monitor could be humans or global positioning systems that can communicate and share the information directly to the robot. Traditional centralized control techniques are found computational intensive and lacks robustness and flexibility. Distributed control techniques are attempted as alternative methods [87].

The following section describes the motivation for experimenting the mine detection application.

6.1.5 Motivation for the Mine Detection Problem

A distributed control environment does not have any centralized control to monitor the individual robots. All the robots are independent in decision-making and execution,
constrained by certain control laws. It is basically in this regard that multi-agent studies or distributed artificial intelligence (DAI) comes into the mine detection domain [87].

Additionally, Gene Peresich from the Boeing Company states that AIS is a newer field of natural computation and describes an immune system method applicable for defense applications. The defense department is making efforts to use AIS phenomenon for such applications [89].

This motivates the application of AISIMAM to a mine detection problem since the model deals with the multiagent theory and the artificial immune systems. In future, the mobile micro robots can replace agents in the simulation and artificial immune systems can be used to detect the mines similar to antigens being detected and destroyed by the immune mechanisms.

6.2 Application of AISIMAM to a Mine Detection Problem

To experimentally verify the architecture, AISIMAM is applied to a specific problem. The problem implemented is mine detection and defusion. The experiment is simulated in MATLAB. The following section discusses the parameters of AISIMAM used for this specific application and the pseudo code for the problem.

6.2.1 Parameter Definitions

The following section briefly describes the characteristics of NAGS, SAGS and the environment.

6.2.1.1 NAGS and their Characteristics

The antigen or the non-self agent (NAG) is the mine. The area to be explored for detecting the mine corresponds to the environment in the model. This defines the
boundary for the environment for the agents to detect the mine. Mines are deployed in a uniform distribution within the environment. The initial locations correspond to the epitope or the receptor of the antigen. Characteristics of the mines are assumed to be stationary, unfriendly and competitive.

6.2.1.2 SAGS and their Characteristics
The $B$ cells are the self-agents (SAGS). Each SAG is assumed to be a mobile robot. All the SAGS are deployed in a uniform distribution within the environment. The initial locations of the SAGS correspond to the receptors of the $B$ cells. The SAGS have the characteristics of being itinerant, independent, cooperative, altruistic and deliberative. That is, the self-agents are mobile agents that work together, yet have an independent goal and are helpful to each other.

The environment is assumed to be an accessible environment in which the agent gets the complete information about the environment to make decisions. All the SAGS possess the capability to sense the mine and communicate between the agents within the sensory and communication circles respectively. Euclidean distance metric is used for both the sensory and communication circles. Each SAG (robot) recognizes the mine and identifies the location of the mine within this sensory circle. On identification of the $NAG$ (mine) each SAG communicates to the other SAGS in a Jerne’s network. For this problem, Jerne’s network is assumed to be a broadcast network. That is, the SAGS, broadcasts the information, so that all the SAGs can receive the information transmitted by the other. In the simulation, this is implemented by making the environment global. It is also assumed that the communication between the SAGS is greater than the capacity of every SAG to sense the $NAG$.
The robots or the self-agents are assumed to have an equal capability to perform the operation. In other words, all the robots have the same characteristics to perform the individual goal and have equal potential for the cooperative behavior amongst them to achieve the common goal. The same assumption holds good for the non-self agents or the mines.

6.2.2 Mine Defusion

Circling the mine is defined to be defusing the mine. The simulation assumes that four mines circle a single mine for defusion. Once the mine is defused it is removed from the environment matrix.

Application of AISIMAM to mine detection leads to three cases namely detection of mine, detection of no mine but detection of a SAG, detection of neither a mine nor a SAG. Figure 6-1 shows the flowchart for the mine detection problem. From Figure 6-1, it can be seen that on detection of mine within the sensory circle, each SAG identifies the location of the mine. On identifying the location, it generates eight possible locations around it and chooses the closest position towards the mine. Euclidean distance is used for calculating the distance. Simultaneously, every SAG informs the identified mine and its location to the other SAGs through the broadcast network. On reaching the mine, each SAG waits for three more SAGs for defusion. In the absence of the mine, each SAG looks for another SAG within the communication circle for mine information. If information about the mine is found the same sequence of steps are followed. If there are no mines or no mine information associated with the SAG, the SAGs make random movements detecting mines. This process is repeated until all the mines are defused.

The following section provides the pseudo code the mine detection problem.

11/21/02 Dept of Electrical Engineering
Deploy the mines and the robots within the environment

Non-self agent – antigen (NAG) – Mines - Non-self agent are deployed at in a uniform distribution (Antigens have epitopes of different shapes on its surface)

Self – agents – mobile robots are also deployed in a uniform distribution (SAG) (B cells generate antibodies that are receptors on its surface called the paratopes and idiotopes – B cells are mono specific)

Is there a mine within the sensory circle?

SAG identifies the location of the mine (determined by the identifier function)

SAG generates new action vectors - the 8 different locations to move

Calculate the distance D (affinity function) between the identified mine and generated SAG locations (Affinity maturation (Af) (Affinity between Ag - Ab)

SAG makes a random movement of one step from its current location

Broadcasts to the other agents within the comm circle, the detected mines and the corresponding locations

SAGs within the comm circle acknowledge the stimulus – meaning that it can come there and make movement towards the mine

Wait for 3 more SAGs

Defuse the mine

Memory is dummy here

Figure 6-1 Flowchart representation of the mine detection problem
6.3 Pseudo Code For The Mine Detection Problem

The pseudo code for the mine detection problem is as follows.

1. Initialize the SAGs and NAGs in a uniform distribution.

2. Defuse = 0; (Initially there is no defusion)

2.1 While (defuse $\neq$ number of mines, $N_i$)

2.2 For each $SAG S_j$, do the following

   If (there is a mine within the sensory circle)

   a) Identify the location of the mine

   b) Inform the locations of the mines to the other self-agents within the
      communication circle. This corresponds to the communication through the
      immune network.

   c) SAG generates new actions that are eight different new locations to move

   d) Find out the distance (affinity function) between these locations and mine
      locations. The Euclidean distance between the generated locations and the
      robot location calculates the affinity.

   e) Choose the distance that is lesser than an affinity threshold and move to
      that location.

   If (this location is the mine location)

      If (there are 4 SAGs around the mine)

   Defuse the mines, update the number of mines defused, (defuse = defuse + 1);

   If (defuse == number of mines),

   Break; End If; End While

STOP
Else wait until there are four SAGs around the mine; **End If**

Else **do step 2.2. c. End If**

**Else If** (there are any self-agents within the communication circle)

**If** (non-self information is available), repeat from step 2.2. **End If**

**Else**

Make random movements from the current location, since there is no NAG information from other self-agents and no mine detected within the sensory circle

**End If; End For; End While; STOP**

Memory is not used in this problem since there is no usefulness in remembering the location of the mine once it is detected and defused.

The following section explains the simulation details of the mine detection problem.

**6.3.1 Simulation Details**

The simulation assumes that prior knowledge of the minefield intensity is known in a given environment. In the simulation, this means that the number of mines in the given environment is known. Therefore, known number of mines is deployed in a uniformly distributed manner. This creates the minefield. A known number of mobile robots are deployed in a uniformly distributed manner in the environment.

In order to differentiate between the mobile robot and the mine the simulation uses a ‘+’ for a mine and a ‘o’ for representation. However, the code provides the mine by a suitable identification for the mine and the robot. A mine is identified by a ‘0’ and the robot by a ‘1’. The information vector for the mine and the robots contains the
initially deployed location information along with the identifier. Table 6-1 shows an example of the mine and the information vector for the robot.

Table 6-1 An example of information vector of mines and robots

<table>
<thead>
<tr>
<th></th>
<th>X coordinate</th>
<th>Y coordinate</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mine</td>
<td>4</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Robot</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

A sample environment vector is as shown in Table 6-2. The location of mines and robots with the identifier, and the number of detected mines along with their locations are shown in Table 6-2.

Table 6-2 An Example of the environment vector

<table>
<thead>
<tr>
<th></th>
<th>Coordinates (Initial)</th>
<th>Identifier</th>
<th>No of mines detected</th>
<th>Detected Mine locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mines</td>
<td>X</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>0</td>
<td>0,0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0,0</td>
</tr>
<tr>
<td>Robots</td>
<td>X</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>4,5</td>
</tr>
</tbody>
</table>

The simulation also sets the sensory and communication circles of the robot. The communication circle is assumed to be greater than the sensory circle.

A typical simulation screen is shown in Figure 6-2 and Figure 6-3. The results are shown for an environment of 10x10 with 2 mines and 8 robots. The sensory circle is three (sen_c = 3) and communication circle is five (c_cir = 5). Figures 6-2 shows the locations of mines and robots and new locations for movement after two iterations. Figure 6-3 shows four robots moved to the desired locations after two more iterations.

11/21/02 Dept of Electrical Engineering
Since there are four robots around the mine, the mine is defused in the following iteration. The next section explains the experimental results of the mine detection problem.

Figure 6-2 Initial locations of the mines, robots and the locations after 2 iterations

Figure 6-3 Four robots have circled one mine and the environment is updated.
6.3.2 Experimental Results and Analysis

The experimental results of mine detection problem are analyzed in terms of five parameters namely the number of mines, number of agents, size of the environment, sensory and communication circles. The simulations were conducted for the following variations.

- Variation in the environment sizes of 32 x 32 and 100 x 100.
- Constant sensory circle and variation in communication circle for a constant environment size.
- Simultaneous variation in sensory and communication circle for a constant environment size.
- Variation in the number of robots and number of mines simultaneously.

The results present the computational complexity of the AISIMAM architecture in terms of the rate of convergence.

Figure 6-4 shows the results for an environment size of 32 x 32 for 60 agents and 30 mines. The sensory circle is set to a constant value of three. The communication circle is varied between 5 to 9. It can be observed from Figure 6-4 that increasing the communication circle decreases the average rate of convergence.

Figure 6-5 shows the effect on the average rate of convergence with a simultaneous variation in both the sensory and communication circle. The environment is fixed to 32 x 32 for 60 agents and 30 mines. It can be observed that the average rate of convergence decreases with an increase in sensory and communication circle. This is because the robots can detect more area. At the same time the robots can also...
communicate to more number of robots due to larger communication circle. This results in better average rate of convergence.

Figure 6-4 Average rate of convergence vs. variation in comm circle

Figure 6-5 Average rate of convergence vs. variation in sensory and comm circle
For the same environment size of 32 x 32, the number of agents is varied from 10 to 70 in steps of 10. Similarly, the number of mines is varied from 40 to 100 in increments of 10. The experiment is repeated for different values of sensory and communication circle in pairs of (3,5), (5, 7), (7, 9) and (9, 11). The rate of convergence is an average of five iterations measured for every variation. Figure 6-6 shows that average rate of convergence decreases when the sensory and communication circles are increased.

![Figure 6-6 Average rate of convergence vs. sen & comm. circles & mines/robots](image)

The results of the proposed architecture are compared with the results of swarm intelligence based method [90]. The comparison is done for an environment of 100 x 100. The sensory circle and communication circles are set to 1.414 and 3.7. The number of robots is varied between 10 and 100 in steps of 10. Mines are varied between 10 and 200 in steps of 10. Figure 6-7 shows that increasing the number of robots improve the rate of convergence. The swarm intelligence model [90] experimented uses a rectangular
grid for sensing (1x1) and communication (3x3). Since AISIMAM uses a circular distance, a sensory and communication circle of 1.414 and 3.7 provides a common input for comparison of results.

![Diagram](image_url)

**Figure 6-7 Average rate of convergence vs. variation in robots and mines**

The results of the swarm intelligence model are shown in Figure 6-8. It can be observed from Figure 6-8 that AISIMAM converges faster than the swarm intelligence based model [90].

Even though AISIMAM outperformed the swarm intelligence model, its abilities are limited due to small sensory and communication circles. Therefore, AISIMAM is simulated for larger sensory and communication circles to study their effect on the rate of convergence. For an environment of 100 x 100, the robots and mines are set to 80 and 20. The sensory and communication circle are varied in pairs of (1.414, 3.7), (5,7) and (10,15). The results are shown in Figure 6-9.

11/22/02 Dept of Electrical Engineering
The rate of convergence falls drastically for a large increase in sensory and communication circles.

Figure 6-8 Average rate of convergence vs. variation in robots and mines

Figure 6-9 Rate of convergence vs. variation in sensory and comm. circle
6.4 Implementation of Binding Period Resolves Freezing

In the mine detection application, each mine is defused when four robots circle the mine. In this process, each robot waits for other three robots, after it has moved to the closest mine location. For a certain combination of robots and mines, every robot could be tied to a mine, but none of the mine has four robots. Since the waiting duration is limited, each robot will wait for other robots indefinitely. This process is called freezing. Freezing is a function of the number of mines and robots and their distribution in the environment.

This problem can be solved by employing the binding period property of the AISIMAM architecture. The simulation is repeated by incorporating binding period. By incorporating the binding period that is similar to the tolerization period in the immune system, every robot waits for a predefined number of steps after they find a mine. This process thus avoids freezing.

Resolving freezing also gives a measure of completeness of the model. This is measured by rate of success or failure of the model.

In the experimental setup for a 100 x 100 environment, freezing data is obtained. An average of five runs on each data set converted to percentage provides the rate of success or failure. Figure 6-7 shows the freezing shown by the flat surface in the plot.

The above results that AISIMAM solves the mine detection problem successfully. The following sections deal with the AIS based image classification problem.
6.5 An AIS based Color Image Classification Problem in a Real Time Industrial Application

This chapter deals with an AIS based pattern recognition and classification problem. The problem experimented is a color image classification problem. Initially, the concepts of classification theory and the other methods of classification are studied. Then an AIS based image classification algorithm is described. Genetic algorithm is applied to the algorithm. Thus, concepts of genetic algorithms are also studied.

Later sections of the chapter discuss the artificial immune system algorithm for color image classification and the experimental results are provided.

6.6 Introduction to Classification

Classification is the task of assigning or sorting data into categories or classes. The data depends on the type of input that needs to be classified. This depends on the application and the system to be evaluated. In generic terms, these data or objects are defined as patterns [91]. The pattern can fall into two categories namely abstract and concrete. Abstract patterns include ideas and arguments. Concrete patterns include characters, symbols, pictures, biomedical images, three-dimensional objects, speech waveforms and seismic data [92].

Pattern recognition is the scientific discipline whose goal is the identification of such patterns from different categories [93]. Recognition of abstract patterns is called conceptual recognition, belongs to a branch of artificial intelligence [92]. Classification is a part of pattern recognition.

Pattern classification can also be defined as an information transformation process. In other words, a relatively large set of mixed data is transformed into smaller
units of useful data. Classifier is the device or the process of implementing these classifications [91]. The following section explains the stages involved in a basic classification system. This consists of class definitions, classifier types and class descriptions.

6.7 A Pattern Recognition System

This section gives a brief description about the basic elements of the pattern recognition system. A basic classification system involves four stages [93, 94].

- An image or data of interest or input patterns to be classified
- A sensor to sense the input data
- Feature extraction
- Classifier system as a decision maker

Figure 6-10 shows the block diagram of a basic pattern recognition and classification system. As stated earlier, the set of data can be an object to be converted into an image for processing or a signal depending on the type of measurement. As can be seen from Figure 6-10, the first stage in any pattern recognition system is the sensor that acts as a signal converter and provides the information. Second is the feature extraction. Feature extraction is a process by which the attributes of the signal or the data are computed and collected in a fixed format. This process becomes an important aspect of the recognition task because irrelevant data is removed from the raw data pattern.

Once the relevant features are extracted, the classifier sorts the data into classes by a set of procedures. As shown by the block diagram, the stages of the classifier system are not independent. Depending on the results, it may be necessary to get back to
the previous stage and redesign the whole system to improve the overall performance of the system.

![Diagram of Pattern classification system](image)

**Figure 6-10 A Pattern classification system [58]**

The next section focuses on some of the class definitions and class descriptions that are necessary to follow the rest of the classification theory.

### 6.8 Class Descriptions

To start with, a *class* is defined as a set of predefined attributes or features. The *data* to be classified are structured as *vectors* in a region called *feature space*. In other words, every data point in the feature space is treated as a vector. Each component $X_i$, where $i = 1, \ldots, n$, of the feature vector $X$ is usually a feature, attribute or property of an object under analysis. These feature vectors in a given class occupy a region in the feature space called *class region*. Depending on how the feature vectors are spread over a class region, classes are divided into *separable* and *non-separable* classes. This is explained better by the *compactness hypothesis* [91].

Compactness hypothesis says that the feature vectors of a given class are in some sense nearer to all feature vectors in that particular class region than to most of the feature vectors in other class regions. That is, compactness is a measure of the closeness of the vectors within the class regions. In general, if the class regions do not overlap on each
other and are bounded then, such classes are defined as separable classes and supposed to possess the property of separability. If for every such class region, a hyper plane could be placed such that it separates each class region from all other class regions, then such classes are said to be linearly separable classes. On the other hand, classes in which, the feature region is not well defined and overlaps on the other class regions are called non-separable classes. Non-separable classes do not obey the compactness hypothesis [91]. In the next section, the general concepts of classifier theory are explained.

6.9 Classification Theory

Classification includes two stages.

- The first stage is to train the system with a set of data
- The second stage is to classify a new set of unclassified data.

Let $C$ denote a classifier and let $I$ denote the information received. Here training is the process by which the parameters of $C$ are adjusted in response to $I$. The classifier may be a machine or a human being. Practically, a training procedure is an algorithm that implements the training rules. Classification is related to how the system learns the training data.

Learning is the system’s performance from one level to the other. Learning is said to be positive if it is in the direction of increased effectiveness. Learning is associated with feedback and provides a means to automatically correct or adapt to changing environments. Such classifiers that can improve its performance in response to the information it receives as a function of time are called trainable classifiers. To be more specific, the classifier assigns every feature vector to a particular decision region $R_j$ in feature space by means of a set of decision hyper surfaces. Each such assignment may

11/21/02

Dept of Electrical Engineering
or may not correspond to the correct or desirable classification. Trainable classifiers are the ones which attempt to make the number of incorrect classifications small or zero by adjusting the set of decision region \( R_j \) where \( j = 1, \ldots, n \), in response to observations on a sequence of feature vectors \( W_j \). Along with the feature vectors, the observation may include information that correctly classifies these feature vectors [91].

During training, if the set of data were available, and the classifier was designed by exploiting this known information, then the training set is said to be supervised training. Bayesian networks are one of the methods to perform supervised classification [95]. However, this is not always the case. There are tasks for which the training data of known classes are not available, and for a set of feature vectors \( X \), the goal is to unravel the underlying similarities and group the vectors together. This is known as unsupervised training. This is also defined as clustering. In other words, the purpose of clustering is to obtain a partition \( P \) of a set \( E \) of \( m \) objects \( X_i \), by the use of the similarity measure [96].

A major issue in unsupervised classification is to define the similarity between the two feature vectors and to choose an appropriate measure for it. Similarity measure gives a numerical value to the notion of closeness between the objects.

With this background on classifiers, the next section deals with genetic algorithms followed by the AIS based classification problem.

6.10 Genetic Algorithms

Some of the features of the evolutionary theory intrigued John Holland, the inventor of genetic algorithms to appropriately incorporate the technique of evolution in computer algorithms to solve complex problems. They were invented to mimic some of the
processes observed in the natural evolution. Goldberg defines genetic algorithms as search algorithms based on the mechanics of natural selection and natural genetics [97].

The mechanisms of this evolution are not fully understood, but some of its features are known. According to Davis [98], the natural evolution theory undergoes three basic steps.

- **Evolution** takes place on chromosomes, organic devices for encoding the structure of living beings, rather than on the living beings they encode.

- **Natural selection** is the link between the chromosomes and the performance of their decoded structures. Processes of natural selection cause those chromosomes that encode successful structures to reproduce more often than those that do not.

- **Reproduction** that operates on chromosomes is the point at which evolution takes place. Mutations may cause chromosomes of biological children to be different from those of their parents, and recombination processes may create quite different chromosomes in the children by combining material from the chromosomes of two parents.

### 6.11 Description of Genetic Algorithm

From the theory of evolution, genetic algorithm is defined as follows [97].

- Initialize a population of chromosomes.

- Evaluate each chromosome in the population by means of a fitness function.

- Create new chromosomes by mating current chromosomes of better fitness.

  Apply mutation and recombination as the present chromosomes mate.
• Delete members of the population to make room for the new chromosomes of higher fitness values.
• Evaluate the new chromosomes and insert them into population.
• Repeat the process until the best chromosome is found out.

The following paragraphs will explain the components of the genetic algorithm where the chromosomes are represented by binary string patterns.

6.12 Crossover

Crossover is a very important component of genetic algorithm. Many researchers believe that if crossover is removed from genetic algorithm, it is no longer a genetic algorithm. In nature, crossover occurs when two parents exchange part of their chromosomes to create children. In genetic algorithm, crossover recombines the genetic material in two parent chromosomes to create children. The binary string pattern is treated as the genetic material for cross over resulting in reproduction of different binary string patterns. There are several kinds of crossover such as the one-point crossover, two-point crossover and uniform crossover in the literature [98].

One point crossover as experimented by John Holland occurs, when parts of two parent chromosomes are swapped after a randomly selected point, shown in Figure 6-11. Binary strings represent the two parents and the generation of children can be seen. Here, the last two bits of parent1 and 2 chosen at random are crossed over.

```
Parent1: 11 11 11 11 11
Parent2: 00 00 00 11 00

Child1: 11 11 11 00
Child2: 00 00 00 11
```

Figure 6-11 Example of one point crossover
One important feature of the one point crossover is that it can produce children that are radically different from their parents. Another important feature is that one point crossover will not introduce differences for a bit position where both the parents have the same value as shown in Figure 6-12. It can be seen that the bits 3 and 4 of the parents as well as the children are the same even after crossover. In this case, crossover does not help in creating diversity in children. This is an extreme instance that both parents are identical. To overcome this problem, two-point crossover is generally employed. This is similar to one point crossover except that this uses two switching points chosen randomly for crossover. This effectively increases the possibility of generating diversity in children even if the parent chromosomes are similar.

![Figure 6-12 Example of one point crossover](image)

Figure 6-12 Example of one point crossover

Figure 6-13 explains the two-point crossover. Sometimes it is possible that two-point crossover cannot combine to given unique children. In such cases, uniform crossover is preferred.

![Figure 6-13 Example of two-point crossover](image)

Figure 6-13 Example of two-point crossover

In uniform crossover, two parents are selected, and two children are produced. For each bit position on the two children, it is randomly decided as to which parent contributes its bit value to its child. In this method, a template determines the switching.
According to the template, each parent contributes its value in that position to the first child. The second child receives the bit value in that position from the other parent. Figure 6-14 explains the uniform crossover.

\[
\begin{array}{c|c|c}
\text{Parent1:} & 1001011 & \text{Child1:} & 1001101 \\
\text{Parent2:} & 0101101 & \text{Child2:} & 0101011 \\
\text{Template:} & 1101001 & \\
\end{array}
\]

**Figure 6-14 Uniform crossover**

### 6.12.1 Mutation

Mutation is a procedure when applied to bit strings, sweeps down the list of bits, replacing each by a randomly selected bit if a probability test is passed. Bit mutation has an associated probability parameter that is low. If the randomly chosen number happens to be the same as the value in the chromosome, mutation will be no use. As a result of this, many genetic algorithm practitioners use bit mutation to flip bits [97].

### 6.12.2 Elitism

If no increasingly fit individual has been discovered between generations, the elitist policy simply carries forward the fittest individual from the previous generation into the next. Elitism is used for solving the problem that the best member of each generation into the succeeding generation [98].

The following section describes an AIS based solution to image classification problem in detail.

### 6.13 An AIS Approach to a Color Image Classification Problem

In order to understand the computational abilities of the AIS, a problem whose results are already known with a different method is experimented. The problem experimented is 11/21/02

Dept of Electrical Engineering
the color image classification problem. Zhao and Sahin solved the color image classification problem using the same set of input images using minimum distance classifiers and radial basis function networks [8, 9].

In this research, our goal was to bring out a biological based solution using AIS for the classification problem. Artificial immune systems (AIS) possess properties such as self/non-self identification, (negative selection), pattern recognition that could be used for our problem.

In chapter II, the basic theory and concepts of the human immune system and the applications of AIS in several engineering applications were introduced. The following sections of the chapter present the research work done on the AIS based image classification problem. This include the overview of an image classification task, need for such color image classification task, AIS based image classification problem and the simulation results. During the research, it is also inferred that the efficiency of the algorithm could be improved by making the system more biological. Therefore, genetic algorithm is also applied along with the immune system properties.

6.14 Color Image Classification Problem

Image classification is the task of classifying the color images into pre-defined categories or classes. A common approach to an image classification task involves three issues namely feature extraction of the input images, training procedure for the obtained images and choosing a classifier to classify the images into desired classes [56, 58]. The color image classification system is shown in Figure 6-15. The sensor is a color camera that creates the input images. The second stage is the feature extraction. In the experiment, the average RGB color values are extracted from the images. Once the useful
information concerning the features of the object under study is obtained, the next stage would involve comparing that information with a set of known data about the object to find the best match. The average RGB color values are used for classification.

AIS based image classifier classifies the images into different classes. In this case, the decision of classifying is based on the property of negative selection of AIS and genetic algorithm.

Figure 6-15 A color image classification system

The system is evaluated with a set of test images and checked if they are classified correctly. The following section explains the need for such a color image classification problem.

6.15 Need for Color Image Classification

The primary objective for some computer vision systems is to automatically analyze multi spectral images of object surfaces. Upon identification of the object, it is then necessary to discover the best match from a set of known class models to implement the recognition task [9]. In addition to the knowledge of the classification procedure, it is also necessary to understand the need for such a classification task.

Color classification is a difficult problem for many types of objects. However, there are some applications where classification becomes an important task. For
example, identification of different types of wood by color becomes a difficult process in natural materials such as wood as they show random textures [9].

*American Woodmark Corporation* uses one such real time application to classify the wooden components of the kitchen cabinets. American Woodmark Corporation needs to classify finished wooden cabinets because during production the same wood type can be stained with different colors. Figure 6-16 illustrates the experimental layout of the typical color and species classification system. This automatic vision based system is developed at Virginia State and Polytechnic University.

![Diagram of a color and species classification system](image)

**Figure 6-16 A color and species classification system**

The images are captured by a color camera and processed by the PC. The JVC TK-1070U color camera captures the red, green, blue (RGB) images of the samples taken from the moving wooden components on a conveyor [9]. A frame grabber, located in a 486-based PC, takes these images. After the image is captured, the program finds which color and species the image sample belongs to and classifies the wooden images into that class. In other words, the various classes according to the color and type of wood are classified components of wooden images

Components of wood

Conveyor

Color camera

Classified components of wooden images

11/22/02

Dept of Electrical Engineering
predefined and are a part of the training information that is used during classification. The components move on the conveyor without pre-classification of the wood. The finished wooden components are of different types or painted or stained in a number of colors. There is sometimes only a slight difference between the colors. The variation in the texture and the grain of the wood create slight, yet noticeable differences between the species of wood. These woods of all different colors and species can be together on the conveyor. Since there are a great number of component types, and many of them are very similar in appearance, it is tedious, labor-intensive, and error-prone work to classify these components manually [9]. In such cases, it is sometimes difficult even for human eyes to accurately identify and categorize the finished product. For these reasons, it is desirable to develop a machine-vision system to improve the inspection process, creating a simple, accurate, and cost-effective classification system.

The database of 480 images, grabbed by an automatic machine-vision system, has been gathered from the Spatial Data Laboratory at Virginia Tech. These images are stored in raw format. They have only color information of each pixel. Since the raw formatted files have no header and size information, it is very easy to use these files in a C++ or a MATLAB program [9]. The image structure can be seen in Figure 6-17.

```
Figure 6-17 Structure of the images
```

11/22/02 Dept of Electrical Engineering
Every image has 16,384 (64x256) pixels. Every pixel has red, green, and blue components. Every pixel is represented by 24 bits (3 bytes), eight for each color and therefore the total image size is 49,512 bytes (16,384x3).

6.16 An AIS based Image Classifier

In color image classification, color images of woods are the patterns to be classified into specific classes. The AIS based classifier is designed to classify these images according to the color and type of wood into pre-defined classes. The AIS based classifier performs unsupervised classification.

In AIS based image classification task, there are sets of sample images and test images. The sample images are used to train AIS to obtain a set of antibodies. The test images to be classified correspond to the antigens in the immune system. The obtained images from training correspond to the antibodies. That is, these images are similar to the complementary receptors of the B cells. These images are used to match the test images in a similar way of complementary match between the epitopes of the antigen and the idiotypes of the B cell.

The color information is a source for the image classification task. For every sample image, the average of the red, green and blue is found. The average value of each color is converted into eight bit binary strings. Each image is then represented by the 24-bit binary string pattern, eight bits for each color. In AIS classification, 100 detectors, each of 24-bit length is randomly generated. These binary strings are matched with each sample image to train the AIS.

The method of classification involves two stages. First stage involves the training of the images and the second involves the classification of the images that belongs to a 11/21/02

Dept of Electrical Engineering
particular class. The sample images contain 8 classes, with each class having $N_i$ sets of images where $i = 1 \cdots 8$. In the first stage, a set of randomly generated binary strings is created. The sample images are converted into binary strings. Ninety-five out of 480 images are used as training samples while the remaining 385 images are used to test the system. All the sample images are grouped into eight different classes. In other words, the class in which each image falls is known. Since the training samples have to represent the classes optimally, the same training samples as in [8, 9] are used. Additionally, this provides an accurate comparison between our method and the methods in [8, 9]. The sample images chosen for experiment are those of different colors and of different types of woods. Each image is classified into a set of classes according to the type of wood and color. Figure 6-18 shows the sample images used for training the AIS.

![Sample Images](image_url)

**Figure 6-18 Eight classes of wood used to train the AIS**
In AIS image classification algorithm once the training is complete, the test images are given for classification. There are two methods by which the training is performed. One is the matching algorithm and the second is using genetic algorithm. They are explained in detail in the following sections. However both the training procedures include the negative selection algorithm as the outline for training.

6.16.1 Training using Matching Algorithm and Negative Selection

The negative selection algorithm as summarized by Dasgupta [1] is as follows.

- Define self as a collection of strings $S$ of finite length $L$ that needs to be monitored.
- Generate a set of detectors $R$ each of which fails to match any string in $S$.
- Monitor $S$ for changes by continually matching the detectors in $R$ against $S$. If any detector matches, then a change is known to have occurred, because the detectors are designed not to match any of the original strings in $S$.

In this method, the negative selection property uses matching algorithm to train the AIS. The sample images are treated as self-cells ($B$ cells). This means that they belong to the training set. Non-self cells (antigens) are the test images. Sets of randomly generated binary strings are created. This corresponds to the antibody generation in the immune system. This corresponds to the detectors in the negative selection algorithm.

Then, using the matching algorithm, every image of each class is matched with a set of randomly generated antibodies. The antibodies that have the matching value a greater value than a set threshold are selected. The matching threshold is similar to the affinity threshold in the immune system. This results in a set of antibodies for each sample image. Hence, for each image of each class, we obtain $M$ randomly generated

11/21/02

Dept of Electrical Engineering
antibodies that have a matching value more than the threshold. This generates the set of antibodies that are to be compared with the particular antigen (test image). The AIS based image classification algorithm is as follows.

Define self as $S_i$, where $i = 1 \ldots 95$ as a collection of sample images. Each sample image is represented by the binary values of the average red, green and blue color values of the image, eight bits corresponding to each color value. For example, the binary representation of the first image is shown in Figure 6-19.

Figure 6-19 Binary representation of an image

$$00001100\text{(Red)} - 00001011\text{(Green)} - 00001011\text{(Blue)}$$

Generate a set of detectors $R_j$, where $j = 1 \ldots 100$. These randomly generated binary strings are of equal length as the sample image. In this case, 100 random strings each of 24-bit length are generated. These detectors correspond to the antibodies in the immune system.

Match $S_i$ with every $R_j$ and find the matching between the strings. The XOR operator achieves the detection of mismatch. This expresses the function of negative selection where cells that react with self are ignored and the ones react with the non-self cells are chosen. Here, the bits in $S_i$ that match with the detector $R_j$ are ignored and the bits that have a mismatch are chosen.

The length of the mismatch in the bits is counted and the matching score $S_m$ is found out by Eq. (6.1). $C$ in Eq. (6.1) is the count or length of the total number of
mismatches in the string and \( l \) represents the total number of count of consequent mismatches in the bits.

\[
S_m = C + \sum_{i=1}^{C} 2^i
\]  

(6.1)

The strings \( N_k \) where \( k = 1 \cdots 20 \) with a matching score more than set threshold chosen and the rest are rejected. The matching score is used as the similarity measure for clustering.

This process is repeated for all the 95 sample images. When the process is completed for all the sample images and the strings are chosen according to the matching values, the training is said to be complete with a set of antibodies for each of the sample images. An example of the matching value calculation is shown in Figure 6-20. The calculation is shown only for eight bits. The matching is performed in three sets of eight bits for all the three-color values and the process is repeated for all the sample images.

| Sample image | 1 1 0 0 0 0 1 |
| Detectors    | 0 1 0 1 1 0 1 0 |
| XOR          | 1 0 0 1 1 0 1 1 |
| Count        | 1 2 2 2 = 5 |
| Matching Score | \( 5 + 2^1 + 2^2 + 2^2 = 15 \) |

**Figure 6-20 An example of the matching algorithm**

Once the training is completed, the test images are given to the algorithm and they are classified. To verify the efficiency of the algorithm, first the sample images used for training are used as test images. Secondly the test images are classified. The percentage of correct classification according to the matching algorithm for the sample as well as for the test images are 16.61% and 9.61%.
The algorithm yielded a very poor percentage of classification for the sample images. Therefore, the test images yielded a lesser percentage of correct classification. This procedure had the major limitation of generating valid detectors due to the limited size of the randomly generated detectors. The algorithm did not ensure that the generated detectors are not repeated in the process, hence resulted in a poor percentage of correct classification. Therefore, a better approach to solve the problem was needed and genetic algorithm was explored.

6.16.2 AIS based Genetic Algorithm

Using genetic algorithms, a set of antibodies for each antigen or the test image (treated as antigen) to be classified is generated. Once the training is complete, the average RGB values of the test images in also converted to binary sequence and matched with the trained AIS for correct classification.

In this method, the genetic algorithm approach is used. This is an iterative process that involves, generation of initial population, which is the set of random strings corresponding to the images to be classified. The fitness function is the Euclidian distance between the average RGB values of the sample images and generated initial population. The fitness values that are greater than a set threshold are chosen and are treated as the new pool. Crossover is employed to generate the children pool. The process is repeated till the fitness values are lesser than a set threshold. Reiterate the procedure to find the best matches for each sample image. This completes the training with the best matching values for all the sample images. In our case, the average RGB values of the sample and test images are represented in binary strings that represent the chromosomes. The algorithm works as follows.

11/21/02 Dept of Electrical Engineering
Randomly initialize the gene population of binary strings that are equal in length to the sample images. Evaluate the fitness of each string with each sample image.

The fitness function is the Euclidean distance between the average RGB values of the sample images and the generated initial population. The Euclidean distance is calculated as shown in Eq. (6.2)

\[
\text{Euclidean distance} = \sqrt{(G - X_v)^2}
\]  

(6.2)

Where \( G = \sum_{m=1}^{n} g(m : n) \), and \( g \) is the decimal equivalent of every eight bits of the randomly generated population and \( X_v = \) average \( r, g, b \) of the sample image. Once the fitness is found, the strings are sorted according to the ascending order of the fitness values. The sorted population undergoes reproduction. The fitness values obtained by the Euclidean distance provide the similarity measure.

An example of the fitness selection for a sample images and a few iterations with which the fitness is less than the threshold are as shown below. As can be seen the fitness values reduce towards the set threshold. The same procedure is repeated for all the sample images. The binary representations of the chosen images corresponding to the fitness values are shown in Table 6-3.

Reproduction is achieved by crossover. The sorted binary population undergoes crossover to produce new string values. Crossover here is implemented genetically, that is, randomly, as it would happen in nature. In our case, two-point crossover is employed.
Table 6-3 Fitness values in six iterations and the corresponding images

<table>
<thead>
<tr>
<th>Image 1 Threshold = 5</th>
<th>Fitness Values</th>
<th>Five antibodies for image 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Representation of the image - 1</td>
<td>Iterations</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>7</td>
</tr>
</tbody>
</table>

The generated new strings along with the parents are evaluated for fitness and the best two of the four binary strings are chosen. This is an iterative process until the best binary strings for the total population is found. The binary strings whose fitness values that are lesser than a set threshold are chosen to be the new population. Cross over the chosen population to generate the children pool again. Repeat the process till the fitness values are lesser than a set threshold. Reiterate the procedure to find the best matches for each sample image. This completes the training with the best fitness values for all the sample images. In the simulation, five best images for each sample image are chosen.

An example of this process is shown in Figure 6-21. It shows the initial population represented by 8 bit binary strings. The values shown beside each string are the fitness values of each string. The second block shows that the bit patterns sorted according to the fitness values. Then they undergo crossover. The example shows a four bit crossover, which means that four bits of each bit string, is crossed over with the four bits of the other binary sequence. A comparison of the AIS negative selection and an equivalent analogy of the classification are provided for clear understanding of how both...
are analyzed. The algorithm is also tested with the sample images as test images to check the efficiency of the algorithm.

![Image](image.png)

**Figure 6-21 Initial population and four-bit are crossed-over in the eight bits**

Table 6-4 and Table 6-5 compare the negative selection algorithm and the AIS based image classifier.

### 6.17 Threshold Setting

In the image classification algorithm, binary strings whose fitness is more than a set threshold is selected as the best fitness value. The threshold is experimented in two methods.

**First method:** This method sets the threshold according to Eq. (6.3)

$$Threshold = \sqrt{D_r^2 + D_g^2 + D_b^2}$$  \hspace{1cm} (6.3)

Where $D_r = C_r \times (0.001), D_g = C_g \times (0.001), D_b = C_b \times (0.001)$ and $C_r, C_g, C_b$ are the average values of the red, green and blue values of each image.
Second Method: This method sets the threshold to find the closest fitness values. The threshold was set according to the trial and error method by changing the values of the threshold between 1 and 5.

Table 6-4 Negative selection based and image classification algorithm

<table>
<thead>
<tr>
<th>Negative Selection Algorithm</th>
<th>Negative Selection Algorithm for Image Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define Self as a collection of randomly generated strings S of length L</td>
<td>S_i, i = 1...95 - sample images.</td>
</tr>
<tr>
<td>Generate a set of detectors, R, each of which fails to match any string S.</td>
<td>S - set of programs, files or software</td>
</tr>
<tr>
<td>Monitor S for changes by continually matching the detectors in R against S.</td>
<td>Generate R_j, where j = 1...100, of equal length as the sample images</td>
</tr>
<tr>
<td>If any detector ever matches, then a change is known to have occurred because the detectors are designed not to match any of the original strings.</td>
<td>Matching algorithm chooses N images that have the maximum mismatch values.</td>
</tr>
<tr>
<td></td>
<td>Same process is repeated for all 95 sample images – antibodies corresponding to all images</td>
</tr>
</tbody>
</table>

Table 6-5 AIS and genetic algorithm based image classifier

<table>
<thead>
<tr>
<th>Genetic Algorithm</th>
<th>AIS and genetic algorithm for Image Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human system has random gene pool generation.</td>
<td>Generation of the image pool corresponds to the initial antibody population.</td>
</tr>
<tr>
<td>On breeding it produces children pool &amp; genetically it is crossed over.</td>
<td>Apply crossover to generate images</td>
</tr>
<tr>
<td>Fitness of the produced children</td>
<td>Find the fitness of the initial population with the sample images</td>
</tr>
<tr>
<td>It is possible that children genes may be better than the parent genes measure by fitness of the individual.</td>
<td>Cross over the generated initial pool to get the children pool of images. Repeat the process and generate the children pool till the fitness value is lesser than a particular threshold.</td>
</tr>
</tbody>
</table>
Figure 6-22 shows the flowchart of the algorithm.

Create the initial population
- Generate random numbers - images

Find the fitness - Fitness is the Euclidean distance between each sample image and the initial population

Sort and select the first 100 closer or better matches

Cross over the selected population genetically (Randomly)

Treat the second population generated as the initial population

Nth fitness < threshold

Choose the best L matches for a single image and store

Repeat for all the sample images

Introduce the test images, find the fitness with the chosen images by training, and classify each image into the corresponding class

Figure 6-22 Flowchart representation of the AIS based image classification algorithm
6.18 Classification Results

Once the training is complete, the AIS is ready to classify the images. In this stage, we apply the test images (antigens) to the system. This test images are matched with the set of images of best fitness (antibodies) obtained during the training. The algorithm used training is repeated for the test images. Among the trained data, the antibody for the maximum matching or the best fitness is determined. Finally, the image is classified into the class that has the maximum matching value or best fitness value. Choosing the best fitness was done in two procedures: the minimum and the average fitness of the selected antibodies. In our case, AIS is trained using 95 sample images and 385 are tested for correct classification. The percentage of correct classification $C$ is calculated using Eq. (6.4)

$$\%C = (N_c + N_i) \times 100$$  \hspace{1cm} (6.4)

Where $N_c$ is the number of correctly classified images and $N_i$ is the total number of images to be classified.

It is seen that matching algorithm resulted in a poorer percentage of correct classification. Genetic algorithm improved the random number generation without increasing the size of the total number of detectors. It was possible to achieve this by implementing crossover. Crossover also helped in increasing the number of combinations of the detectors without actually increasing the random detectors database thus saving memory.

To identify how well the system was trained and able to classify, the experiment was repeated with the sample images as the test images.
Table 6-6 shows the percentage of correct classification obtained using Euclidean distance method by setting the threshold as defined by method I.

Table 6-6 Percentage of correct classification using the Euclidean distance method (Average and Minimum methods)

<table>
<thead>
<tr>
<th>Input</th>
<th>Euclidean Distance</th>
<th>% of error</th>
<th>Method - I</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Minimum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Image</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30.1</td>
<td>22.0</td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>30.6</td>
<td>22.5</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>31.6</td>
<td>27.2</td>
<td>3%</td>
<td></td>
</tr>
<tr>
<td>28.0</td>
<td>26.2</td>
<td>4%</td>
<td></td>
</tr>
<tr>
<td>25.1</td>
<td>25.45</td>
<td>5%</td>
<td></td>
</tr>
<tr>
<td>Sample Image</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22.1</td>
<td>34.74</td>
<td>1%</td>
<td></td>
</tr>
<tr>
<td>20.0</td>
<td>32.63</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>16.8</td>
<td>30.53</td>
<td>3%</td>
<td></td>
</tr>
<tr>
<td>18.9</td>
<td>26.32</td>
<td>4%</td>
<td></td>
</tr>
<tr>
<td>20.0</td>
<td>16.84</td>
<td>5%</td>
<td></td>
</tr>
</tbody>
</table>

Table 6-7 shows the results of the same algorithm with the threshold setting defined by method II. For the same average red, green and blue values of the images, the traditional RBFN of [4] generated 55.06% of correct classification. However, it is observed that this system involves only one property of AIS. By applying more properties of the immune system, better results can be obtained.

Table 6-7 Results of AIS and genetic algorithm based image classification algorithm

<table>
<thead>
<tr>
<th>Input</th>
<th>Method II Euclidean Distance, Threshold = 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average (%)                  Minimum (%)</td>
</tr>
<tr>
<td>Test</td>
<td>35.5844                      41.8188</td>
</tr>
<tr>
<td></td>
<td>35.3247                      44.9351</td>
</tr>
<tr>
<td>Sample</td>
<td>35.7895                      85.2632</td>
</tr>
<tr>
<td></td>
<td>36.8421                      82.6147</td>
</tr>
</tbody>
</table>
6.19 Summary of Results

The simulation results of the AIS and genetic algorithm based methods are closer to the results obtained by the earlier researchers for the same set of images. Hence, AIS application provides scope for future work. It can also be inferred that by adding some properties of AIS and genetic algorithm the system can be made more efficient. A approach to a color image classification problem that is useful in a real time industrial application is evolved.
7. Future Work

7.1 Introduction

This chapter discusses the future work in the AIS based intelligent multiagent model (AISIMAM) and the mine detection problem followed by the negative selection based image classification problem.

Improvements in AISIMAM

AISIMAM has the following features of the immune system namely pattern recognition by self/non-self discrimination, affinity maturation, immune network, decentralized control and global problem solving. AISIMAM also addresses the issues of immune memory, memory responses, clonal selection and expansion and reinforcement learning mathematically. However, the above-mentioned processes can be elaborated in future. The following paragraphs explain the possible future work in the model.

7.2.1 Immune Memory

Currently, AISIMAM contains memory in the form of memory agents that remember the mature action for a particular NAG. The characteristics of the memory agents are different from the SAGs. The model also views the co-stimulation of the immune network as the random actions performed by the SAGs in the absence of the NAG. However, there is scope in AISIMAM, for further study of the memory response by applying the other two sources for immune memory. This can be achieved by defining the property of the memory in agents. The effectiveness of the memory can also be analyzed by incorporating these features in the current model. The characteristics of the memory agents to be included are as follows.

11/21/02 Dept of Electrical Engineering
• Longer lifetime of the memory agent
• Re-stimulating the memory with predefined memory duration
• Retaining the presence of the NAG in number or by percentage of actions that is defined by the application.

Memory can also be a part of the SAGs instead of being memory agents. In other
is, once the global goal is achieved, SAGs will remember the actions performed to
destroy or achieve the NAG.

An application for the memory needs to be simulated to verify the model, since
the experimented application namely the mine detection problem does not involve
memory.

7.2.2 Reinforcement Learning in Immune System

Immune learning is an associated property of immune memory [1]. The following issues
can be analyzed in the model with respect to learning. An experimental analysis for the
mathematical representation of the primary and secondary responses of the immune
system needs to be performed. In addition to the application, the following parameters
can also be studied.

• The efficiency of the primary and secondary responses in terms of the response
time for the SAGs to react to the NAGs.
• The relation of rate of convergence of the algorithm or the computational
complexity with and without response time.
• Success/failure rate analysis of the algorithm in reaching the desired goal.
A mathematical representation for the cross-reactive response or the associative memory can be modeled. The effect of cross-reactive response in relation to the primary and secondary responses is an important issue for the model in solving a given problem. The importance of the cross-reactive memory can be understood by exploring the agent applications such as object recognition by a mobile robot in the field, shape identification by the micro robots in industry or image classification of objects with similar features. The following paragraph describes the features of self-organization to be researched and included in the model.

### 7.2.3 Self-organization

Another similarity observed between the immune system and the multiagent system is self-organization. Self-regulation or self-organization essentially deals with the property of distributed control [1] in both the immune system and the agents. This property essentially involves the capability of the SAGs to decentralize the control amongst them for a given NAG. By varying the characteristics of the agents such as the self-sufficiency or autonomy, the intelligence with which the agents makes the decisions or rationality and its level of cognition, the effect of the immune response can be analyzed. In other words, the characteristics of the agents that contribute to the distributed control and communication or self-organization exhibited by the SAGs can be studied.

It can also be found in the literature that the agent’s organization and adaptation to the environment is an optimization problem. AISIMAM can be verified as an optimization algorithm with the environment $E$ as the search space and the objective function to be the $NAG$.
7.2.4 Immune Network

According to Jerne’s hypothesis [24, 25, 26, 27], B cells in the immune system work in a network. AISIMAM also possess an agent network with which the SAGs communicate the information about the NAGs. However, the immune modeling in the AISIMAM is al. That is, the information processing performed by the idiotopes and paratopes are modeled separately. AISIMAM views both of them as a single entity namely the information vector. In future, modeling these features separately can make significant contributions to the efficiency of the model. This is because such a feature increases the ability or intelligence of the SAGs in terms of the information about the NAGs. A general mathematical representation of the immune network would also be a key addition to the model. Representing the agent network by a generalized function would provide an easy approach to defining the different types of communication network that will be used for a specific application.

7.3 Issues in Mine Detection Problem

Further conclusions can be arrived from the following additions. The present simulations of the mine detection application assume that the robots themselves do not get destroyed in the defusion process. But in practice, a robot can fall on the mine during deployment. In future, the algorithm can be modified to analyze the case of robot falling on a mine while deployment. NAGs in this application are assumed to be static. But in other applications, NAGs could also be dynamic and hence agent behavior in such a case can be studied.

In the mine detection application, memory is not used. This is because, there is no usefulness in remembering either the location information of the mine or the type of

mine itself since the algorithm defuses any kind of mine irrespective of the position and type. In future, the application can be redefined more specifically, by employing different functions for different kinds of mine. In this process, memory will be helpful in remembering the information about the type of mine that could be useful rather than the ion.

**AIS based Image Classification Problem**

There are a number of areas that can be addressed in the future for the AIS based image classification problem. At present, the algorithm uses only the property of negative selection. There are other valuable properties of AIS such as the primary and secondary response and associative memory of the immune system that can be employed. The primary and secondary response helps in learning the existing population faster. Associative memory helps to associate the new test image with the existing population instead of generating a new population for that particular image.

The property of elitism in genetic algorithm would help in storing the string that has an exact match with the test image. Mutation would help in speeding up the process of finding the best match. This can be achieved by randomly changing the bit pattern of the image from the selected images derived by training.

Currently, AIS based image classification approach employs only the average RGB values for feature extraction. In future, other features of the image like histogram and covariance matrices can be employed and results can be compared with the standard methods.
8. Conclusions

This thesis concentrates on artificial immune systems and multiagent systems. The research involves two problems. The first problem is a color image classification problem. A new AIS based genetic algorithm is proposed to solve the problem. The

ts are compared with the radial basis function approach that was experimented with
ame input images.

The second problem of research draws a generic model named AISIMAM, which is based on artificial immune system applied to multiagent systems. An application for the model is simulated. The problem experimented is mine detection and the results of the simulation are shows that AISIMAM is successful in solving the application. This proves that the model can be experimented on agent-based applications. The motivation for this application is that in future the mine detection can be performed efficiently by deploying mobile robots that have enough intelligence, communication and coordination to detect and defuse the mines. To verify the generality of the model, more applications would have to be simulated and verified in the future.

This research is conducted with the support of Gleason R&D Fund in Multi-agent Bio-Robotics Lab (MABL).
9. References


11/21/02

Dept of Electrical Engineering


[33] Starlab, URL: http://www.starlab.org/genes/AIS.


11/21/02 Dept of Electrical Engineering


[75] www.ornl.gov/Press_Releases/archive/mr19970709-00.html


[77] www.mech.uwa.edu.au/jpt/demining/map.html

11/21/02 Dept of Electrical Engineering


www.doe.carleton.ca/~jknight/LandMines/weplinks.htm
www.onr.navy.mil/sci-tech/ocean/jcm/cimmd.html

[84] www.humanitydog.se/minedog.html
[86] www.rit.edu/~sxs4446/resources/mine.htm


[88] http://jin.jcic.or.jp/nipponia/nipponia13/sp05.html


11/21/02 Dept of Electrical Engineering


10. Appendix A

This contains the CD ROM with the source codes for both the problems experimented.

- *AIS* based color image classification problem
- *AISIMAM*- An artificial Immune System based Intelligent Multiagent model and its application to a mine detection problem.