Immune-Inspired Self-Healing Swarm Robotic Systems

Amelia Ritahani Ismail

Submitted for the degree of Doctor of Philosophy

University of York

Department of Computer Science

September 2011

Abstract

The field of artificial immune system (AIS) is an example of biologically inspired computing that takes its inspiration from various aspects of immunology. Techniques from AIS have been applied in solving many different problems such as classification, optimisation and anomaly detection. However, despite the apparent success of the AIS approach, the unique advantages of AIS over and above other computational intelligence approaches are not clear. In order to address this, AIS practitioners need to carefully consider the application area and design methodologies that they adopt. It has been argued that of increasing importance is the development of a greater understanding of the underlying immunological system that acts as inspiration, as well as the understanding of the problem that need to be solved before proposing the immune-inspired solution to solve the desired problem. This thesis therefore aims to pursue a more principled approach for the development of an AIS, considering the application areas that are suitable based on the underlying biological system under study, as well as the engineering problems that needs to be solved. This directs us to recognise a methodology for developing AIS that integrates several explicit modelling phases to extract the key features of the biological system. An analysis of the immunological literature acknowledges our immune inspiration: granuloma formation, which represents a chronic inflammatory reaction initiated by various infectious and non-infectious agents. Our first step in developing an AIS supported by these properties is to construct an Unified Modelling Language (UML) model agent-based simulation to understand the underlying properties of granuloma formation. Based on the model and simulation, we then investigate the development of granuloma formation, based on the interactions of different signalling mechanisms and the recruitment of different cells in the system. Using the insight gained from these investigation, we construct a design principles to be incorporated into AIS algorithm development. The design principles are then instantiated for a self-healing algorithm for swarm robotic systems, specifically in the case of swarm beacon taxis when there exist failure of robots' energy in the systems. The self-healing algorithm, which is inspired by the granuloma formation of immune systems is then tested in swarm robotics simulation. To conclude, we analyse the process we have pursued to develop our AIS and evaluate the advantages and the disadvantages of the approach that we have taken, showing how a more principled approach with careful consideration the application area can be applied to the design of biologically-inspired algorithms.

Contents

List of Tables

List of Figures

List of Algorithms

Acknowledgements

Having received so much advice and support during this research, it's hard to know where to begin with the thanks when everyone's input was so enthusiastic and valuable. Heartfelt thanks to everyone who helped me to make this thesis possible. My biggest debt of gratitude must go to my husband for support, childminding and somewhat baffled encouragement; to my supervisor Jon Timmis for handling me, providing the inspiration for this work for four years; to Alan Winfield for the incredible effort, time and support for giving ideas on swarm robotic systems; to my family for always believing in me; to all my friends for ideas and support.

Declaration

Parts of the research reported within this thesis are based on my previous presented publications:Ismail & Timmis (2009), Ismail & Timmis (2010), Timmis et al. (2010b), Ismail et al. (2011) and Ismail & Timmis (2011).

CHAPTER 1

Introduction

A biologically-inspired system is one that has been designed with inspiration drawn from biological systems (Forbes, 2000). The system is intended to replicate the properties of a biological system with the aim of delivering new perspectives and solutions for engineering and computational problems. Bio-inspired algorithms are a set of algorithms inspired by the biological systems in solving engineering and computational problems, which is the main aim of this thesis. In particular, the aim of this thesis is to develop a novel immune-inspired algorithm for self-healing swarm robotic systems, that is capable of dealing with certain failure modes. To achieve this aim we propose a novel immune-inspired algorithm inspired from the process of granuloma formation in immune systems that is developed using the conceptual framework (Stepney et al., 2005) approach as a methodology in developing the algorithm.

This chapter is organised as follows: Section 1.1 offers some motivation for this thesis, along with the a brief overview of biologically-inspired computation, artificial immune systems and approaches for bio-inspired algorithm design. Following this, we highlight the issues in swarm robotics in section 1.2. Finally, we provide the structure and content of the thesis, stating the goal and research questions, as well as the publications produced whilst working towards this thesis in section 1.3.

1.1 Motivation

This section presents a summary of the current state of research in biologically-inspired computing and artificial immune systems as well as highlighting the issues in swarm

robotics that motivated the work undertaken in this thesis.

1.1.1 Biologically-Inspired Computation

The field of biologically-inspired computation has a twofold definition (Forbes, 2000)

- the use of biology or biological process as metaphor, inspiration, or enabler in developing new computing technologies and new areas
- the use of information science concepts and tools to explore biology from different theoretical perspectives

The inspiration that biology provides for computer engineering has resulted in a wide range of bio-inspired computing techniques, which are applied to a wide range of computational and engineering problems. Examples of these techniques are: evolutionary computation and genetic algorithm (Mitchell, 1998); artificial neural networks (Gurney, 1997); artificial life (Adami, 1998); swarm intelligence (Kennedy & Eberhart, 2001); and artificial immune systems (de Castro & Timmis, 2002). These techniques are inspired by the architectural and behavioural characteristics identified in biological systems such as distributed knowledge, robustness, fault tolerance, decentralised control, scalability, learning, memory and self-organisation, which exist in solving problems and facilitating life.

There are two general properties of biological systems that are of interest to computer scientists. The first is the robustness of biological systems. Robustness is the property that allows a system to maintain its function against internal and external perturbations (Kitano, 2004). All biological systems must be robust enough against environmental and genetic perturbation to be evolvable and to survive sufficiently for a certain period of time for reproduction (Kitano, 2004). The second important property of biological systems is self-organisation. Camazine et al. (2001) defines a self-organised as a process whereby the pattern of the global level of the system emerges from the result of many interactions among the lower level components of the system. The rules specifying the interactions among the system's lower level components are executed using only local information, without reference to the global pattern (Camazine et al., 2001). Such self-organised patterns are known as emergent phenomena, which cannot be understood by the examination of individual components within the system alone. The immune system is an excellent example of a robust, self-organised system (Cohen, 2000). It has quite remarkable abilities: recognition/discrimination (Cohen, 2000); maintenance (Cohen, 2000); inference from danger/context (Matzinger, 2002); and memory (Janeway et al., 2005). In maintaining a healthy state, immune system computes the input to the system, which is the state

of the body and the output of the immune system is the healing process (Cohen, 2006). In this sense, immune systems work like a computational machine that transforms body-state data into immune-system data while simultaneously providing feedback on the body to modify its state and restore a healthy state (Cohen, 2006).

To date, biologically-inspired computation techniques have had various level of success in different types of problem (Kelsey & Timmis, 2003; Garrett, 2005; Cutello et al., 2005; Ji & Dasgupta, 2007; Bernardino et al., 2010). Despite the success, there has been little work that addresses the methodology or process that need to be adopted while trying to develop a bio-inspired algorithm. The inspiration was often naive and many algorithms are developed by 'reasoning by metaphor' (Stepney et al., 2005). Algorithms that are developed are poorly understood and the developers are unable to produce the behaviour of the biology that acts as the source of inspiration for their work, which is discussed in the majority of artificial immune systems review papers (Hart & Timmis, 2008; Timmis et al., 2008a). Stepney et al. (2005) further argue that biologically-inspired algorithms are best developed within a 'conceptual framework' that includes modelling and analysis of the systems to help in understanding the systems under study before any algorithm is developed. In the conceptual framework a principle biological modelling and abstraction approach needs to be adopted before developing algorithms.

A thesis prepared by (Andrews, 2008) discusses the conceptual framework in detail and performs instantiations of the framework, moving from understanding the biology from the model and simulation to the development of novel algorithm. In reflecting on the framework, Andrews (2008) notes that the the conceptual framework provides good advice and also draws attention to the work needed before the development of an algorithm. This is also explored by Hart & Davoudani (2009), who present a study in which an agent-based modelling technique is used to construct a model of dendritic-cell trafficking in the natural immune system, with the aim of translating this model to an engineered system: a large-scale wireless sensor network. Detailed information regarding the mapping of the behaviour of dendritic cells to the engineered system is available in Davoudani et al. (2008). As described in Hart & Davoudani (2009), there are some generic issues which may arise when modelling biology with the intention of applying the results to AIS, rather than when modelling in order to replicate observed biological data. Hart & Davoudani (2009) further suggest that the constraints of the engineered system must be considered when iterating the model, and certain aspects of the biology may not be appropriate for the system.

1.1.2 Artificial Immune Systems

The CFA concepts offered by Stepney et al. (2005) are investigated in the context of artificial immune systems (AIS). AIS is a diverse area of research that attempts to bridge the divide between immunology and engineering which combine elements of immunology with the engineering sciences (both computational and mathematical approaches). From the bio-inspired computing point of view, de Castro & Timmis (2002) define AIS as: 'adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving'.

AIS has been developed through the application of techniques such as mathematical and computational modelling of immunology (Read et al., 2008; Kelsey et al., 2008; Andrews, 2008), abstraction of the model into algorithms and implementation in the context of engineering (Owens et al., 2007, 2008; Aickelin & Greensmith, 2008). AIS is typically applied to many of the same applications just as other bio-inspired approaches are applied, such as learning, anomaly detection and optimisation. The vast majority of AIS has been inspired by four key immune ideas: negative selection, clonal selection, immune networks and dendritic cells (Hart & Timmis, 2008). The immune system, however, is an immensely rich system comprising far more than these four mechanisms. There are many immune processes that are not well understood, and in addition, there is little agreement amongst many immunologists regarding many of the key immune principles leading to a lack of clarity as to the functioning of many immune processes. This leaves the AIS practitioners to decide which aspects of immunological theory to take inspiration from, as there is a wide range of choices, and AIS practitioners can model any of this theory and implement it in a wide range of applications as mentioned in Hart & Timmis (2008). Hart & Timmis (2008) further highlighted that the true value of the immune metaphor will only be revealed in systems which exploit the full richness of the natural immune system, which is gained through the synergistic interaction between different type of cell in an innate and adaptive immune system. They further argue that any systems which can exploit this interaction have huge potential to benefit from the application of the immune metaphor, and categorically distinguish themselves from other biologically inspired paradigms (Timmis et al., 2010a).

1.2 Swarm Robotics

Swarm robotics is an approach to the coordination and organisation of multi-robot systems of relatively simple robots (Şahin, 2005). When compared to traditional multi-robot systems that employ centralised or hierarchical control and communication systems to coordinate the behaviours of the robots, swarm robotics adopts a decentralised approach,

in which the desired collective behaviours emerge from the local interactions and communications between robots and their environment. Such swarm robotic systems may demonstrate three desired characteristics for multi-robot systems: *robustness, flexibility and scalability.* Bayindir & Şahin (2007) define these characteristics as:

- robustness is the degree to which a system can still function in the presence of partial failures or other abnormal conditions;
- flexibility is the capability to adapt to new, diverse, or changing requirements of the environment;
- scalability can be defined as the ability to expand a self-organised mechanism to support larger or smaller numbers of individuals without impacting performance considerably.

Şahin (2005) argues that a significant benefit of swarm robotics is its robustness to failure. However, recent work has shown that swarm robotic systems are not as robust as first thought (Bjerknes, 2009; Bjerknes & Winfield, 2010). To demonstrate these issues, a simple but effective algorithm for emergent swarm taxis (swarm motion towards a beacon) is proposed by Bjerknes (2009); Bjerknes & Winfield (2010). In order to achieve beacon-taxis, these algorithms allow the swarm to move together towards an infrared beacon using a simple symmetry breaking mechanism without communication between robots. To understand the reliability of the system, the evaluation of the effect of the failing robot(s) on the operation of the overall swarm was investigated (Winfield & Nembrini, 2006). These include the (1) complete failures of individual robots due to a power failure, for instance, (2) failure of a robot's infrared sensor and (3) failures of robot's motors only, leaving all other functions operational, including sensing and signalling. The study revealed that the effect of motor failures will have a potentially serious effect in terms of causing the partially failed robot to 'anchor' the swarm impeding the movement towards the beacon. Winfield & Nembrini (2006) then concluded that: (1) analysis of fault tolerance in swarms critically needs to consider the consequence of partial robot failures, and (2) future safety-critical swarms would need designed-in measures to counter the effect of such partial failures. One of the examples is to envisage (form) a new robot behaviour that identifies neighbours who have partial failure, then 'isolates' those robots from the rest of the swarm: a kind of built-in immune response to failed robots (Bjerknes & Winfield, 2010).

The work proposed in this thesis considers this failure mode in the robots and attempts to address the issue of the emergence of 'anchor points' under the case of partial failure of robots, in which the robot's motor is no longer moving due to lack of power, but it

has enough power for simple signalling. In dealing with this issue, this thesis proposes and implements a novel immune-inspired solution, which enables the swarm to self-heal under certain failure modes, and continue to operate and complete the task. We therefore propose an extension to the existing ω-algorithm (Bjerknes, 2009) that affords a self-healing property that functions under certain failure modes. This approach is in line with the work in Timmis et al. (2010a) that suggested there is a great deal to offer between the area of AIS and swarm robotics in particular. To develop this approach, we have drawn the inspiration from the process of granuloma formation, a process of containment and repair observed in the immune system, from which we derive a set of design principles that we use to instantiate an algorithm capable of isolating the effect of the failure, and to initiate a repair sequence to allow the swarm to continue operating.

1.3 Outline of Thesis

We illustrate here details relating to the content and structure of the thesis. This can be categorised into three subsections. Firstly, we present the research goal and contributions for this thesis. Secondly, a structure and content of the thesis, summarising each stage of our work and how it fits with our thesis goal are presented. We finally list the papers that have been published whilst working headed for this thesis, highlighting where it has been emphasised in the content of the thesis chapter.

1.3.1 Research Goal and Contributions

This thesis develops a novel immune-inspired algorithm with the conceptual framework (Stepney et al., 2005) as a methodology that can be applied to issues in fault tolerance in swarm robotic systems. This is in accordance with our discussion in section 1.2, that mentioned that since a swarm is a completely decentralised system, there is a need to introduce new behaviours of individual robots that allows robots to detect and respond to failures of the other robots or self-healing behaviour. We propose an immune-inspired solution which under certain failure modes, enables the swarm to self-heal and continue operation and complete the task for the issues of motors failures due to the lack of energy in the robots and attempts to address the issue under a certain failure mode. To develop the algorithm, we have taken inspiration from the process of granuloma formation, a process of containment and repair observed in the immune system, from which we derive a collection of design principles that we use to instantiate an algorithm capable of isolating the effect of the failure, initiate a repair sequence to allow the swarm to continue operation.

Therefore, the goal of this thesis is:

to develop a novel immune-inspired algorithm in a principled manner for self-healing swarm robotic systems that is capable of recovery from certain failure modes

Based on the descriptions of our research goal, the following research question will be addressed during the course of the thesis:

to what extent can an effective algorithm for achieving fault-tolerance with respect to beacon taxis in a robotic swarm can be developed using immunological inspiration?

The research goal will be addressed by the following:

1. analysis of the model and simulation of granuloma formation.
2. the abstraction of results obtain from the model and simulation into a collection of design principles to be instantiated to a novel immune-inspired algorithm.
3. demonstration of the novel immune-inspired algorithm for self-healing swarm robotic systems by comparing with other available solutions.

These steps are achieved in 7 chapters and appendices, which are described in section 1.3.2.

1.3.2 Thesis Structure

This thesis is broken down into 7 chapters where we describe the development process of an AIS, presenting a chronological investigation of the problem domain, the description of the immune ideas for AIS inspiration, the development of models and simulations, and finally the construction of the design principles and a specific AIS algorithm. Naturally, this thesis structure follows closely the stages of CFA.

In chapter 2, we first provide a critical review of the related work and literature in the field of swarm robotics with an overview of complexity and self-organisation, that act as a basis of the field of swarm intelligence and swarm robotics. Secondly, we introduce the field of swarm intelligence, followed by the discussion on swarm robotic systems, which is inspired by the properties of self-organisation in complex systems. We also emphasise the discussion on the suitable applications in swarm robotic systems including its challenges and issues. Next, we focus on the discussion of swarm beacon taxis, which is an aggregation task in swarm robotic systems to move towards a beacon that will be used in this thesis. Here, the algorithms for swarm beacon taxis; α, β and ω algorithms are

described. In these algorithms, robots only move towards the beacon due to the symmetry breaking mechanism and they are not pre-programmed to move towards the beacon. Finally, we describe the possible anchoring issues in the ω algorithm that have been analysed Bjerknes (2009).

In chapter 3, we first provide the investigation on how AIS has influenced the development of swarm robotics solutions to date. Thirdly, we detail our first attempt to elaborate and investigate the ideas of development and morphology of granuloma formation. The final section in this chapter describes the immuno-engineering and conceptual framework approaches that will be used as a basis in developing AIS solution in swarm robotic systems. We concludes the section on AIS with a critical discussion on the application of AIS in solving issues in swarm robotic systems. This chapter ends with a critical discussion of the relationship between the properties of swarm robotics and granuloma formation and how both can be related to each other.

Chapter 4 details our attempt to develop models and simulation of granuloma formation. In order to do this, we need to return to immunology to identify the suitable biological details necessary for models and simulation construction, which has been described in chapter 3. We first construct the unified modelling language (UML) model and agent based simulation based on the understanding of the principles of granuloma formation that is described in section 3. We then prepare a set of experiments to understand the process of granuloma formation and we examine the performance of the simulation using various parameter settings.

In chapter 5 we describe our proposed immune-inspired algorithm, the granuloma formation algorithm, which is inspired by the process of granuloma formation in immune systems. Based on the model and agent based simulation of granuloma formation developed in chapter 4, we first identify the patterns of reactions during formation of granuloma that can be instantiated as a set of design principles for the design algorithm for self-healing in swarm robotic systems. This leads us to present an AIS algorithm framework, which incorporates the design principles into algorithm development.

Chapter 6 will focus on explaining the simulation results that we obtained from simulating ω-algorithm and granuloma formation algorithm with Player/Stage; the robots simulator. First, we describe the experimental protocol and the performance matrices for the experiments. Next, we explain the results obtained from the ω-algorithm that will act as the baseline throughout this chapter. Before explaining the results obtained from the granuloma formation algorithm, we include the results obtained from the single and shared charger algorithm that work as the comparison.

To conclude the work of the thesis, chapter 7 reflects on the experience of developing an immune-inspired algorithm for self-healing swarm robotics systems. We begin by

summarising the work presented in chapters 4, 5 and 6 with respect to the AIS method identified in chapter 3, the CFA. We then conclude the thesis by summarising our granuloma formation algorithm that we developed based on the ideas of granuloma formation for self-healing swarm robotic systems, identifying future work, and returning to assess how we have addressed our research questions laid out in the introductory section.

1.3.3 Publications

A number of publications have resulted from preparing this thesis, and these form the basis of much of the work presented within. These publications, along with how they relate to this thesis, are:

1. Ismail, A. R., & Timmis, J. (2009). Aggregation of swarms for fault tolerance in swarm robotics using an immuno-engineering approach. In Proceedings of the 9th Annual Workshop on Computational Intelligence.
   I am the principal author of this conference paper that influenced the ideas presented in chapter 5. In this paper, the initial idea of the granuloma formation algorithm is presented.

2. Ismail, A. R., & Timmis, J. (2010). Towards self-healing swarm robotic systems inspired by granuloma formation. In Special Session: Complex Systems Modelling and Simulation, part of ICECCS 2010, IEEE, (pp. 313 - 314)
   I am the principal author on this conference paper describing on how granuloma formation can be the source of inspiration in the development of the immune-inspired algorithm, most of which is discussed thoroughly in chapter 4.

3. Timmis, J., Tyrrell, A., Mokhtar, M., Ismail, A. R., Owens, N., & Bi, R. (2010c). An artificial immune system for robot organisms. In P. Levi, & S. Kernback (Eds.) Symbiotic Multi-Robot Organisms: Reliability, Adaptability and Evolution, vol. 7, (pp. 279 - 302). Springer.
   I am the co-author on this book chapter that extends the ideas presented in the previous publication Ismail & Timmis (2009), which described the mapping of the properties of granuloma formation and swarm robotics systems

4. Ismail, A. R., Timmis, J., Bjerknes, J. D., & Winfield, A. F. T. (2011). An immune inspired swarm aggregation algorithm for self-healing swarm robotic systems. Swarm Intelligence. (in submission).
   I am the principal author on this journal paper, discussing on the result and analysis on the granuloma formation algorithm most of which is reproduced in chapter 6.

5. Ismail, A. R., & Timmis, J. (2011). Modelling Containment Mechanisms in the Immune System for Applications in Engineering. In Proceedings of the 9th International Conference on Artificial Immune Systems (ICARIS 2011), vol. 5132 of LNCS, (pp. 340 - 351). Springer.
   I am the principal author and part of chapter 5 is presented in this conference paper. In this paper, the discussion on the simplified model and simulation of granuloma formation are presented.

CHAPTER 2

Swarm Robotics

The aim of this chapter is to provide a critical review of the related work and literature in the field of swarm robotics. The review is divided into five main sections. Firstly, we give an overview of swarm intelligence in section 2.1 with a discussion on the meaning, properties and algorithms in swarm intelligence. This is followed by a discussion on swarm robotics in section 2.2 that is inspired by the properties of self-organisation in complex systems. Here we place emphasis on the discussion on the suitable applications in swarm robotic systems such as foraging, surveillance and aggregation. This section also describes the challenges and issues in swarm robotics, specifically in maintaining the robustness of the swarm robotic systems. Having described the applications in swarm robotics, we draw our attention to the discussion on swarm taxis algorithms in section 2.3. Here an aggregation task in swarm robotic systems called the swarm beacon taxis is described where the swarm collectively moves towards a beacon due to the symmetry breaking mechanism in the algorithm, which will be used as our experimental case study in this thesis. The algorithms for swarm beacon taxis discussed in this section are; α, β and ω algorithms. We then describe other swarm intelligence approaches in swarm robotics in section 2.4. In this section some of the swarm robotics projects that use the swarm intelligence approaches are described. It includes the Swarm-bots project, the Pheromone Robotic project, the I-swarm project and the recent Symbrion and Replicator project. This chapter ends with section 2.5, which is conclusion and discussion on the needs of the researchers in swarm robots to study certain issues such as power and energy autonomy, dependability and other related issues, to allow swarm robotic systems to be more robust. This section also describe on the anchoring issues in the ω algorithm that

has been analysed Bjerknes (2009) which is addressed and solved through out this thesis.

2.1 Swarm Intelligence

Robustness, self organisation and adaptation are essential properties that have been a source of inspiration for research in computer science. In biological systems, robustness is a fundamental characteristic. As has been reported in Kitano (2007), numerous articles have been published on how robustness is involved in various biological processes and on mechanisms that give rise to robustness in living systems (Bhalla & Iyengar, 2001; Kitano, 2004). In biological systems robustness is defined by Kitano (2004) as:

> *'a property that allows a system to maintain its functions despite external and internal perturbations. It is one of the fundamental and ubiquitously observed systems-level phenomena that cannot be understood by looking at the individual components. A system must be robust to function in unpredictable environments using unreliable components'*

Besides robustness, other biological properties that have inspired researches in computer science are self-organisation and adaptation. Self-organisation, or decentralised control, is widespread in biological systems, including cells, organisms and groups that posses a large number of subunits, and these subunits lack either the communication abilities or the computational abilities, or both, that are needed to implement centralised control (Camazine et al., 2001).

Adaptation is a basic phenomena of biology (Williams, 1966), whereby an organism becomes better suited to its habitat. The term may also refer to the adaptation of the organism, which is especially important for an organism's survival: for example, the adaptation of horses' teeth to the grinding of grass, or their ability to run faster and escape from predators. Such adaptations are produced in a variable population by the better suited forms reproducing more successfully, that is, by natural selection (Williams, 1966). According to Williams (1966), adaptive traits may be structural, behavioural or psychological. Williams (1966), mentions that structural adaptations are physical features of an organism such as shape, body covering and defensive or offensive armament. Behavioural adaptations are composed of inherited behaviour chains and/or the ability to learn: behaviours may be inherited in detail (instincts), or a tendency for learning may be inherited for example by searching for food, mating and vocalisation. Finally, physiological adaptations may permit the organism to perform for instance by making venom or secreting slime (Williams, 1966).

This section therefore explores the field of swarm intelligence that is motivated by the properties of self-organisation and adaptation in earlier discussion. The section also describes the features of swarm intelligence systems with examples of swarm intelligence algorithms.

The word swarm is defined by Hinchey et al. (2007) as:

> *'images of large groups of small insects in which each member performs a simple role, but the action produces complex behaviour as a whole'*

It consists of many simple entities that have local interactions, including interacting with the environment, leading to the emergence of complex, or macroscopic behaviours and the ability to achieve significant results as a team resulting from the combination of simple, or microscopic, behaviours of each entities (Hinchey et al., 2007). Similar complex social structures also occur in higher-order animals and insects such as colonies of ants, flocks of birds, or packs of wolves. Although there is normally no centralised control structure dictating how individual agents should behave, local interactions between such agents often lead to the emergence of global behaviour. Examples of systems like this can be found in nature, including ant colonies, bird flocking, animal herding and fish schooling. Social insects coordinate their actions to accomplish tasks that are beyond the capabilities of a single individual. For example, termites will be able to build large mounds, and based on the foraging raids ants can collectively carry large prey. Different groups behave like swarms in different ways. Wolves, for example, accept the alpha male and female as leaders that communicate with the pack via body language and facial expressions. The alpha male marks his pack's territory and excludes wolves that are not members (Hinchey et al., 2007). With no centralised co-ordination mechanisms behind the synchronised operation of biological systems, the system level operates in a robust, flexible and scalable manner (Camazine et al., 2001).

Based on the definition of swarm, we extend the meaning to swarm intelligence, which is denoted by Dorigo & Birattari (2007) as:

> *'the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralised control and self-organisation. In particular, the discipline focuses on the collective behaviours that result from the local interactions of the individuals with each other and with their environment'*

As described in Timmis et al. (2010a), this definition encapsulates the key properties of a swarm system which occur in both natural and artificial systems. Timmis et al.

(2010a) further elaborate the fact that Dorigo & Birattari (2007) consider that in the natural world, swarm intelligence studies a wide variety of systems ranging from ant colonies to flocks of birds and from an engineering perspective, Dorigo & Birattari (2007) consider swarm intelligence covering systems from multi-robot systems to optimisation.

Timmis et al. (2010a) further argue that according to Dorigo & Birattari (2007), swarm intelligence is a broad area that can be discussed under two orthogonal classifications: natural vs. artificial (the study of biological systems or human engineered artefact); and science vs. engineering. This is elaborated in table 2.1, which shows the comparison of the science vs. engineering classification for swarm intelligence and AIS, which exposes a natural relationship between the goals of the two fields, as has been proposed by Timmis et al. (2010a), who explain that despite their similarities, the two fields can complement each other.

Table 2.1: Classification of the role of Swarm Intelligence and Artificial Immune Systems in Science and Engineering (Timmis et al., 2010a)

	Swarm Intelligence	Artificial Immune Systems
Science	Understand how local individual behaviours result in coordinated population behaviours	Use models to explain phenomena and guide experimental work
Engineering	Exploit the understanding of natural swarms in designing problem solving systems	Apply systems inspired by immune functions, principles and models to problem solving

Several areas of engineering have adopted the idea that swarms can solve complex problems and some of them are described in Bonabeau et al. (1999). Some of the examples highlighted by Bonabeau et al. (1999) are combinatorial optimisation, routing communications network, as well as solving robotics applications (Beni, 2005). According to Bonabeau et al. (1999), the two best known swarm intelligence algorithms are: Particle Swarm Optimisation (PSO) and Ant Colony Optimisation (ACO). The ideas of PSO emerged from the swarming behaviours observed in flocks of birds, swarms of bees and school of fish (Bonabeau et al., 1999). The individuals in PSO communicate either directly or indirectly with one another. As an algorithm, PSO can be applied to solve various function optimisation problems, as the main strength of the algorithm is its fast convergence (Abraham et al., 2006). However, in successfully applying PSO, it is suggested that one of the key issues is to find a way of mapping the problems in PSO particle, which directly affects its feasibility and performance (Abraham et al., 2006). The Ant Colony Optimisation (ACO) represents the model of the collective foraging behaviour of ants which shows the path selected by ants to a food source Deneubourg et al. (1990). The main idea in this algorithm is the indirect communication between the ants which is

established by the means of pheromones in finding the shortest path between their nest and food (Abraham et al., 2006). This is also in accordance with the terms 'stigmergy' to describe the particular type of communication that is stimulated by the ants in the environment, which is observed in colonies of ants. Once the presence of the pheromone is perceived by the other ants in the environment, they tend to follow the paths where the pheromone concentration is higher. Through this mechanism, ants are able to transport food to their nest in a remarkably effective way (Dorigo et al., 2006). According to Dorigo et al. (2006) the main characteristics of stigmergy that differentiate it from other forms of communications are:

- it is indirect; non-symbolic form of communication mediated by the environment.
- stigmergic information is local: it can only be accessed by those insects that visit the locus in which it has been released or by its immediate neighbour.

Different types of ACO algorithms have been applied and proposed to solve different types of problem. The original ACO algorithm is known as 'Ant System' and was proposed in early nineties by Dorigo et al. (1991); Dorigo (1991). Since then, many researchers have introduced a number of other ACO algorithms which have been applied to problems such as assignments (Socha et al., 2002, 2003), scheduling (Merkle et al., 2000; Yagmahan & Yenisey, 2008) and other applications. All these algorithms still share the same characteristics idea of path finding using the high concentration value of pheromones. Even though ACO has been applied successfully in many different types of engineering problem, Dorigo et al. (2006) further argues that increasing attention is needed to apply ACO to more challenging problems that involves dynamic modification of data or the stochastic nature of the objectives constraints as well as extending the capability of ACO from discrete to continuous problems in optimisation.

2.2 Swarm Robotics and Its Applications

Inspired from the observation of swarms of social insects such as ants, termites, wasps and bees, which are the fascinating examples of interactions among individual, a novel approach to the coordination of a large number of robots termed as *swarm robotics* has been introduced by artificial intelligence researchers. Hinchey et al. (2007) describes swarm robotics as:

> *'an application of swarm intelligence techniques to the analysis of activities in which the agents are physical robotic devices that can effect changes in their environments based on intelligent decision-making from various input'*

It is also explained by Şahin (2005) that swarm robotics is the co-ordination and organisation of multi-robot systems or relatively simple robots. When compared to traditional multi-robot systems that employ centralised or hierarchical control and communication systems in order to coordinate behaviours of the robots, swarm robotics adopts a decentralised approach, in which the desired collective behaviours emerge from local interactions and communications between robots and their environment. Such swarm robotic systems demonstrate three desired characteristics, which are defined by Bayindir & Şahin (2007) as:

- robustness is the degree to which a system can still function in the presence of partial failures or other abnormal conditions
- flexibility is the capability to adapt to new, diverse, or changing requirements of the environment
- scalability can be defined as the ability to expand a self-organised mechanism to support larger or smaller numbers of individuals without impacting performance considerably

According to Şahin (2005), swarms are appealing to robotic systems because:

- they have simple components as compared to centralised systems designed for the same task. Thus, the robotic units could be, in principle, modularised, mass produced, and could be interchangeable and maybe disposable.
- of the reliability of swarm that allow them to be designed to survive through many kinds of disturbance.
- of redundancy, the swarm would have the ability to adapt dynamically to the working environment; another feature required for high reliability. It was also possible to envision the swarm as acting like a massive parallel computational system, and thus carry out tasks beyond those possible in other types of robotic systems, either complex single robots or centralised groups of robots.

Other than the above-mentioned advantages of swarms that are applicable to robotics systems, swarms also have properties such as self organising and coordination, which are still beyond the reach of current multi-robot systems (Şahin, 2005). A recent paper by Şahin & Winfield (2008) mentioned that the key benefit of the swarm robotics approach is robustness, which manifests itself in a number of ways. Firstly, because a swarm of robots consists of a number of relatively simple and typically homogeneous robots, which are not pre-assigned to an explicit role or tasks within the swarm, then the swarm can

self-organise or dynamically restructure the way individual robot is arranged. Secondly, the swarm approach is highly tolerant to the failure of an individual robot. The failure of an individual robot does not affect the goal of the systems of the whole. Thirdly, decentralised control in swarms means that there is no common-mode failure point or vulnerability in the swarm. Indeed, it could be said that the high level of robustness evident in robotic swarms comes for free in the sense that it is intrinsic to the swarm robotics approach, which contrasts with the high engineering cost of fault tolerance in conventional robotic system (Şahin & Winfield, 2008).

Some criteria in distinguishing swarm robotics research from other robotics research have been put forwarded and explicitly stated by Şahin (2005). However the author emphasises that the definition and the list of criteria is based on their understanding and these criteria are not meant to be used as a checklist for determining whether a particular study is a swarm robotics study or not. The criteria, which are taken directly from Şahin (2005) are:

1. autonomous robots: individuals should have a physical embodiment in the world, be situated, can physically interact with the world and be autonomous.

2. large number of robots: the study should be relevant for the coordination of a swarm of robots. Therefore, studies that are applicable to the control of only a small number of robots, and do not aim for scalability, fall outside swarm robotics.

3. few homogeneous groups of robots: the robotic system being studied should consist of relatively few homogeneous groups of robots, and the number of robots in each group should be large. That is, studies that are concerned with highly heterogeneous robot groups, no matter how large the group is, are considered to be less swarm robotic.

4. relatively incapable or inefficient robots: the robots being used in the study should be relatively incapable or inefficient on their own with respect to the task at hand. That is, either 1) the robots should have difficulties in carrying out the task on their own, and the cooperation of a group of robots should be essential, or 2) the deployment of a group of robots should improve the performance on the handling of the task.

5. robots with local sensing and communication capabilities: the robots being used in the study should only have local and limited sensing and communication abilities. This constraint ensures that the coordination between the robots is distributed. In fact, the use of global communication channels within the robot group is likely to

result in unscalable coordination mechanisms and would therefore act against the first criterion mentioned above.

There are a number of different application areas for swarms of robots, which can be useful in circumstances where one robot is not capable of completing the task or multiple simultaneous tasks are required to achieve the task. Some of the ideas in the domain of application that is applicable to swarm of robots have been put forward by Şahin (2005). Below, we present a number of task domains as highlighted by Şahin (2005), which also emphasise the properties of the tasks that make them suitable for swarm robotic systems together with real-world problems as examples.

1. Tasks that cover a region: Swarm robotic systems are distributed systems and would be well-suited for tasks that are concerned with the state of a space. The distributed sensing ability of swarm robotic systems can provide surveillance for the immediate detection of hazardous events, such as the accidental leakage of a chemical. In dealing with this, a swarm robotic system would have two major advantages over sensor networks. They are:

    - a swarm robotic system has the ability to focus on the location of a problem by mobilising its members towards the source of the problem that would allow the swarm to better localise and identify the nature of the problem;
    - a swarm robotic system also can self-assemble forming a patch that would block the leakage.

2. Tasks that are too dangerous: Individuals that create a swarm robotic system are dispensable in making the system suitable for domains with dangerous tasks. For example clearing a corridor on a mining field.

3. Tasks that scale up or scale down in time: A swarm robotic system has the power to scale up or scale down with according to the task. For example, the scale of an oil leakage from a sunk ship can increase dramatically as the tanks of the ship breaks down. Thus, a swarm robotic system can be scaled up by the pouring more robots into the area in the sunk ship.

4. Tasks that require redundancy: The robustness of swarm robotic systems come from implicit redundancy in the swarm, allowing the system to degrade peacefully making the system less prone to catastrophic failures. For instance, swarm robotic systems can create dynamic communication networks in the battlefield. Such networks can enjoy the robustness achieved through the re-configuration of the communication nodes when some of the nodes are hit by enemy fire.

Further to the above-mentioned tasks, swarms of robots are also useful when multiple objects are required for moving an object from different location, building an object from smaller objects. This is because when many robots perform these tasks, they can be completed quicker. Other tasks may include any type of task that needs to be completed faster by multiple robots such as foraging, surveillance, exploration, mapping and aggregation.

According to Winfield (2009b), 'foraging is a complex task involving the coordination of several tasks including efficient exploration (searching) for objects, food or prey; physical collection or harvesting of objects; homing or navigation whilst transporting those objects to collection point(s), and deposition of the objects before returning to foraging' (Winfield, 2009b). As highlighted in Winfield (2009b), there are few types of foraging robots employed in real-world applications. Some of them suggested applications includes search and rescue, mowing a lawn, tidying up a room (Winfield, 2009a), harvesting a resource (Sheng et al., 2006; Hsieh et al., 2008) or cleaning up hazardous waste (Sheng et al., 2006). Surveillance systems are often needed in areas that are dangerous for a human presence and intervention. They can take in many forms such as tracking of a target or environmental monitoring (Winfield, 2009b). In more traditional methods of sensing, many probes will be placed at fixed locations around an area. However, with a swarm of robots they can organise themselves to cover the area concerned and will dynamically adapt if the environment changes. Furthermore, it is possible to manipulate the interactions of multiple small, low-cost robots, with a limited range of local communication ability, to collaboratively search and engage in tasks in an unknown large-scale hostile area which is dangerous for human presence.

Aggregation is one of the fundamental behaviours of swarms in nature that has been observed in different types of organism, ranging from bacteria to social insects and mammals (Camazine et al., 2001). With aggregation, organisms can avoid and escape from their predators, resist hostile environmental conditions and find mates (Soysal et al., 2007). One of the earliest robotics applications in aggregation is demonstrated by Melhuish et al. (1999), where the robots are required to form a predetermined size of cluster around an infrared beacon. In this method, robots need to simultaneously produce sound similar to the sound produced by birds called 'chorus'. However, the results obtained were only useful when the environment was noiseless. Another study related to the aggregation of robots has been undertaken by Nembrini et al. (2002) and Bjerknes (2009). The algorithms that have developed make use of local wireless connectivity information alone to achieve swarm aggregation namely the α (Nembrini et al., 2002), β (Nembrini et al., 2002) and ω Bjerknes (2009) algorithms. These algorithms are inspired by the framework of minimalist design introduced by Melhuish & Hoddell (1998), which focused on very limited robots that are able to communicate locally but lack global knowledge of the

environment. These algorithms are discussed further in section 2.3.

Although the principles of robot foraging, robot surveillance, aggregation and other swarm robotics applications are well understood, Winfield (2009b) argues that most of the applications are only found in research laboratories and need further research if they are to be applied to the real world. This is further explored in section 2.2.1.

2.2.1 Challenges and Issues

It is evident that, whilst attracting much interest from the research community and being well funded, swarm robotics is still at an early stage of development. Many critical issues remain unresolved, including gaining a full understanding of swarming behaviour and translating this into workable technology, which is perhaps the greatest challenge; the development of the associated robots, which must be relatively inexpensive and task-specific; development of suitably low-cost and miniaturised sensors and actuators; reducing power consumption, with the possibility of utilising energy harvesting from the robot's environment; and integrating all of these technologies and concepts into robust and reliable systems.

Şahin (2005) argues that a significant benefit of swarm robotics is robustness to failure. However, recent work has shown that swarm robotic systems are not as robust as first thought (Bjerknes, 2009; Bjerknes & Winfield, 2010). To demonstrate the issues, a simple but effective algorithm for emergent swarm taxis (swarm motion towards a beacon) is proposed by Bjerknes (2009); Bjerknes & Winfield (2010), which is further explained in section 2.3. The study showed that the effect of motor failures will have a potentially serious effect of causing the partially-failed robot(s) to 'anchor' the swarm, impeding the movement towards the beacon.

As argued in Winfield & Nembrini (2006) high levels of robustness in swarm robotics are frequently not supported by empirical or theoretical analysis; the paper also raised various questions such as what is meant by robustness and how one can measure robustness or fault tolerance of a swarm robotic system. In an attempt to answer these questions, Winfield & Nembrini (2006) explored fault tolerance in robot swarms through failure mode and effect analysis (FMEA) [1] illustrating these by a case study of a wireless connected robot swarm, in both simulation and real laboratory experiments. The FMEA case study showed that a robot swarm is remarkably tolerant to the complete failure of robot(s) but is less tolerant to partially failed robots. For example, a robot with failed motors but with all other sub-systems functioning can have the effect of anchoring the swarm and

[1] A failure mode and effects analysis (FMEA) is a procedure for analysis of potential failure modes within a system for classification by severity or determination of the effect of failures on the system. Failure modes are any errors or defects in a process, design, or item, leading to the studying the consequences of those failures to the systems.

hindering or preventing swarm motion. The authors then concluded that: (1) analysis of fault tolerance in swarms critically needs to consider the consequence of partial robot failures, and (2) future safety-critical swarms would need designed-in measures to counter the effect of such partial failures (Winfield & Nembrini, 2006). In Bjerknes & Winfield (2010), the authors envisaged a new robot behaviour that identifies neighbours who have partial failure, then 'isolates' those robots from the rest of the swarm: a kind of built-in immune response to failed robots.

In swarm robotic systems, various types of failure modes and the effect of individual robot failures on the swarm have been analysed. As stated in Bjerknes & Winfield (2010) the failure modes and effects for swarm beacon taxis are as follows:

- Case 1: complete failures of individual robots (completely failed robots due, for instance, to a power failure) might have the effect of slowing down the swarm taxis towards the beacon. These are relatively benign, in the sense that 'dead' robots simply become obstacles in the environment to be avoided by other robots of the swarm. Obviously, given the fact that they are obstacles, there will be a reduction in the number of robots available for team work. However, as the number of failing robot increases, there is a possibility that the failing robot will 'anchor' the swarm, impeding its taxis toward the beacon.

- Case 2: failure of a robot's IR sensors. This could conceivably result in the robot leaving the swarm and becoming lost. The robot that leave the swarm would become a moving obstacle to the rest of the swarm. When some of the robots have lost and becoming the moving obstacle this might reduce the number of robots required for the team work as some of the robots now have lost and move away from the swarm.

- Case 3: failure of a robot's motors only. Motor failure only leaving all other functions operational, including IR sensing and signalling will have the potentially serious effect of causing the partially-failed robot to 'anchor' the swarm, impeding its taxis toward the beacon.

2.3 Swarm Taxis Algorithms

As described in section 2.2, swarm robotics is an approach to the coordination of multi-robot systems consisting of large numbers of simple physical robots, which emerged from the field of biological studies of insects, ants and other fields in nature, where swarm behaviour occurs. In swarm robotics, the desired collective behaviour emerges from the

interactions between the robots and the environment. Some of the application areas for swarm of robots discussed in section 2.2 are foraging, surveillance and aggregation.

Aggregation of a swarm requires that robots in the systems to have a physical coherence when performing a task. Robots are randomly placed in an environment and are required to interact with each other. This is relatively easy when a centralised control approach is used but very challenging with a distributed approach (Bayindir & Şahin, 2007). Nembrini et al. (2002) and Bjerknes (2009) developed a class of aggregation algorithm, which makes use of local wireless connectivity information alone to achieve swarm aggregation, namely the α algorithm (Nembrini et al., 2002), β algorithm (Nembrini et al., 2002) and ω algorithm (Bjerknes, 2009). These algorithms, which also called as the 'swarm beacon taxis algorithms' are inspired by the framework of minimalist design introduced by Melhuish & Hoddell (1998), which focused on very limited robots that are able to communicate locally but lack of global knowledge of the environment. The only sensor information available, besides the basic obstacle avoiding infrared (IR) sensors, are the beacon sensors and the radio communication. It is assumed that the communication hardware has a limited range, is omni-directional and the quality of the transmission is not optimal. The aim is to keep the robot as simple as possible, as it is believed that stability is reachable only with a limited range radio device and proximity sensors for avoidance. As mentioned in Winfield et al. (2008), the advantages of this approach are:

- the robots need neither absolute or relative positional information.
- the swarm is able to maintain aggregation (stay together) even in unbounded space.
- the connectivity needed for and generated by the algorithm means that the swarm naturally forms an ad-hoc communication network, that would be advantageous in many swarm robotics application such as distributed sensing, exploring or mapping, because it allows data to be communicated between any two robots and facilitates data collection from the whole swarm via single connection with just one robot.

In swarm beacon taxis algorithms, in order for the robots to achieve a beacon, Nembrini et al. (2002) and Bjerknes (2009) allow the swarm to move together (taxis) towards an IR beacon. Only the robots who have direct line to the IR beacon are attracted to the beacon and illuminate a beacon sensor. An emergent property of this setup is swarm taxis towards the beacon. This is shown in figure 2.1, where a group of robots have to stay together and at the same time move toward a beacon, which typically be a light source. However, the robots do not individually have the necessary sensing capability to determine the heading toward the beacon. The robots must cooperate to achieve movement in the right direction. In achieving this task, there are three different mechanisms that must be in place (Nembrini et al., 2002) :

- there must be something which prevents the swarm from disintegrating, if a robots moves too far it must turn around and move back to others.
- the robot must maintain a minimum distance from each other to avoid colliding.
- once both conditions are satisfied, a symmetry breaking mechanism must be introduces to ensure that the swarm moves in the right direction.

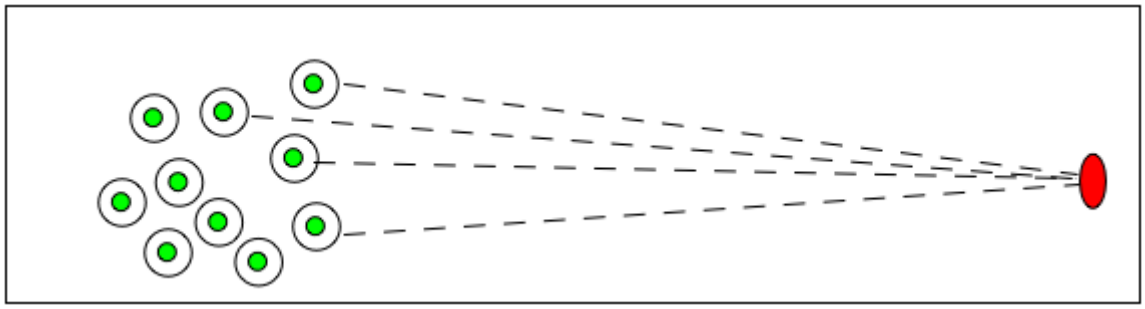


Figure 2.1: The setup of swarm beacon taxis. A swarm of robots (left) with limited sensors must move to a beacon (on the right).

In swarm taxis algorithms the swarm maintains aggregation by the following (Bjerknes et al., 2007):

- coherence behaviour:

  The coherence behaviour works as follows. Each robot has range-limited wireless communication and, while moving, periodically broadcasts an 'I am here' message. The message will of course be received only by those robots that are within wireless range. Robots do not communicate any information on their internal state etc, nor is it possible for a robot to determine its heading relative to the communicating robot. If a robot loses a connection and the number of remaining neighbours is less than or equal to the threshold , then it assumes it is moving out of the swarm and will execute a 180° turn. When the number of connections rises (i.e. when the swarm is regained) the robot chooses a new direction at random. We say that the swarm is coherent if any break in its overall connectivity lasts less than a given time constant.

- avoidance behaviour:

  Each robot has short-range avoidance sensors and a long-range beacon sensor for their avoidance behaviours. The short range collision avoidance sensor is used this to avoid colliding into other robots or obstacles in the environment. The short-range collision avoidance sensor provides robots with information about the relative

direction towards the obstacle and the long-range beacon sensor can detect if the robot is illuminated by the beacon source.

- symmetry breaking behaviour:

  Symmetry breaking in swarm beacon taxis algorithms means that the information of the direction towards the beacon must somehow be captured by the robots in the swarm. An example of symmetry breaking mechanism is shown in figure 2.2. From figure 2.2, in a swarm of robots in the presence of a beacon, some of the robots will be directly exposed to the beacon, and some of the robots could be occluded dependent on the position in the swarm. From figure 2.2, we can see that robot *D*, which is illuminated, tries to avoid robot *C*, since *C* is within *D's* avoidance radius (a range sets in the algorithms). But there will be no similar behaviour for *C* since *C's* avoidance radius is smaller, and hence *C* does not detect *D*. The difference in the avoid radius definitely gives rise to the symmetry breaking behaviour in swarm beacon taxis algorithms.

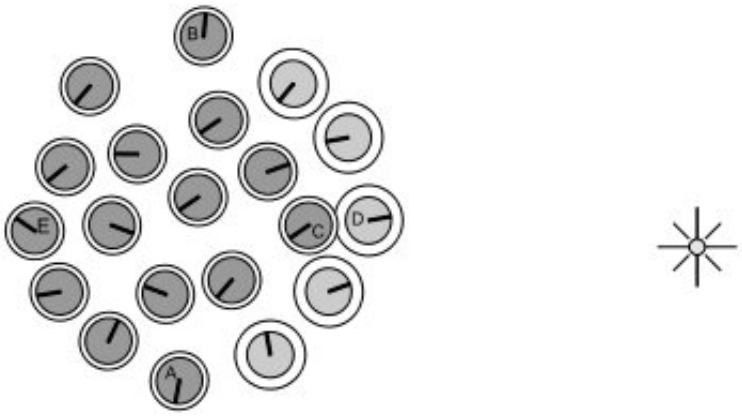


Figure 2.2: The illuminated (light-coloured circle) and the occluded (dark-coloured) robots in swarm beacon taxis (Nembrini et al., 2002)

The following sections discuss the algorithms proposed by Nembrini et al. (2002): the α and β algorithms in detail. Based on both algorithms, the modified version of the α algorithm, called the ω algorithm (Bjerknes, 2009) is described.

2.3.1 α-algorithm

In the α-algorithm, the lowest level swarm behaviour is coherence, which works as described in section 2.3. Nembrini et al. (2002) argue that the swarm is coherent if any break in its overall connectivity lasts less than a given time constant. Coherence gives rise to

the two basic emergent behaviours of swarm aggregation and a connected ad-hoc wireless network. Robots in α-algorithm have five different states. This is depicted in figure 2.3, which is obtained from Bjerknes et al. (2007). From this figure, there are eight transition rules that determine transitions between the states in α-algorithm. The default state in this algorithm is the forward state. Dependent on the robots environment other states can be invoked, but as soon as the corresponding behaviour is performed the robot returns to the forward state. Meanwhile, in the forward state the robots continuously monitor the number of robots or neighbours within communication range, the avoidance range and the beacon sensors. If the number of neighbours falls below a predefined value, the robot will enter the coherence state. In this state the robot will perform a $180\,^{\circ}$ turn. Assuming that a robot lost a connection because of moving away from the swarm, a $180\,^{\circ}$ turn will ensure that the robot will reconnect with the swarm again, contributing to maintaining swarm aggregation. As soon as the $180\,^{\circ}$ turn has been performed, the robot re-enters the default forward state. The random state is entered when the robot notices an increase in number of robots within communication range. Since this number is increasing the robot may be moving closer to the centre of the swarm. In this state the robot will make a random turn to a new direction and then return to the forward state. There are two avoid states, one which applies when the robot is illuminated from the beacon, the other when the robot is occluded from the beacon. In α-algorithm, when there is an object is being detected the robot turns in the opposite direction to the object, and then return to the forward state. The difference between the two avoid states is only their range. The avoid radius when a robot is illuminated is greater than the avoid radius when a robot is occluded.

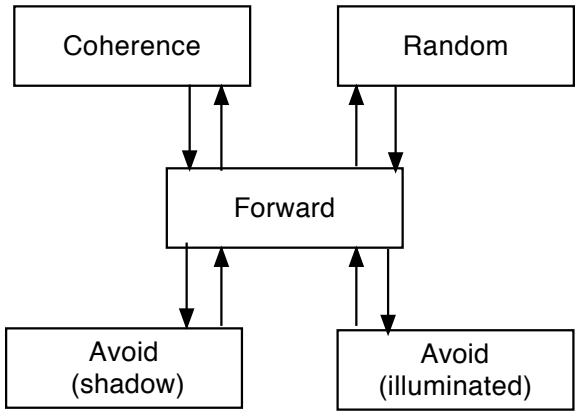


Figure 2.3: The state diagram of α-algorithm (Bjerknes et al., 2007)

The pseudo-code for the α-algorithm is described in algorithm 1. This algorithm restricts itself to use only the information on connections between robots, whether a particular robot is receiving a signal from another or not. The omni-directionality of the

radio implies that there is no positional indication about where to go in case of disconnection. It assumes that robots are able to move forward and turn-on-spot with reasonable precision, that they have infra-red avoidance sensors, are equipped with limited-range radio devices and they carry an omni-directional light sensor which will be able to detect whether a robot is illuminated or not. The algorithm restrict itself to use only the information on connections between robots. As soon as the robot detects a disconnection, the robot assumes it is going in the wrong direction and turns back.

Algorithm 1: Pseudo-code for α-algorithm Nembrini et al. (2002).

```
1  begin
2      Create list of neighbours for robot, Nlist
3      k = number of neighbours in Nlist
4      i = 0
5      repeat
6          if i = 0 then
7              send ID message
8              k = number of neighbours in Nlist
9              if (k<lasK) and k<alpha) then
10                 turn robot through 180 degrees
11             else
12             end
13             make random turn
14         end
15     until forever
16     Steer the robot according to state
17     Listen for calls from robots in range
18     Grow Nlist with neighbours IDs
19     i++
20 end
```

As mentioned by Nembrini et al. (2002), applying the α algorithm to a greater number of robots by making a robot react to every loss of connection leads to an over-reactive swarm which clumps together. To react to every connection is equivalent to aiming towards a complete graph where each vertex is connected to each other which is not the aim of the author. Trying to make the robots less reactive has demonstrated an extreme situation that must be avoided in order to assure the coherence of the swarm. When a robot(s) is linked to the rest of the swarm by a single communication link, a danger lies in the possibility of a robot not reacting to the loss of such connection essential to global connectivity. A limitation of the algorithm is its inability to prevent the swarm splitting into smaller swarms. For example, when two subnets joined by only one connection form, the α algorithm cannot prevent the possibility of the swarm splitting into two. This limitation

has been completely overcome by the more sophisticated 'shared neighbour algorithm' or the β-algorithm (Nembrini et al., 2002) that is described in the following section.

2.3.2 β-algorithm

To avoid the limitation of α-algorithm described in section 2.3.1, Nembrini et al. (2002) introduced the graph theory concept of clustering in β-algorithm. Instead of considering its own degree of connection to trigger a reaction, the robot will receive from its neighbours their adjacency table which has their neighbours' list in order to check whether a particular neighbour is shared by other ones: that is, whether a particular neighbour is the neighbour of other robots' neighbours.

As with the α-algorithm explained in section 2.3.1, the β-algorithm uses radio connectivity to maintain group aggregation. It also has five different states as depicted in figure 2.3, with eight transition rules that determine transitions between the states, and the default state is the forward state But whereas the α-algorithm uses number of robots within communication range as its determining factor, the β-algorithm uses number of shared connections. As the robots constantly move around, the robots broadcast a message with their unique ID and, very importantly, a list of the IDs of all the other robots within their communication range. Based on this list, whenever a robot loses a connection with a robot, it can check with the list received from all other robots to see if they still have a connection with the lost one. In other words, with the increased amount of information broadcast by the robots, they can all calculate a list of their shared connections. If the connection to a robot is lost and the number of shared connections is smaller than a predefined value, β, the robot will turn around.

The pseudo-code for β-algorithm (Nembrini et al., 2002) is described in algorithm 2. The algorithm works as follows:

- for each lost connection, a robot checks how many of its remaining neighbours still have the lost robot in the neighbourhood;
- if the number of lost robot in the neighbourhood is less of equal than the number of the fixed threshold β, the robot turns around and comes back;
- if the degree of connections is increasing, the robot chooses a random headings.

For instance, as shown in figure 2.4, robot A when losing its connection with robot B, will check its connections with other neighbour and finds that robot C and robot D share B as the neighbour. Hence, robot A will react and turn back (only if the number of lost robot in the neighbour is less or equal to a fixed threshold).

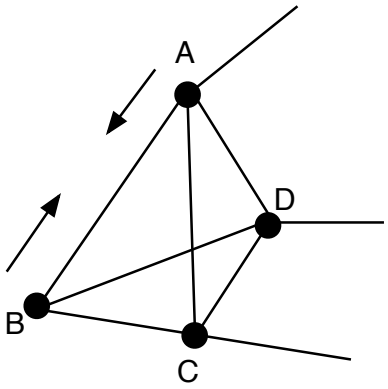


Figure 2.4: The shared neighbour in β-algorithm (Nembrini et al., 2002)

Algorithm 2: Pseudo-code for β-algorithm Nembrini et al. (2002).

```
   input  : W = data from sensors
   output: X = actuation of robot
 1 begin
 2     Create list of neighbours for robot, Nlist
 3     k = number of neighbours in Nlist
 4     repeat
 5         Save copy of Nlist in Oldlist
 6         Save copy of k in LastK
 7         Set reaction of indicator Back to FALSE
 8         Send radio 'ping' to neighbourhood every 100 time steps
 9         Listen for return calls from robots in range that received the 'ping'
10         Create Nlist from all returns
11         k = number of neighbours in Nlist
12         Create LocalList, list of robots which have lost contact since previous
           'ping'
13         for each robot in LostList do
14             Find nShared, number of shared neighbours
15             if nShared <= beta (threshold value) then
16                 Set reaction indicator Back to TRUE
17             end
18         end
19         if Back=TRUE then
20         Turn robot through 180 degrees
21         else if k>LastK then
22             make random turn
23
24     until forever
25 end
```

Simulations performed by Nembrini et al. (2002) has confirmed that β-algorithm increases swarm coherence. The communication bandwidth and the processing power needed for the robots in the β-algorithm are also greater than the α-algorithm. However as explained in Nembrini et al. (2002), the increase in the bandwidth does not really effect the scalability of the algorithm as it is only concerned with exchanging the information between the neighbouring robots.

2.3.3 ω-algorithm

As compared with α and β algorithms, in the ω-algorithm (Bjerknes, 2009) the wireless communication channel is removed and replaced with simple sensors and a timing mechanism. The ω-algorithm has two swarm behaviours: flocking and swarm taxis towards a beacon. The combination means that the swarm maintains itself as a single coherent group while moving towards an IR beacon. Flocking is achieved through a combination of attraction and repulsion mechanisms. Repulsion between robots is achieved using the robots' IR sensors and a simple obstacle avoidance behaviour. Attraction is achieved using a simple timing mechanism. Each robot measures the duration since its last avoidance behaviour, and if that time exceeds a threshold, then the robot turns towards its own estimate of where the centre of the swarm is and moves in that direction for a specified amount of time. This is shown in figure 2.5.

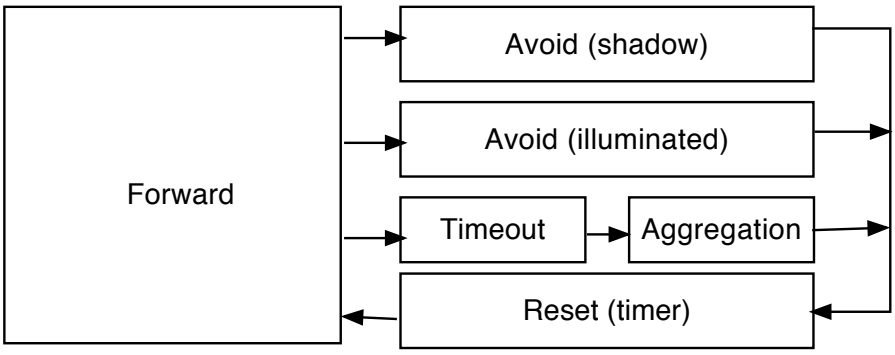


Figure 2.5: The state diagram for the ω-algorithm (Bjerknes & Winfield, 2010)

From figure 2.5, as a robot is moving, it constantly increase a timer, called the aggregation-timer. Each time a robot has to make an avoid movement it resets the aggregation-timer to zero. If the aggregation-timer reaches a certain predefined threshold value, it is most likely that the robot is moving away from the swarm and needs to turn back. However, it does not turn $180\,^\circ$ like the α-algorithm and β-algorithm, but rather toward its perceived center of the swarm. Thus, the robots using the ω-algorithm must have sensors that enable them to estimate their heading to the perceived center of the swarm with respect to the

other robots in the swarm. This is achieved by using the robots proximity sensors together with digital signal processing to increase the robot's estimated range.

Comparing this with the results reported by Nembrini et al. (2002) for the β-algorithm, it is proven by Bjerknes (2009) that the ω-algorithm has a much more stable performance. For swarm aggregation without beacon taxis, Nembrini reports that the β-algorithm never maintained robot aggregation for more than a few minutes at the best. Whereas in the initial set of experiments presented and discussed in Bjerknes (2009), the ω-algorithm maintained aggregation for more than 15 minutes for each experiments. As will be made clear in the section on scaling, the algorithm have been tested on swarms with five to twenty robots, with the increments of five for each experiments. In all these experiments the ω-algorithm has performed without any loss of robots, and always reached the beacon successfully.

While Bjerknes (2009) original motivation for the development of the ω-algorithm was to study fault-tolerance and scalability in real robot swarms, an added benefit of the algorithm, is that it completely frees up wireless communication bandwidth for full use in sensor network applications. However, due to the fact that robots are only able to estimate the centre of the swarm when performing a coherence move, there is a slight risk of individual robots becoming disconnected, requiring a higher swarm density (controlled by the w parameter), to mitigate the risk. This results in reduced area coverage in comparison to the β-algorithm, limiting its usefulness in sensor network tasks which attempt to provide maximum area coverage.

2.4 Swarm Intelligence Approaches to Swarm Robotics

There are a number of current research areas in the field of swarm robotics that uses swarm intelligence approaches as the source of inspiration. These projects confirm the enormous interest in the field of swarm robotics. Some of the researches as described in Jevtić & Andina (2007) include the Swarm-bots project, Pheromone Robotic project and I-swarm project. Further to these, there is also a recent project in swarm robotics; the Symbrion and Replicator project, which focus on the development of symbiotic evolutionary robot organisms based on bio-inspired approaches as well as modern computing paradigms.

The Swarm-bots project [2], which has been concluded in March 2005, aimed to study new approaches to the design and implementation of self-organising and self-assembling artifacts. The objective of the research is the design, hardware implementation, test and the use of self-assembly, self-organising and metamorphic robotic systems which were

[2] readers can refer on the detail and publications on the research of Swarm-bots project at http://www.swarm-bots.org.

composed of a swarm of robots Jevtić & Andina (2007). Inspired by the collective behaviour of social insects colonies, a swarm of simple robots, referred to as s-bots were design which were capable of autonomously carrying out individual and collective behaviour by exploiting local interactions among the robots in the environment as well as the robots and their environment. The projects had demonstrated that the swarm of the s-bots can be used for a collective transport, or to reach the points hardly reachable by a single unit of robot.

The Swarmanoid project [3], which aims to build on the previous research of Swarm-bots and provide additional research into design, implementation and control of novel distributed robotic systems capable of operating in a fully 3-dimensional environment (Jevtić & Andina, 2007). Current results of the research have already been seen with a swarm of robots that are able to locate heavy objects and join together to lift and transport the heavy object to a new location. More recently this project has also developed a heterogeneous swarms with different types of robots in the environment performing different tasks such as observing and providing situational awareness within a 3-dimensional environment.

The Pheromone Robotics project [4] aims to provide a robust, scalable approach for coordinating actions of large numbers of small-scale robots to achieve large scale results in surveillance, reconnaissance and hazard detection (Jevtić & Andina, 2007). This project is inspired by the chemical markers (pheromone) used by insects (especially by ants) for communication and coordinations. Results from the research have shown that the robots are able to perform complex tasks such as leading the way through a building to a hidden intruder as well as locating critical choke points in the environment. This concept can be used for search and rescue operations that is dangerous to be performed by human. For example team of robots can be sent to the dangerous site to investigate environmental parameters, search for survivors, and locate sources of hazards such as chemical or gas spills, toxic pollution,pipe leaks, radioactivity, etc (Jevtić & Andina, 2007).

The I-Swarm project [5] is focussing on developing real micro-robotic swarms, looking towards ants for inspiration in distributed and adaptive systems as well as in self-organising biological swarm systems (Jevtić & Andina, 2007). The project aims to have swarm of huge number of heterogeneous robots, differing in the type of sensors, manipulators and computational power. The robots in the swarm is expected to perform a variety of applications. It includes micro assembly, biological, medical or cleaning tasks.

The most recent projects in the field of swarm robotics inspired by swarm intelligence approaches are the Symbrion and Replicator projects [6]. The main aim of the project is

[3]This project is a follow-up and continuation of the Swarm-bots project and interested readers can refer to http://www.swarmanoid.org.

[4]for further description on this project, readers can refer to http://www.pherobot.com

[5]http://www.i-swarm.org

[6]for list of publications and full descriptions of the project, readers can refer to http://www.symbrion.eu

to develop novel principles of adaptation and evolution for multi-robot organisms based on biologically inspired approaches. The biologically inspired evolutionary approaches combined with robot embodiment and swarm-emergent phenomena in swarm robotics, may enable the robot organisms to autonomously manage their own hardware and software mechanisms. In this way, artificial robotic organisms become self-configuring, self-healing, self-optimising and self-protecting from both hardware and software perspectives leading to an extremely adaptive, evolveable and scalable robotic systems that can be used in performing engineering tasks. The robot organisms also are able to reprogram themselves without human intervention and supervision and for new, previously unforeseen, functionality to emerge.

2.5 Conclusions

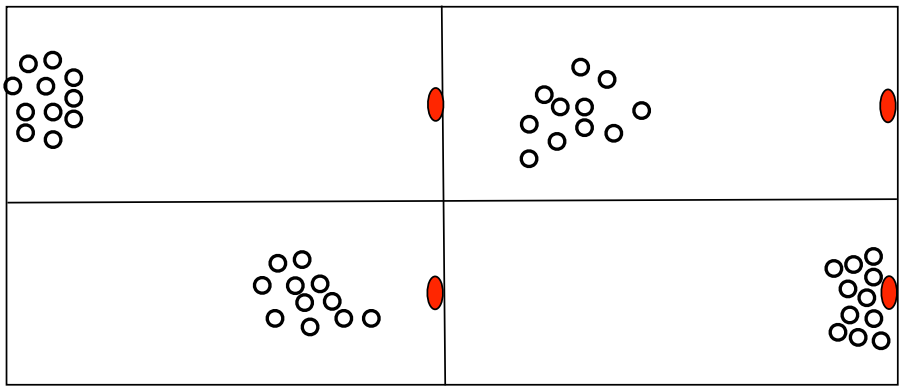
In this chapter we have discussed on the field of swarm intelligence in section 2.1 and its algorithms: particle swarm optimisation (PSO) and ant colony optimisation (ACO). Inspired by swarm intelligence, we further explore the field of swarm robotics describing the main characteristics such as robustness, flexibility and scalability, its applications and challenges and issues in swarm robotics in section 2.2. We then focused our discussion on one specific task in swarm robotics; the *swarm beacon taxis* in section 2.3. Few algorithms have been developed for swarm beacon taxis including α, β and ω algorithms which have different characteristics in communication and symmetry breaking mechanisms. We then discussed on other swarm intelligence approaches in swarm robotics in section 2.4 with the descriptions on some swarm robotics project such as the Swarm-bots project, the Pheromone Robotic project, the I-swarm project and the recent Symbrion and Replicator project.

As mentioned in this chapter, even though the principles of swarm robotic systems are well understood, there is a need for researchers in this field to study the challenges and the issues that have been highlighted in section 2.2.1. Such issues are sensing and situational awareness; power and energy autonomy; actuation and locomotion; safe navigation in unknown physical environments and safety and dependability (Winfield, 2009b). By developing systems that try to solve these issues will lead to the development of real world applications that involves swarms of robots in the future.

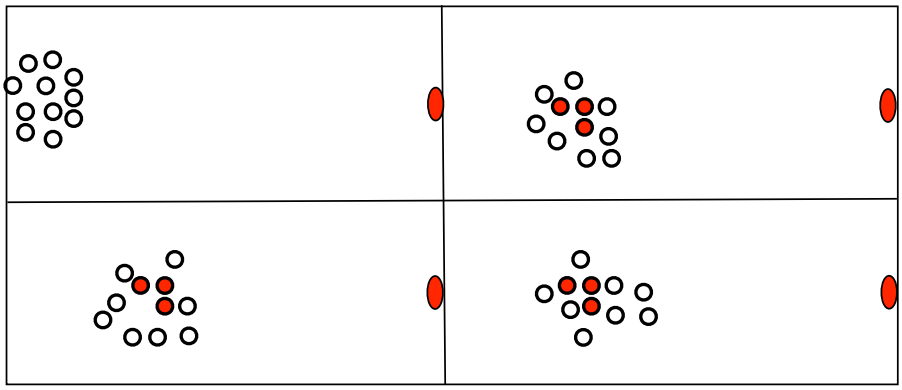
In swarm beacon taxis, the effect of a partially-failed robot to 'anchor' the swarm leading to impeding its taxis toward the beacon is shown pictorially in figure 2.6. This is the effect of the failure of robots' motors. If in the swarm there is only one or two robots that fail, the swarm can still moves towards the beacon. This is a form of *self-repairing*

and http://www.replicators.eu

mechanism inherent in the ω-algorithm (Bjerknes & Winfield, 2010). For example, with complete failure of a robot, the 'dead' robots simply become obstacles in the environment to be avoided by the other robots of the swarm. However the swarm will also experience a serious effect which causes the partially failed robot to 'anchor' the swarm. In swarm beacon taxis, this can only happen if the *anchoring force* caused by the effect of the robot's motors are greater than the *beacon force*.



(a) Swarm beacon taxis without failure. Here we see the swarm successfully move over to the beacon on the right-hand side of the image



(b) Swarm beacon taxis with failure. Here we see the swarm stagnate, and an anchor point emerge, with the swarm now being unable to continue towards the beacon

Figure 2.6: Swarm beacon taxis with and without anchoring

In emergent swarm taxis, a certain number of robots are necessary in maintaining the emergent property. A reliability model (k-out-of-n-system model) of the swarm in swarm beacon taxis has been developed and the results show that there is a point at which swarms are no longer as reliable as first thought (Bjerknes & Winfield, 2010). The result suggests that in a swarm of ten, then at least five of the robots have to be working in

order for swarm taxis to emerge. This would indicate that in order for the swarm to continue operating then some form of *self-healing* mechanism is required aside from *self-repairing* mechanism which is already available in swarm beacon taxis (Bjerknes, 2009). In this thesis, we propose that this issue can be solved by having an immune-inspired fault-tolerant swarm robotics so that the systems could adopt self-organised survivability.

CHAPTER 3

Artificial Immune Systems

As part of our goal of developing Artificial Immune Systems (AIS) to address engineering problems, in accordance to conceptual framework approach, we explore the background of applied AIS in this chapter. We begin by providing an overview of immune systems in section 3.1, covering the brief summary of the field of AIS and its algorithms which we divide into the swarm-like and non-swarm immune inspired algorithms. Secondly we discuss on the immune algorithms for swarm robotic systems in section 3.2. Thirdly we present the review of the design of AIS with conceptual framework in section 3.3. Then, we provide a discussion on the modelling and simulation of immune systems by describing some of the modelling and simulation techniques in immune systems including diagrammatic modelling and agent-based simulation approaches in section 3.4. We also discuss the engineering-informed modelling approach proposed by Hart & Davoudani (2011) in this section. We then turn our attention to the process of granuloma formation in section 3.5 which is the process in immune systems that is studied during this research. Here, a detailed explanation is given, including the meaning and the cells involved during the process of granuloma formation. We also discuss the mapping from the process of granuloma formation to swarm of robots in this section. We conclude this chapter in section 3.6 by discussing the importance of taking inspiration from aspects of the immune systems such as the process of granuloma formation in engineering systems and specifically in engineering swarm robotic systems.

3.1 Artificial Immune Systems and Its Algorithms

AIS is defined by de Castro & Timmis (2002) as:

> *adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving*

This definition highlights the fact that AIS is used in describing a wide range of systems, taking inspiration from different aspects of immunology. Many aspects of immunology have been studied, identified and proposed as mechanisms that can be implemented in the context of engineering. Like other bio-inspired computing paradigms, AIS attempts to capture the properties of biological systems upon which they are based. In the case of immune systems, appealing properties from computational perspectives have been identified, including pattern recognition, learning, memory and self-organisation (Timmis et al., 2008a). Immunology, therefore, provides a diverse source of inspiration for computer scientists, creating much scope for work within the area of AIS.

Over the last 15 years, AIS has matured from a collection of immune theories to the development of AIS algorithms. This can be seen from the collection of proceedings of the previous conferences on AIS (Hart et al., 2010b; Andrews et al., 2009; Bentley et al., 2008; Castro et al., 2007; Bersini & Carneiro, 2006; Jacob et al., 2005; Nicosia et al., 2004; Timmis et al., 2003; Timmis & Bently, 2002). These proceedings provide examples on how AIS algorithms are used to solve the wide area of computer science and engineering problems, such as optimisation, classification and intrusion detections. Examples of the algorithms are clonal selection algorithm (de Castro & Zuben, 2002), negative selection algorithm (Forrest et al., 1994), immune network algorithm (de Castro & Zuben, 2000b) and dendritic algorithm (Greensmith, 2007).

Timmis et al. (2010a) have divided the immune inspired algorithms naturally into two categories; non-swarm and swarm-like algorithms. These algorithms have taken inspiration from a diverse range of immune functions from the immune systems, that occur across varying levels of details. Since the aim of this thesis is to develop an immune-inspired swarm robotic system, we describe the AIS algorithms according to these categories. Using these categories, we can highlight and identify both differences and similarities between immune and swarm inspired algorithms and illustrate when and where the algorithms embody the principles of swarm intelligence as described by Dorigo & Birattari (2007), which is discussed in section 2.1. A detailed discussion of these algorithms lies beyond the scope of this thesis; however, the interested reader is referred to Timmis et al. (2008a,c) for a detailed discussion of the algorithms.

3.1.1 Swarm-like Immune-inspired Algorithms

As discussed by Timmis et al. (2010a), one of the immune mechanism that has influenced the derivation of swarm-like algorithms since the inception of AIS is immune-network theory. The main difference between the immune network algorithm and other immune algorithms is that the components of the system not only interact with antigenic components, but with the other components in the AIS. In other words immune cells in immune systems not only recognise foreign cells but also recognise each other, creating a structural and functional network of cells in the systems that dynamically adapts to stimuli over time (Timmis et al., 2010a). The interactions between cells in the immune systems give rise to the emergent property due to the existence of complex phenomena such as memory (Farmer et al., 1986) and other functionalities that can be observed in immune systems such as tolerance and reactivity (Hart et al., 2007, 2010a).

According to Timmis et al. (2010a), the immune network theory depicts the idea of the most commonly used swarm algorithm: the particle swarm optimisation (PSO) algorithm. The PSO algorithm exploits the direct interaction capabilities between the members of the swarm in the environment. It is also based on a 'particles' or a simplified social model of the local interaction between agents. This model can also be viewed as a set of vectors around a region defined by the particles' historical best position and the best position of other individuals (Eberhart et al., 2001). The topology of the swarm governs which particles from the swarm can influence a particle. In this case, the swarm is usually influenced by the global best particle, as well as particles which are available in the particles' local neighbourhood (Timmis et al., 2010a).

The use of immune network theory in swarm-like systems stems from early work such as aiNet (de Castro & Zuben, 2000b), the modified version of CLONALG (de Castro & Zuben, 2002), which includes suppressive interactions between the antibody components. Timmis & Neal (2001) give an example of the aiNet algorithm for clustering data that is inspired by immune network theory. The algorithm uses a network of antibody components, that adapt to match a population of input components (antigen) representing the data to be clustered. aiNet was originally used in data clustering (Timmis & Neal, 2001) but was later adapted for optimisation (Timmis & Edmonds, 2004) and robot navigation (Neal & Labrosse, 2004). Even though aiNet has been widely used in both swarm-like systems and more general pattern recognition and clustering problems (Andrews & Timmis, 2005; Coelho & Von Zuben, 2006), it has been noted that there is little theoretical work of the immune network approach (Timmis et al., 2008a). Timmis et al. (2008a) reports work by Stibor & Timmis (2007) has shown that aiNet suffers problems when clustering non-uniformly distributed data. This is primarily because of the way the suppression mechanism operates in the algorithm (via a distance matrix), which leads to

either insufficient retention of information in the clusters from when it started, or the algorithm producing more elements to represent the input space than it began with.

Moving on from immune network inspired algorithms, the derivation of swarm-like algorithms has been influenced by dendritic cell trafficking mechanisms from the innate immune system. This also exploits the indirect interactions in a swarm via chemical signals. In this case, the chemical signals are the chemokine signals that form the chemical gradients in the body. This is a source of interaction in immune system, in much the same way that the ant colonies utilise pheromone gradients as a source of information to move from one position to another. Dendritic cells are often referred to as the sentinels of the immune system, in that they circulate through the body, scouting for chemical signals present in the tissues, and then return that information to the lymph node.

Early work on dendritic cell algorithms has been applied to anomaly detection (Greensmith, 2007) and behaviour classification on a robotics platform (Oates et al., 2007). However, mechanisms inspired by the dendritic cells have also been applied to swarm-like systems in the domain of self-organising wireless sensor networks (Davoudani et al., 2007, 2008; Davoudani & Hart, 2008; Hart & Davoudani, 2009). Davoudani et al. (2007) first draws an analogy between the requirement for the network body to be able to sense its state and react accordingly to the acquired information. This is necessary for self-organising wireless sensor networks to be able to monitor their state and react accordingly, focusing on Specknets, which are self-organising networks of sensor devices capable of individually performing limited computation and processing without any central controller. Davoudani et al. (2007, 2008); Davoudani & Hart (2008); Hart & Davoudani (2009) then utilise 'circulating radio messages' that mimic the functionality of dendritic cells in the immune system for collecting information from nodes in the sensor networks. This information is then returned to specks designated as lymph nodes by exploiting the idea of artificial chemical gradients for path-finding. This concept is similar in the immune systems, where the dendritic cells migrate from areas of infection back to the lymph node in order to release a further chemical signal which attracts new immature dendritic cells to the infectious area, which will then create a positive feedback loop. Information collected at the lymph node is interpreted through direct recognition of the information by artificial T-cells, and acted upon appropriately, that is applied throughout their work (Davoudani et al., 2007, 2008; Davoudani & Hart, 2008; Hart & Davoudani, 2009).

3.1.2 Non-swarm Immune-Inspired Algorithms

Non-swarm immune inspired algorithms are algorithms that cannot be described as a swarm-like systems. The most common algorithms in this category are those inspired by the processes of negative selection and clonal selection (Janeway et al., 2005)) in the

immune system. Based on the discussion in Timmis et al. (2010a), these algorithms are categorised as the non-swarm immune inspired algorithms mainly because there is no similarities between these algorithms with swarm inspired algorithms and principles of swarm intelligence as articulated by Dorigo & Birattari (2007).

Negative selection algorithm, plays an important role in the research of AIS. It was first proposed by Forrest et al. (1994) to detect data manipulation caused by a virus in a computer system. Due to the ability of discriminating between self and non-self in negative selection, it fits naturally into the area of intrusion detection and into the area of computer security (González et al., 2002; González & Dasgupta, 2003). A comprehensive survey paper on the development of negative selection algorithm in intrusion detection has been undertaken by Ji & Dasgupta (2007), in an attempt to identify the fundamental characteristics of this family of algorithms and summarise the diversity of the algorithm. Detailed discussion of negative selection algorithm is beyond the scope of this thesis; however, the interested reader is referred to Timmis et al. (2008a) and Timmis et al. (2008c) for a detailed discussion of them.

Utilising the features of clonal selection principles, de Castro & Zuben (2002) have proposed the clonal selection algorithm called CLONALG, which has been used to perform the tasks of pattern matching and multi-modal function optimisation (de Castro & Zuben, 2000a; Walker & Garrett, 2003; Kelsey & Timmis, 2003; White & Garrett, 2003). Clonal selection algorithms have been applied to a wide range of optimisation and clustering problems (Garrett, 2005; Cutello et al., 2005; Hart & Timmis, 2008; Bernardino et al., 2010). In Cutello et al. (2005), the clonal selection algorithm is been applied in solving optimisation problems in a large set of classical numerical functions. For both cases, the clonal selection algorithm has given a promising results when compared with other algorithm such as differential algorithm and swarm intelligence algorithms. Meanwhile, in Bernardino et al. (2010), a similarity-based surrogate model is used with clonal selection algorithm in order to improve its performance when solving optimisation problems involving a computationally expensive object function. For detailed discussion of clonal selection algorithm and their applications, interested readers can refer to Timmis et al. (2008a) and Timmis et al. (2008c).

3.2 Immune Algorithms in Swarm Robotics

The motivation for adapting immune systems to swarm robotics for inspiration is the observation that immune systems is able to achieve homeostasis, which is the ability of the immune systems to maintain a stable state in a dynamic environment.

In immune systems, there are mechanisms that monitor perturbations in the inner

states of the system, and react in ways that promote smooth operation across the various chemical, mechanical, neural and other components (Cohen, 2000). The innate and adaptive immune systems work together to destroy, contain or repair depending on the circumstances occur in the immune systems. The mechanisms of homeostasis in dynamic environments have started to be exploited by the roboticists to design self organising behaviours such as learning (Whitbrook et al., 2008) and foraging (Tsankova et al., 2007).

According to (Polack, 2010) the complex computer architecture in robotics should be able to adapt, to repair or replace components, and to respond to new components, new challenges or changed requirements. Thus the robots need to determine what they need to do by analysing their current environment, which essentially means recognising different sort of obstacles exist in the environment or different sorts of malfunction in themselves or in other robots. This is considered as a specific self-organising ability in each of the robots in the system to respond to different types of change in the environment and its own self. To date, there is a series of immune-inspired research projects in swarm robotics that have been put forwarded by AIS researchers, as well as roboticists in trying to achieve the desired properties in swarm robotic systems.

As described in section 2.2.1, one of the major challenges in swarm robotics is the ability of the systems to work in a desired period of time (long-term autonomy). In order to achieve this, robots and their environment must have two properties: autonomy and self-sufficiency (McFarland & Spier, 1997). Autonomy means that the robot has independence of control and is able to make their own decision and govern their own behaviour, whilst self-sufficiency denotes the ability of a robot to maintain itself in a viable state for a long periods of time. Thus in achieving long-term autonomy, the robot must be able to manage its own energy source in any way that is not dependable upon human intervention (McFarland & Spier, 1997). Since the autonomy of a robot is limited by its own on board energy resources, it is therefore important for the robot to spend their energy economically to full fill the principle of self-sufficiency. One of the ways to achieve this property is by adapting immune systems to the system.

An example of immune-inspired solution for robotic systems has been proposed by Mokhtar et al. (2009). An important aspect of the AIS described by Mokhtar et al. (2009), is the lymphocytes self-detector: an array of values that express the state of a robot as it is performing a particular task. An instance of the self-detector includes the resource characteristics of the task, the state of each of the components during this task, and a number of measured actuation outputs for the running of the task. In the system each self-detector includes a two-part health measure that records the apparent health of the whole system and a measure of whether the self-detector is a candidate for (probabilistic) removal from the self-detector set.

Another example is the SYMBRION/REPLICATOR project [1], which is focussing on the long-term autonomy of the robots. It has a target of operating a 100-robot swarm for 100 days (Kernbach et al., 2010). In this project, the conceptual architecture of the swarm of robots, which are adaptable and can self-organised to join together to form *organisms* of robots and maintain higher robotic organism. The objective is to maintain the long-term autonomy, keeping them operating as long as possible, whilst undertaking some simple task such as exploration (Timmis et al., 2010a). Kernbach et al. (2010) describe an architecture in which each individual robots will have an AIS or artificial homeostatic hormone system (AHHS) that is used to calculate passing information between robots' components or to determine when its operation is moving outside the normal limits. (Timmis et al., 2010b) shows how an immune inspired approach can manage fault tolerance in the SYMBRION robots and robot organisms. Clearly, this challenge will be an interesting test bed for the development of robotic systems specifically in achieving autonomy and self-sufficiency of the system.

The work on swarms and robot organisms described above is concerned with adaptability and fault tolerance the ability to respond to new challenges as well as maintaining dynamic homoeostasis in the organism rather than the malfunctioning of each individual robots and the system. In the context of the SYMBRION robots, Timmis et al. (2010b) propose fault tolerant swarm systems, in which system-level behaviour can be achieved even when some individual robots have failed in such a way that they cannot perform a task. Anomaly detection takes place both in the robots (component systems) and in the whole swarm (the robot organism). Timmis et al. (2010b) propose three types of anomalies that are detectable using the immune-inspired solution that they propose: theoretically-impossible states (Type 0), things which are inconsistent in the current behaviour of the system (Type 1), and longer term divergence (Type 2). At the robot level, type 0 anomalies could be detected using approaches inspired by the innate immune system, which holds a model of the 'normal' system., which holds the model of 'normal' in the system. Detection of type 1 and type 2 anomalies could be seen as analogous to identifying a new antigen; this would use inspiration from the combined innate and adaptive immune systems. The processes rely on continuous analysis of logged performance and audit data, with comparison both against what is expected, and against what has been seen in the past. The recognition of type 1 anomalies is based on an assumption that future behaviour of the system can be predicted from the history and current state of the system (Timmis et al., 2010b).

For organism-level anomaly detection,Timmis et al. (2010b) describe two approaches to sharing of immunological information. The first requires robots to pass generated im-

[1] http://www.symbrion.eu

mune information (danger signal values, clones, receptors) to other robots. The second (which has been used in practice for error diagnosis in ATMs (de Lemos et al., 2007)), maintains a central set of network detectors derived from the immune information of individual components. The cells, instances, signals etc are subject to the same evolutionary processes and maintenance criteria to ensure relevance and freshness.

The SYMBRION proposals for immune-inspired fault tolerance in swarm robotic systems has not yet been reflected in real implemented fault tolerance. Further effort is needed effort is needed to adapt immune-inspired solutions for anomaly detection, there is limited research being done to build systems that allows robots to remain self-sufficient and able to manage its own energy source between themselves. Specifically the system must include some recharging mechanisms i.e. recharging stations, rechargeable batteries or any self-recharging devices.

Non-immune-inspired solutions to the recharging problem have been proposed by Muñoz Meléndez et al. (2002) and Arvin et al. (2009). In Muñoz Meléndez et al. (2002), a robot that does not have enough energy will move to the charging station provided in the environment and charge its own energy. Meanwhile in Arvin et al. (2009), there is a removable charger available in the environment to perform the charging task. In both solutions, robots that do not have enough energy can only be charged by a specific device in the environment: the charger or the removable charger, which is not sufficient when the level of energy of the charger and the removable charger is low and not enough to charge the robots then, they need to be changed or recharged that needs human intervention in dealing with the process. However, there is a need in swarm robotic systems to have a mechanism that allows robots to transfer energy between themselves.

Melhuish & Kubo (2007) propose energy sharing among robots, or *trophallaxis*, in which each robot can donate an amount of its own internal energy reserve to another. Using a simulated test environment, they explore different strategies for energy transfer. The results suggest that robot trophallaxis may confer benefits to tasks that require multiple robots in the system, in the form of task completion, performance and survivability (Melhuish & Kubo, 2007). Most significantly, in trophallaxis, robots can charge each other even when faced with very inefficient energy transfer mechanisms.

3.3 Approaches in Developing Immune-inspired Swarm Robotic Systems

The way in which the immune system is used to inspire existing algorithms in swarm-like robotic systems varies in precision and extent. Stepney et al. (2005) argue that bio-inspired algorithms are best developed and analysed in the context of multidisciplinary

conceptual framework that provides biological models of biological properties. Here, *probes* (observations and experiments) are used to provide a view of biological systems. Based on this view, *simplified abstract representations*, known as models of the biology are built. From these biological models, the *analytical computational framework* is built and validation analysis is undertaken. This framework served as an exposition of the principles for designing and analysing bio-inspired algorithms that are applicable to any non-biological problems. This will lead to a design of algorithms that carefully extract the biological properties. According to Stepney et al. (2005), one could attempt to produce a computational framework based on biology without any algorithm in mind, hoping to come across any applicable computational problems that need to be solved, which would seem to be a very difficult task. Therefore, as suggested in Timmis et al. (2008b), it is easier to orient these steps towards some particular problem giving necessary focus to the modelling work (Freitas & Timmis, 2007).

In order to guide the modelling and simulation process, we review a technical report by Andrews et al. (2010) where they propose a CoSMoS process that is believed can guide the modelling, simulation and analysis of complex system in a principled manner. The process is depicted in figure 3.1, which demonstrates the CFA and CoSMoS process in its current standing. In exploring the modelling and simulation process in the conceptual framework, the CoSMoS minimal process proposed by (Andrews et al., 2010) are presented. Based on the figure 3.1, five distinct products have been identified by the authors. Quoted verbatim from Andrews et al. (2010), the five CoSMoS products are:

- research context: captures the overall scientific research context of the CoSMoS project; including the motivation for doing the research, the questions to be addressed by the simulation platform, and requirements for validation and evaluation.
- domain model: encapsulates understanding of appropriate aspects of the domain into explicit domain understanding, focuses on the scientific understanding.
- platform model: comprises design and implementation models for the simulation platform, based on the domain model and research context.
- simulation platform: encodes the platform model into a software and hardware platform upon which simulations can be performed.
- results model: encapsulates the understanding that results from simulation: the simulation platform behaviour, results of data collection and observations of simulation runs.

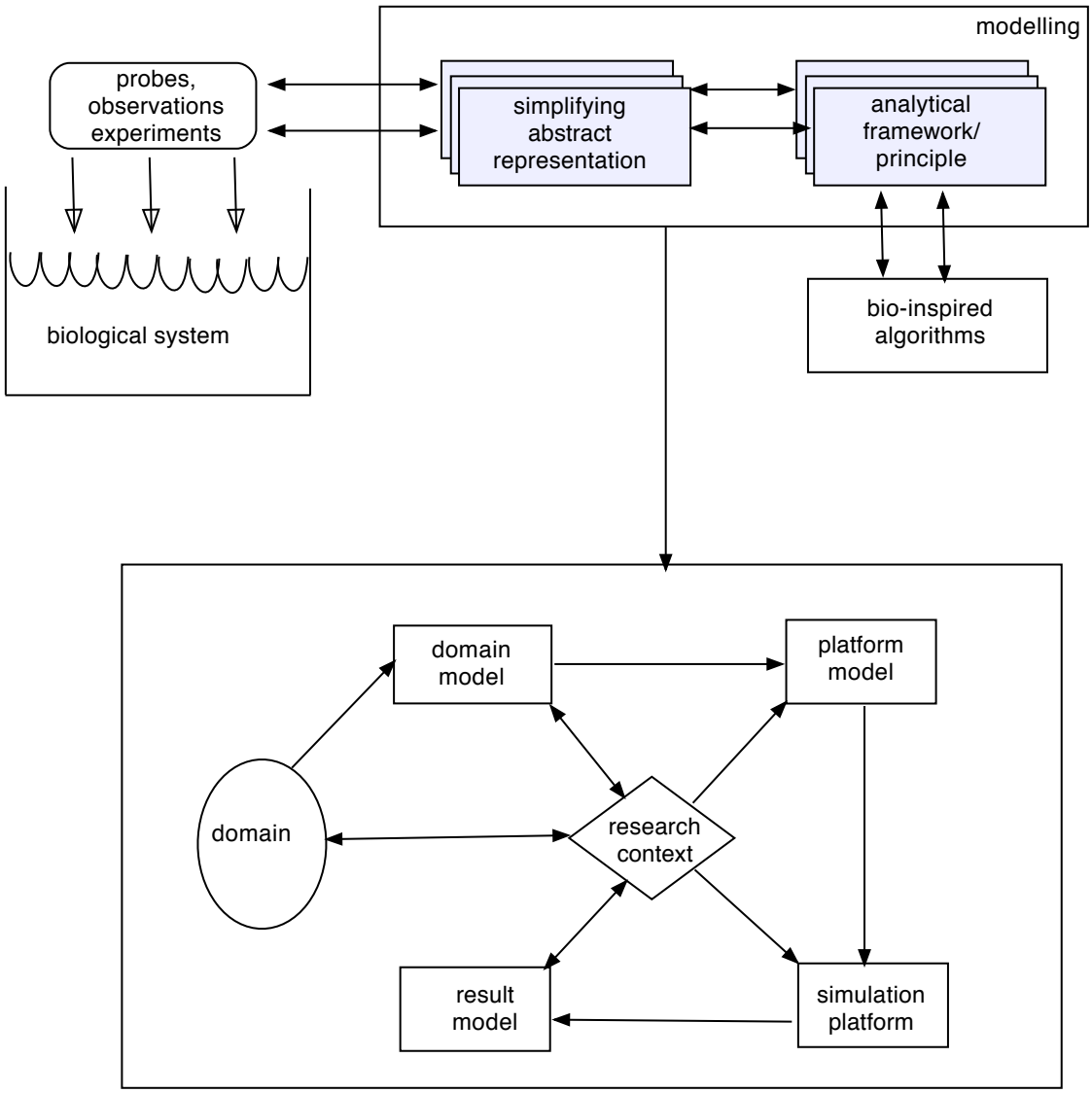


Figure 3.1: Modelling process in conceptual frameworks Stepney et al. (2005) with CoSMoS minimal process Andrews et al. (2010).

The domain model details the current understanding of the biological domain as held by the modeller, and captures the behaviours present in the biological domain with multiple levels of abstraction. It captures the high level depiction of behaviours exhibited at a system wide level, to the low level entities of the system as well as how the interaction between both levels exist. In the CoSMoS process, validation of the domain model plays an important role, and is carried out by a domain expert. If the domain model is invalid during the validation process, then the understanding that the modeller has of the system is most likely not acceptable, and the system is unlikely to be representative of the real biological domain. Based on the concepts captured in the domain model, the software model is constructed. It is tailored towards the design and implementation of the simulator itself. During this process, any explicit notions of the behaviours and the emergent properties will be removed. The specific concepts for the implementation of the model may be introduced. According to Andrews et al. (2010), the software model is constructed at a low level with a bottom up approach. Once the platform model has been developed, the simulation model is derived through experimentation and the observations of the simulation. Again, the validation can be performed between the domain model that has been prepared earlier and the simulation model. If the simulation can capture the desired behaviours present in the domain correctly, then the simulation model should closely resemble the domain model which has been prepared during the earlier stage. As mentioned in Andrews et al. (2010), the CoSMoS process iterates through the three phases until a satisfying stopping condition is met. At each stage, the products may be revised and updated, though it is also possible to simply review the products and pass them on to the next stage. Further to the five CoSMoS products described above, in the CoSMoS process one should be explicit about why the simulation and modelling work is taking place, since the modelling and the simulation work will be driven according to the motivation in mind.

3.4 Modelling and Simulation of the Immune Systems

As part of our work in using conceptual framework in developing novel AIS in section 3.3, we present our investigation into the state of the art in modelling and simulation of immune systems that will act as a basis for our model and simulation work in chapter 4. We first discuss on the meaning of model, followed by the discussion on the concept of simulation.

Lehman (1978) discusses the relationship between theory and model, defining theory as:

a general statement of principle abstracted from data and observation that

purports to explain the behaviour under consideration. Theory will ascribes certain properties to the individual involved and uses those properties to explain behaviour

Lehman (1978) further emphasises that for scientists that are interested in exploring attitudes and social interaction, they can undertake the following procedures:

1. do the theory-testing experiment;
2. abstract the essential elements of the situation and theory into simple form like drawing square and patterns;
3. write computer programs expressing some of the essentials of the social situation and associated theory.

Based on the above-mentioned procedures, Lehman (1978) then describes the model as:

representation of a theory that may be represented in any of several different media

The model therefore is not an explanatory statement. It is only a simplification and abstraction of certain key elements of the theory. A model allows us to explore the consequences of theory but not to explain its behaviour (Lehman, 1978). Polack et al. (2008) distinguish two orthogonal modelling goals: *description* and *definition*. A descriptive model might capture aspects of the observed high-level behaviour; where modellers use models to capture what they observe. Meanwhile, a definitional model is more typical of conventional engineering, insofar as it expresses the required characteristics of a system at an appropriate level of abstraction. This type of model can be refined, translated and analysed, to improve understanding of system characteristics (Polack et al., 2008).

Cohen (2007) states that in order for models to be useful for biologists, they should stimulate experimenters to experiment, while taking into account the data in hand. He also encompasses several characteristics for models to be useful for biologists. The characteristics, quoted verbatim from Cohen (2007) are:

1. bottom-up: a useful model must begin with the data, and move upwards. The data are the essential part in any model.
2. object-oriented: a model is useful to biologists if it deals with the objects; not only dealing with concepts or numbers. This is because biologists do their experiment with objects (for instance molecules, cells and organisms), changes of the state of each objects and how the objects interact.

3. dynamic: the model must allow us to deal with the system's dynamics.

4. multi-scalar: biologists use models to observe and study the emergent properties. Thus, biologists must be able to zoom into cells to see the molecular interactions between cells and how the cellular interactions affect the objects in the systems.

5. modular: a useful model should be modular; new data can be added to the existing model without having to redo the whole model.

6. interactive: a useful model must act as an introduction to doing real experiments (in vitro and in vivo) allowing in silico experiments to be conducted. Playing and experimenting with the model will allow biologists to see the consequences of modifying the component, concentration and reactions of the objects.

7. realistic: a useful model must be able to show to the biologists things that trigger a 'wow' response. It must be able to communicate to us in a language that can easily be understood, such as visual representations.

Cohen (2007) finally concludes in his paper by giving some conditions for modelling biological systems. The models need to suit the reality of living systems, not merely adhere to top-down logic, abet experimentation by stimulating new ideas for novel experiments and engage the minds of the experimentalist with understandable representations of the model for current and future references.

A computational model such as simulation has been used in modelling the biological systems for many years. Traditional simulations generate from output from equations such as differential equations, and markovian models have been developed to mirror trends or behaviour in, for instance, biological populations (Polack et al., 2008). It is also beneficial as they support hypotheses as well as the connection and causalities in the biological systems that are being focused on (Jackson et al., 2000).

As described by Kleinstein & Seiden (2000), simulation is *an imitation of the real system, where it has inputs entered by the user and outputs that are being observed throughout the simulation run*. Among the most compelling reasons for simulation are the theoretical considerations, which have several potential advantages. The advantages highlighted by Lehman (1978) are as follows:

1. clarification of theoretical statements;

2. may also be a critique of a theory;

3. may lead to a more complete expression of the theory;

4. allow to generate and explore new hypotheses and implications of the theory.

According to Andrews et al. (2008) simulations have mainly two purposes. Firstly, some simulations are built in co-operation with research scientists in an effort to improve understanding of natural systems under study, and secondly, simulations are also built as artificial systems to construct and explore alternatives to realities. Andrews et al. (2008) further explain that computer simulation for biological phenomena is important because static models cannot capture the dynamic features that characterise the behaviour of complex systems. They give an example from Polack et al. (2008) where they highlight that systems biologists increasingly adopt conventional software engineering design diagrams to express static structures and patterns of interaction in their models, where the modelling approaches that they use cannot express time, space or the features and consequences of large numbers of interacting instances Andrews et al. (2008).

Further to their discussion, Andrews et al. (2008) also highlight that a scientifically-valid simulation has to extract suitable environmental aspects, at an appropriate level of abstraction; where it also has to provide evidence that its environmental representations; as well as its scientific model, are adequate abstractions from the biological reality. Most of the time simulation is judged by its ability to produce something like the expected results, and little concern are given for the quality or scientific relevance of the underlying simulations (Epstein, 1999). Scientific validity is explained by Andrews et al. (2008) as a way that is possible to demonstrate with evidence, how models express the scientific realities, where it implies both adequate abstraction, and adequate development processes.

Andrews et al. (2008) further explain that the simulation process is an iterative cycle, which includes an experimentation link between the problem entity and the simulation. This allows the developers to test the simulation elements and settings, as well as comparing the results obtained from the simulation with the problem under study. The second important aspect of the simulation process is the explicit inclusion of verification and validation of the simulation against reality, and the data used to test the model and simulation (Andrews et al., 2008). Similar validation concepts come from (Sargent, 1986), which has been pointed out by Andrews et al. (2008) as elaborating the validation aspects of the simulation process. This is shown in figure 3.2, obtained from Sargent (1985) that highlighted that a model should be developed for a specific purpose, and its validity determined with respect to that purpose. From the problem entity, a conceptual model needs to be developed in a suitable representation, such as diagrammatic models, and also mathematical or logical modelling paradigms. Sargent (1986) further highlights that the level of assurance needed depends on the purpose of the simulation, and they should be set independently from the development of the simulation. Sargent (1985) development process (lifecycle) for simulations explicitly incorporates verification and validation activities, and Sargent (1986) further proposes approaches to validating the simulation that has been previously

developed. This validation approaches are summarised in Andrews et al. (2008)

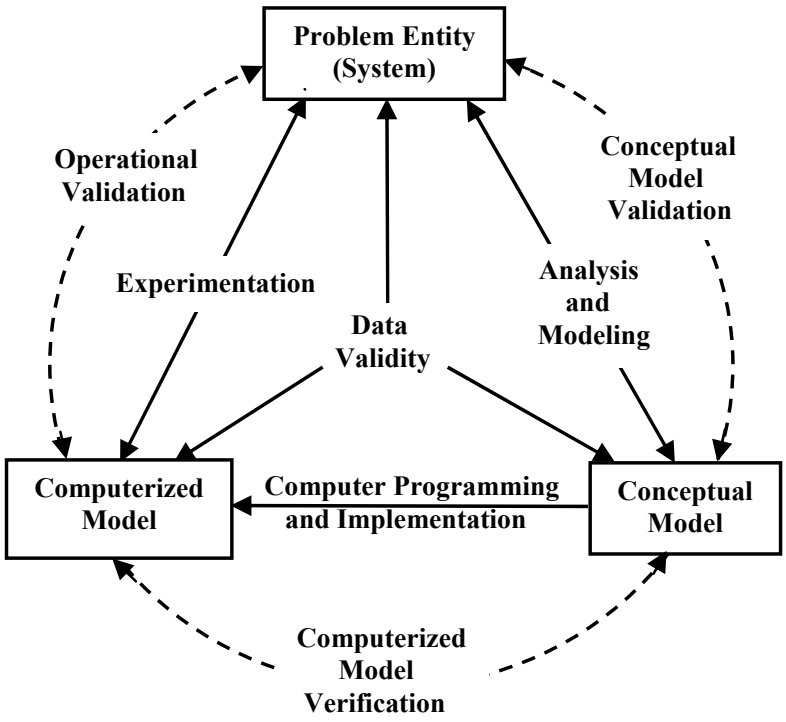


Figure 3.2: Sargent's model of the simulation development process (Sargent, 1985)

For inspiration for our modelling and simulation work, we turn to two main examples: diagrammatic and software engineering approaches, and agent-based simulation. In diagrammatic and software engineering approaches, we discuss the needs of graphical design notations in biological systems and some examples of the notations. We then describe the graphical notation in software engineering called the Unified Modelling Language (UML), which has been used in modelling biological systems. We then turn our discussion to the agent-based simulation, highlighting some of the simulation work that has been used in biological systems. For each approach we identify the advantages and disadvantages and for further references, users can refer to Forrest & Beauchemin (2007) and Polack et al. (2008).

3.4.1 Diagrammatic and Software Engineering Approaches

Graphical design notations have been around for quite some time, with their primary value is in communication and understanding. A good diagram can often help to communicate ideas about a specific design, particularly when we want to avoid a great deal of information and details. With diagrams it will ease the understanding either on a specific software system or a business process (Fowler, 2004). Diagrams can also act as the mean of communication between a team in an organisation to understand ideas. Although they are not,

at least yet, a replacement for textual programming languages, they are a helpful assistant (Fowler, 2004).

A survey on the needs of specialised diagrammatic modelling techniques for modelling biological systems have been summarised by Klipp et al. (2007). Even if it is somewhat unclear what the notations should look like, some of the main characteristics identified by Klipp et al. (2007) are: being able to describe graphical representation of biochemical networks, act as an assisting tool for experimental procedures and support model encoding and model exchange. Klipp et al. (2007) further highlight that specialised diagrammatic modelling techniques for biological systems are needed to ease the process of building models and quantitative simulations of complex biological systems. These can also help to facilitate collaboration and communication between modellers and experimenters from diverse scientific backgrounds. In Kitano (2003), the authors highlight the ambiguity that occurs in traditional biology diagrams that use visual representation to capture different types of stages of cells and molecules in biological systems. An example of this type of diagram can be seen in figure 3.3, which captures the relationship between molecules and their transition from one state to another. In this diagram, an arrow is used for activation, but another arrow, in the same diagram, may mean transition of states or translocation. Without consistent and unambiguous rules for representation, information is lost and wrong information could be disseminated (Kitano, 2003).

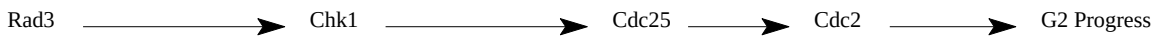


Figure 3.3: An informal biological pathway diagram (Kitano, 2003)

Some initial work has been conducted by researchers to bring out a standard diagrammatical representation of biological systems to capture the biological information (Kohn, 1999, 2001; Maimon & Browning, 2001; Demir et al., 2002; Kitano et al., 2005). These include some proposals on the specialised diagrammatical approach for modelling the molecular level of the biological systems. They include work by Kohn (1999) called a *Molecular Interaction Map*, Kitano (2003) known as *Kitano Process Diagram*, Demir et al. (2002) called *PATIKA*, *Diagrammatic Cell Language* proposed by Maimon & Browning (2001) and *Prototype Biological Description* proposed by Cook et al. (2001). Unfortunately, none of the proposals has been widely used for a variety reasons, as well as there being numbers of issues to overcome. For further descriptions, readers can refer to Kohn (1999, 2001); Maimon & Browning (2001); Demir et al. (2002); Kitano et al. (2005); Fowler (2004).

In addition to the tools developed by the biologists, there is a wealth of diagrammatic modelling experience in computer science and software engineering that has culminated in the unified modelling language (UML) (Fowler, 2004). The UML consists of different

types of diagram that can model different aspects of structure and behaviour. The advantage of the UML is its non-domain specific nature and subsequent ability to capture abstractions. The UML (and related diagrams such as statecharts) have started to become a powerful tool in modelling aspects of biological systems (Efroni et al., 2003, 2005; Read et al., 2009a,b). UML has become not only the dominant graphical notation within the object-oriented world, but also a popular technique in non-object-oriented circles. The difference between a graphical language such as UML and ad-hoc illustrations is that the symbols used in the UML are linked to a semantic (Fowler, 2004). Therefore with UML, it is possible to make a description less ambiguous and even automatically processable for computers. UML 2.0 defines thirteen types of diagram, divided into three categories; six diagram types represent static application structure; three represent general types of behaviour; and four represent different aspects of interaction (Fowler, 2004):

- structure diagrams: include the class diagram, object diagram, component diagram, composite structure diagram, package diagram, and deployment diagram;
- behaviour diagrams: include the use case diagram (used by some methodologies during requirements gathering), activity diagram, and state machine diagram;
- interaction diagrams: all derived from the more general behaviour diagram, include the sequence diagram, communication diagram, timing diagram, and interaction overview diagram.

Use case diagram is use to display the relationship among actors in the system. It gives a description of the behaviour of the systems as it responds to any request that originates from outside of that system. Use case diagram is also used to capture the functional requirements of a system by describing the interaction between a primary actor and the system itself, represented as a sequence of simple steps available in the system. Actors are something or someone which exist outside the system, and take part in a sequence of activities in a dialogue with the system to achieve certain goal. They may be end users, other systems or hardware devices. Each use case is a complete series of events, described from the point of view of the actor. In UML, use case diagrams are helpful in three areas (Fowler, 2004):

- determining features (requirements): use cases often generate new requirements as the system is analysed and the design takes shape.
- communicating with clients: their notational simplicity makes use case diagrams a good way for developers to communicate with clients.

- generating test cases: the collection of scenarios for a use case may suggest test cases for those scenarios.

A class diagram gives an overview of a system by showing its classes and the relationships among them. A class diagram is static; it only displays 'what' interacts but not 'what happens' when they do interact. Some of the relationships that can be explained with a class diagram are:

- association - a relationship between instances of two classes. There is an association between two classes if an instance of one class must know about the other in order to perform its work. In a diagram, an association is a link connecting two classes.
- aggregation - an association in which one class belongs to a collection. An aggregation has a diamond end pointing to the part containing the whole.
- generalisation - an inheritance link indicating one class is a super class of the other. A generalisation has a triangle pointing to the super class.

UML defines several forms of interaction diagram, of which the most common is the sequence diagram. Typically, a sequence diagram captures the behaviour of a single scenario that shows a number of objects and the messages that are passed between these objects within the use case (Fowler, 2004). A sequence diagram is useful for depicting the interaction between objects over time. Sequence diagram is often interpreted from top to bottom. A state diagram in figure is similar to an activity diagram, with the crucial difference being that the they do not contain activities, but states of an object. For further explanation regarding the diagrams, readers are pointed the original source of this section in Fowler (2004).

Recently, biological researchers have started to use object-oriented and UML diagrams for modelling the information and processes of biological systems (Johnson et al., 2004; Shegogue & Zheng, 2005; Signorini & Greussay, 2004). UML can also be used for conceptual modelling of biological systems (Bornberg-Bauer & Paton, 2002; Dori & Choder, 2007; El-Ghalayini et al., 2006; Paton et al., 2000), where it also can work as a tool for communication across disciplines (Heemskerk & Pavao-Zuckerman, 2003). The conceptual model is useful as it can express the descriptions of the concepts of the biological systems and captures the principal structural properties of the data, where towards the end it can be transformed in systematic ways for implementation, using any type of platform (Bornberg-Bauer & Paton, 2002). By far the most advanced use of the UML and statecharts in immunology is that of Efroni et al. (2003, 2005), who have built a sophisticated and predictive model of T-cell maturation in the thymus using a simulation

tool called reactive animation, which combines the execution of state-charts and other behavioural diagrams.

Signorini & Greussay (2004) have shown how a sequence diagram and an activity diagram can be used to capture the sequence of messages for a blood clotting systems. In the paper, the authors show how the comprehension of the sequence of messages and methods, between objects, is represented using a sequence diagram. Obviously, detailed timing analysis should add time values to the diagram as constraints or synchronisation patterns. However, Signorini & Greussay (2004) explain that those time values can only be investigated through in vitro experiments, and are themselves elements of a comprehensive model.

Flügge et al. (2009) present a case study to understand the process of granuloma formation by using UML diagrams. Flügge et al. (2009) capture the behaviour of each cell in granuloma formation using activity and sequence diagrams. Flügge et al. (2009) also make use the state diagram to describe the behaviour of an innate liver cell and macrophages. Despite class, activity and state diagrams, Flügge et al. (2009) investigate the applicability of sequence and communication diagrams in representing hypotheses. Based on the work done by Flügge et al. (2009), they come to an initial conclusion that UML diagrams can be explored in presenting not only static relationship and behaviour of each cells, but it can also be used in representing hypothesis for the biological cases.

Read et al. (2009a) have selected UML as a tool to construct a model of EAE (autoimmune disease in mice) and its regulatory before preparing the agent-based simulation, which serves as a concise detail of their understanding of the biological domain. Based on their work, they discovered that the constructions of the model has raised various questions on the biological model under study. They reported the following findings as the feedback of their experiences in constructing the model with UML (Read et al., 2009a,b):

- Capturing system wide behaviours with activity diagrams: activity diagrams can be an appropriate medium through which to model the high level behaviour of the biological system that we intend to capture from the interactions of the low level components. The semantics of the diagram also allow for the expression of concurrent activities; which is an intrinsic quality of biological systems.

- Representing static relationship with class diagrams: the construction of class diagrams to be effective at generating questions relating to the entities that may partake in an activity in a static manner, which is not as informative as the systems that need to be modelled from a dynamic viewpoint of the biological systems.

- Sequence diagrams are misleading: the syntax of sequence diagrams does not communicate well, and rather implies that an entity be temporarily suspended whilst

activities still proceed elsewhere in the environment.

- Low level dynamics and state machine diagrams: the ability of the diagram to depict low level dynamics is shown to be very informative; the ability to express orthogonality, concurrency, mutual exclusion, and containment of states renders them appropriate for expressing behaviour of cells in biological systems.

- Depicting feedback with the UML: there are aspects of the biological system that cannot satisfactorily express with UML diagrams. Activity diagrams can demonstrate the order of the critical interactions and events that must take place for a high level behaviour of the systems to manifest. In reality the entity responsible for a preceding activity does not stop but continues and can potentially perform the same activity again.

In addition to the UML diagram, there are other techniques used in software engineering, which Bersini (2006) suggests can facilitate the development and communication of immune modelling. These include object oriented technologies such as object oriented programming and design patterns (Gamma et al., 1994). The perceived benefit is the clarification of immune objects and their relationships. To support this, Bersini (2006) provides an example of how clonal selection can be modelled with a simple state diagram. Design patterns are also proposed by Babaoglu et al. (2006) as an alternative route to exploiting biology for the benefit of computing techniques. They have succeed in identifying a number of suitable patterns common in biological systems, such as diffusion, replication, stigmergy and chemotaxis, that can be applied to distributed computing problems (Babaoglu et al., 2006). They suggest that design patterns can be extracted and abstracted from biology to transfer knowledge to the field of distributed computing that will form a bridge between the engineering and biological systems.

3.4.2 Agent-based Simulation for Biological System

An agent-based simulation (ABS) is a class of computational model for simulating the actions and interactions of autonomous agents (individual or collective entities in an organisations or groups) with a view to assessing the effects of the agents on the whole system (Gilbert, 2008). An interesting observation in ABS, is that by using only simple agents that interact locally with simple rules of behaviour and limited actions to respond to the environmental cues, we can observe the result of their behaviour which leads to a higher-level behaviour of the system than those of each agent (Gilbert, 2008). Agents in ABS share the ability to adapt and modify their behaviour in accordance with their

environment, though they only have discrete, diverse and heterogeneous entities (Gilbert, 2008).

ABS has shown great potential in simulating biological systems, as it offers a tool for biological modelling that is easy to implement and understand. Forrest & Beauchemin (2007) highlight the advantages of using ABS such as:

1. the agent behaviour directly incorporates biological knowledge or hypotheses about low-level components, even if they cannot be expressed mathematically.

2. data from multiple experiments can be combined into single simulation.

3. the immune system is a complex biological system with many interacting mechanisms and many biological relevant values cannot be measured directly.

4. in ABS it is relatively easy to disable the mechanisms altogether adjust their relative contributions and perform sensitivity testing of parameters.

5. there are important spatial and temporal interactions easily studied in ABS for example para cellular signalling between infected and uninfected cells.

A review of modelling approaches in immunology, that focuses on ABS as a tool in simulating a cell as an individual agents has been provided by Forrest & Beauchemin (2007). The authors argue that within ABS, it is possible to observe quite easily the dynamics of the agent population in the immune systems that arise as a result of the interactions between the agents in the simulation, as this will clearly affect how the simulation operates. Forrest & Beauchemin (2007) also argue that ABS might be an appropriate tool for modelling immunology due to the ease of incorporating the knowledge into the model, which may not be able to expresses mathematically and multiple experiments can be run easily. This concurs with the view of Bersini (2006) who advocates the use of object-oriented techniques like UML and design patters in modelling and simulating the biological systems. However, as described in Timmis et al. (2008a) one difficult aspect of ABS is in defining the right level of abstraction for each agent in the simulation as it will clearly gives an affect on how the simulation will operate.

ABS has been used in modelling biological systems as well as been used in engineering. Some of the work that use ABS in biological systems have been done by Auska et al. (2006); Mansury et al. (2002); Segovia-Juarez et al. (2004) and d'Inverno & Prophet (2005). Auska et al. (2006) developed an ABS simulation that simulates real-time signalling induced in osteocytic networks by mechanical stimuli. The modelled cellular functions and interactions between cellular functions resulted in distinct real-time signalling responses when osteocytic networks were subject to cyclical and rest-inserted

loading. Meanwhile, Mansury et al. (2002) proposed a novel ABS tool of spatio-temporal search and agglomeration, designed to investigate the dynamics of cell motility and aggregation by assuming that tumors behave as complex dynamic self-organising biosystems. Rather than simulating cells to obey fixed instructions imposed upon them externally, a new, entirely different, approach is attempted, by introducing non-deterministic stochastic elements in the behaviour of tumor cells and allowing for sequential observation of the spatio-temporal progression of brain tumors in a space and time discrete model. With the aim of understanding the dynamical relationship between the main tumor and its satellites as well as the tumor system itself, the dynamic ABS has proved their hypothesis, showed that the spatio-temporal dynamics of the evolving tumor system is distinctly influenced by the cluster patterns. Another work in ABS done by d'Inverno & Prophet (2005) that implement ABS for predicting the social behaviour of cells in the epithelial tissue. They demonstrate how ABS can be applied to biology and to actually understand important clinical problems using this tool. The results of this static rule-based model, where cells are considered as autonomous agents executing a set of rules depending to their immediate environment, their position in the cell cycle or the differentiation state, in comparison with the in vitro systems examined, suggest that even a model based on simple rules like this can successfully reproduce the behaviour of a real biological system.

3.4.3 Engineering-Informed Modelling Approach

As opposed to section 3.4.1 and 3.4.2 that model and simulate biological systems for the biologists, in modelling the biological systems that serves as an inspiration for AIS, Hart & Davoudani (2011) proposed that the development of the models to be driven by the engineering problem. They highlighted that the constraints of the engineered systems must be informed during the model development phase as well as the validation phase. This is presented as a methodology in the development of abstract model of dendritic-cell trafficking as an inspiration of the development of self-organsing wireless sensor network for temperature monitoring and maintenance. The methodology enables the development of the development of ABS which is consistent with the application constraint and can be validated in terms of functional requirements of the application rather than towards the biological needs (Hart & Davoudani, 2011).

Hart & Davoudani (2011) again emphasised that with engineering functionality and constraint that has been clearly identified, the abstract model of the biology can be constructed at an appropriate level and translated into computational model such as ABS which is consistent with the application constraint. Hart & Davoudani (2011) further discussed on how the model need to be validated as it is no longer appropriate to calibrate it against experimental immunological data as it is developed in accordance to the engi-

neering constraint. They proposed that the validation process to be shifted to validating functionality with the ability of the engineering system to replicate experimental results. The validation process is no longer to validate the simulation with the immunological data but to achieve similar functionality in both biological and the engineering systems (Hart & Davoudani, 2011). As mentioned in Hart & Davoudani (2011) although the result models are not likely what is needed and seen in biology, the models however enable the engineer to better exploit the underlying metaphor that lead to the reduced development time of the engineered systems. Quoted from Hart & Davoudani (2011), although the process lead to the development of a biological unfaithful model it generally helps in the development of engineered systems, in particular:

1. it provides a mechanism for defining the correct level of abstraction of the biological system such that high-level properties of interest are conserved using minimal complexity.

2. it is able to illuminate aspects of the model which cannot be transferred to an engineered system for practical reasons.

3. it provides an environment in which modifications can be made to the model in light of engineering constraints, and tested to ensure that the model retains similar emergent properties.

4. it facilitates the study of the computational aspects of the engineered model in an environment free from complex engineering constraints.

In accordance to the thesis that propose an immune-inspired self-healing swarm robotic systems for anchoring issues, which we described in section 2.2.1, we study the biological systems that have the properties with the similar functions with our constraint and issues. The biological systems that we study is the formation of granuloma that we discuss in section 3.5.

3.5 Granuloma Formation as an Inspirations for Swarm Robotic Systems

There are several reasons why we are interested in taking the inspirations from the process of granuloma formation in immune system for swarm robotic systems. The main reason is the interactions and communications between cells in granuloma formation that are complex and interesting to be studied. Granuloma formation also has several functions in the immune systems. The functions of granuloma formation as described by Adams (1976) are:

1. to get rid of the host of unwanted substances
2. effective in destroying pathogens
3. operative in inducing immunity (injection of an antigen into areas of granulomatous inflammation increases the immune response to that antigen; involves the stimulation of macrophages as well as T and B lymphocytes)
4. may take part in the destruction of neoplasms (a new and abnormal growth of tissue in some part of the body)

In Sandor et al. (2003), the authors also highlighted that granulomas are excellent models to study local immuno-regulation and the effector functions of immunity. Granuloma formation is important for the host because the absence of granulomas greatly increases the lethality of infections Sandor et al. (2004). In humans, immunodeficiency (failure of the immune system to protect the body adequately from infection, due to the absence or insufficiency of some component process or substance) that restrict granuloma formation ultimately lead to uncontrolled growth and dissemination of bacteria. This motivates us in understanding granuloma formation and instantiate the ideas into self-healing swarm robotic systems.

To enhance our discussion, in this section we introduce the concept of granuloma formation and explain the properties of granuloma formation that are relevant to the properties of swarm robotic systems specifically for self-healing mechanism. We begin by discussing the meaning of granuloma in section 3.5.1. We then continue our discussion on granuloma formation by describing the cells involved during the formation such as macrophages and different signalling mechanisms and the description of the development of granuloma formation in section 3.5.2. We end this section by explaining the properties of granuloma formation that are relevant to swarm robotic systems particularly in the issues of self-healing in section 3.5.3.

3.5.1 Biological Background of Granuloma Formation

A granuloma formation can be defined as (Adams, 1976):

> *a compact (organised) collection of mature mononuclear phagocytes (macrophages and/or epitheliod cells) which may or may not accompanied by accessory features such as necrosis of the infiltration of other inflammatory leukocytes*

From this definition, a simplified version of this terminology is given by Adams (1976) as an *organised collection of macrophages*. Adams (1976) explained that granulomas evolve conceptually in three (3) stages:

1. the development of an infiltrate of young mononuclear phagocytes.

2. the maturation and aggregation of these cells (phagocytes) into a mature granuloma.

3. the further maturation of mature granuloma into epithelioid granuloma.

This conceptual stages of granuloma formation quoted from Adams (1976) are further elaborated by Facco et al. (2007) as the following processes:

1. the triggering of T-cells by antigen presenting cells, represented by macrophages and dendritic cells;

2. the release of cytokines and chemokines by macrophages, activated by lymphocytes, dendritic cells and other cells;

3. the stable and dynamic accumulation of immunocompetent cells and the formation of the organised structure of granuloma;

4. the last phase of granuloma formation generally ends in fibrosis (the thickening and scarring of connective tissue, usually as a result of injury).

Granuloma formation is a complex process involving a variety of mechanisms acting in concert to bring an inflammatory lesion that is able to contain and destroy intracellular pathogens. While it is crucial to host defence, inappropriate granulomatous inflammation can also be considered as damage to host defence. The main actors in granuloma formation are; macrophages, t-cells and cytokines. By means of example, we prepare a simple illustration of granuloma formation in figure 3.4. Based on this figure, granuloma formation begins when infectious diseases are brought in by bacteria. Macrophages 'eat' or 'engulf' bacteria to prevent it from spreading; however, the bacteria infects macrophages and duplicates themselves. Thus, despite the macrophages able to stop the infections, bacteria uses macrophages as a 'taxi' to spread diseases leading to the cell lysis or breaking down the structure of the cell. Infected macrophages then emit signal indirecting that they are infected and this signal leads other macrophages to move to the site of infection, to form a 'wall' around the infected macrophages to isolate the infected cells from the uninfected cells. This finally lead to the formation of a granuloma that represents a chronic inflammatory response initiated by various infectious and non-infectious agents. The centre of a granuloma consists of infected macrophages, which can become necrotic.

3.5.2 Cells and Development of Granuloma Formation

The pathogenesis or the development of granulomatous inflammation is complex and involves a variety of mechanisms acting in concert to bring about an inflammatory lesion

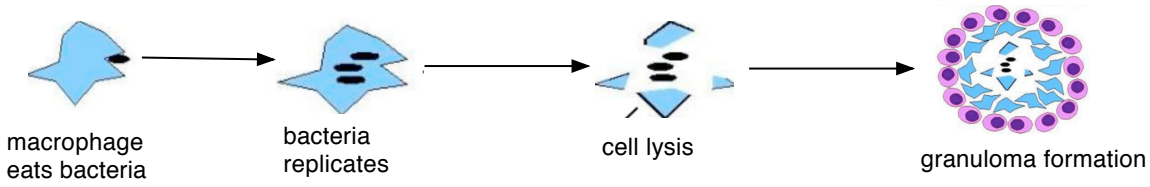


Figure 3.4: Simple ideas of granuloma formation

that is able to contain and destroy intracellular pathogens. A better understanding of these events will allow us to more precisely modulate the granulomatous inflammatory response. Figure 3.5 shows the morphology of granuloma formation summarised based on descriptions given by Adams (1976), which is based on the explanation in Adams (1974). Referring to figure 3.5, we tried to give basic explanation on the morphology of granuloma formation, further to this discussion, we pointed out the original sources (Adams, 1974, 1976) for further revisions. Granuloma formation started with an injury to human body. The next event that occur once the injury is not cured in the acute inflammation. After several days, the young mononuclear phagocytes developed into immature macrophages resulting the chronic inflammation to occur. By 3 to 7 days, reactions will occurs (mostly comprises of the mature macrophages plus foreign body of giant cells (Adams, 1976), that will compactly aggregated into organised nests and sheets that will develop the mature granuloma. Certain stimuli and reactions will induce further development of mature granuloma. The macrophages finally enlarged and formed the epitheliod granulomas.

According to the definition given in section 3.5.1, the main cells that define granuloma formation are macrophages. The other key term in the above definition is the word *organised*, which refers to a tight, ball-like formation. The macrophages in these formations are typically so tightly clustered that the borders of individual cells are difficult to identify. Granulomas may contain additional cells that act as a clue to the cause of granuloma or other diseases. Furthermore, the antigen causing the formation of a granuloma is most often an infectious pathogen or a substance foreign to the body, but often the offending antigen is unknown (as in autoimmune disorders).

Other than macrophages, cytokines serve as crucial signal transmitters between cells in granulomatous lesions, and are required for the recruitment of lymphoid cells and efficient activation of macrophages. Although up-regulation of a number of cytokines is seen in granulomatous inflammation, available evidence suggests that the TH-1 cytokines IFN-γ, IL-12, and TNF are required for normal granuloma formation and maintenance. In models of infectious granulomatous inflammation, IFN-γ is produced early in the in-

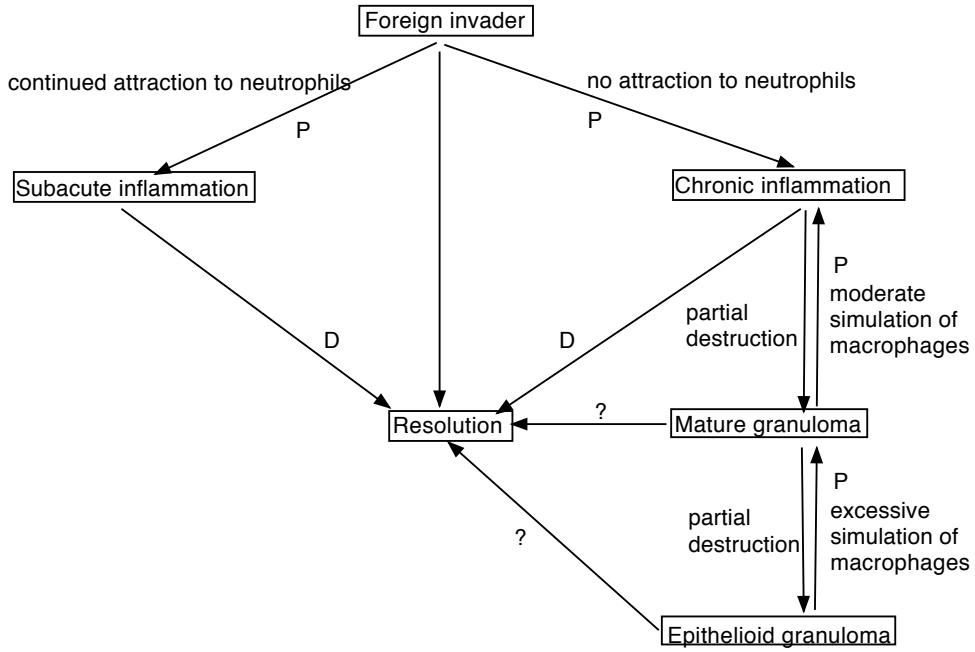


Figure 3.5: Conceptual model of granulomatous inflammation. Persistence of the foreign invader (P) and its destruction (D) are denoted, (?) is the unknown invaders (Adams, 1976)

fection by NK cells and later by T-cells (Sarraf & Sneller, 2005). IFN-γ has a number of effector functions relevant to granuloma formation, including activation of macrophage bactericidal mechanisms, induction of TNF secretion by macrophages, activation of the endothelium to promote T-cell adhesion, and promotion of T-cell differentiation. TNF may function to limit the influx of neutrophils that cause tissue damage, while promoting the recruitment and migration of T-cells into granulomas where they can interact with macrophages (Sarraf & Sneller, 2005).

During the process of granuloma formation, T-cells leave the lymph node and migrate to the focus of infection where they secrete soluble mediators that play a central role in initiating and sustaining granuloma formation (Sarraf & Sneller, 2005). The dendritic cells produce interleukin 12 (IL-12) and present antigen to naïve CD4+ T-cells. Under the influence of IL-12, naïve CD4+ T-cells differentiate into T helper 1 (Th1) cells. Activated Th1 cells secrete IL-2, which promotes T-cells survival and proliferation, leading to expansion of the population of antigen-specific Th1 cells.

Although the identification of critical determinants of granuloma formation and evolution remains elusive, (Sarraf & Sneller, 2005) identify some of the cells that involved in granuloma formation. This is represented in Figure 3.6 that explains on the development and interaction of cells in granuloma formation. Based on Figure 3.6, the interactions of

cells starts when it is exposed to antigen. Within seconds or minutes after the exposure to antigen, resident cells initiate cellular recruitment. Pre-stored tumor necrosis factor (TNF) released by macrophages recruits neutrophils, which turn signal to and circulate monocytes that will lead to the granulomatous inflammation. Interferon γ (IFN- γ) produced by local natural killer (NK) and T-cells further activates resident tissues histiocytes and dendritic cells that also release lots of chemokines and TNF, that alter the local microcirculatory environment and facilitate cellular trafficking into tissues.

Following the accumulation (mass gathering) and activation of macrophages, the inflammatory lesion (region that has suffered damaged) begins to take on a granulomatous form. With the arrival of antigen-specific T-cells, the lesion transforms into a mature granuloma where activation of macrophages by IFN-γ) and tumor necrosis factor (TNF) results in inhibition (slowing or prevention) of microbial growth. Eventually, the granuloma becomes encapsulated by a fibrotic rim and, in the case of infections, the centre becomes necrotic (death). These tissue reactions function to protect the host by promoting microbial containment (under control) and reducing the nutrient supply to the pathogen (Sarraf & Sneller, 2005).

3.5.3 Mapping from Granuloma Formation to Swarm of Robots

As described in chapter 2, self-organisation models of social insects and animals have already been used as inspiration sources for many swarm robotics studies. Here, we would like to draw attention a line of research, which we believe, contains ideas that can act as inspiration that, we consider, most relevant and inspiring for swarm robotics research. Other then studying social insect, studies of biological systems can also act as source of inspiration for swarm robotics. This is because, studying biological systems involve the study of self-organisation, which is defined by Camazine et al. (2001) as :

> *'a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. The rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern.'*

Self-organisation, or decentralised control, is widespread in biological systems, including cells, organisms, and groups that possesses a large number of subunits, and these subunits lack either the communicational abilities or the computational abilities, or both, that are needed to implement centralised control (Camazine et al., 2001). The study of self-organisation of biological systems (such as granuloma formation) has been is another potential Seeley (2002). Camazine et al. (2001) have revealed that there is no centralised co-ordination mechanisms behind the synchronised operation of biological systems, yet

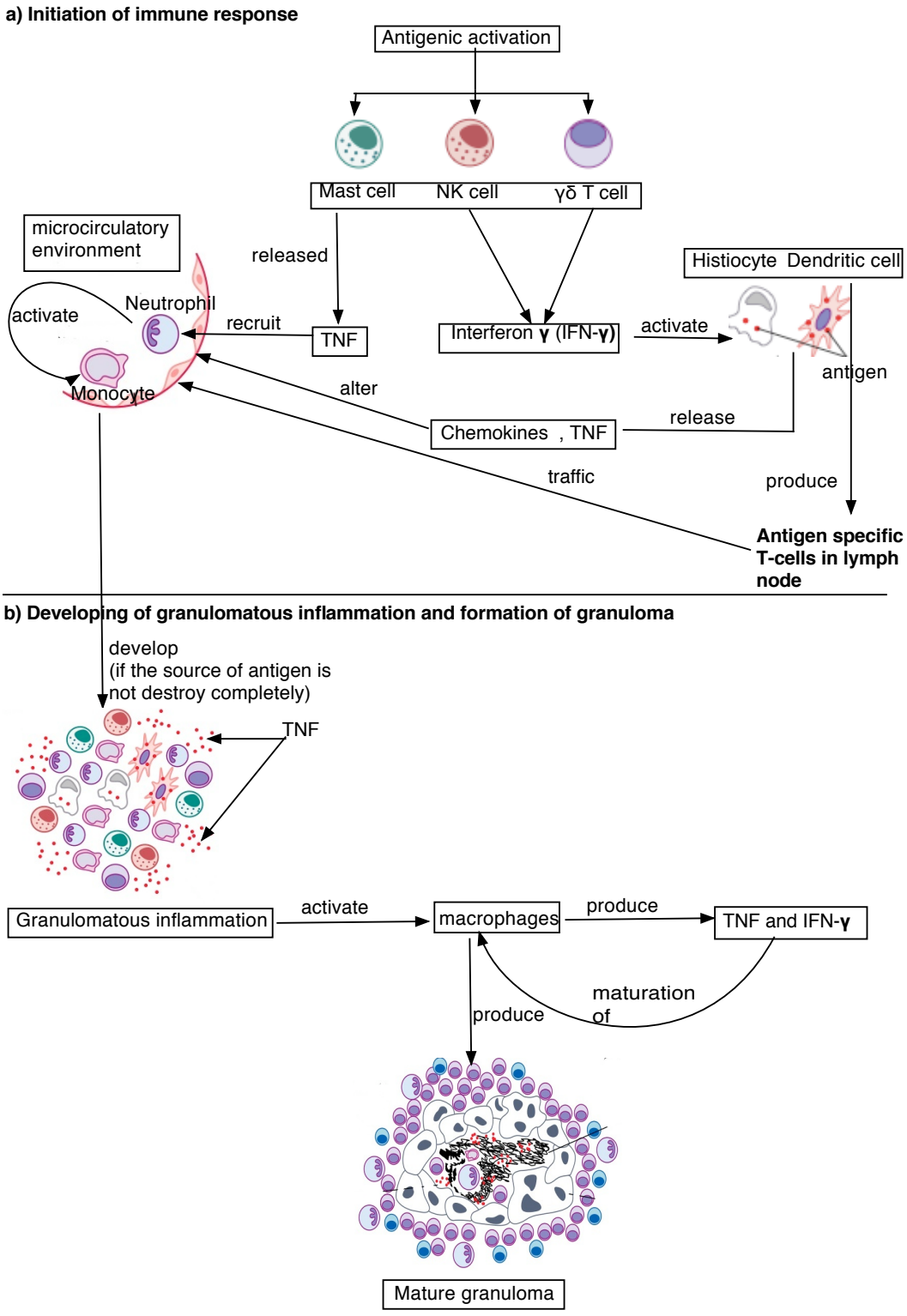


Figure 3.6: Morphology of various elements of mononuclear phagocyte system (Upon initiation of immune response to antigen). Detailed definitions of the cells are found in (Sarraf & Sneller, 2005)

the system level operating with robust, flexible and scalable manner. Several areas of computer science have adopted the idea that swarms can solve complex problems and some of them are described in Bonabeau et al. (1999) for example combinatorial optimisation and routing communications network. It is also used in solving robotics applications (Beni, 2005). Since study of self-organisation models of social insects and animals have been used as inspiration in swarm robotics studies, studying the self-organisation models of granuloma formation may be suitable as another inspiration for swarm robotics studies. This is in line with the suggestion given by Şahin (2005), that suggest new source of inspiration such as communication and information exchange in in other system such as bacterial system that can improve the robustness of the swarm robotics system.

This thesis therefore is attempted to uncover the principles behind the emergence of self-organisation in granuloma formation, by developing models that are built with simplified interactions between the cells and behavioural mechanisms in each cells. Studies on granuloma formation as the subset of studying biological systems as the whole will allow us to see how the cells in the systems act as a pattern analyser as well as a system that is capable on solving problems in their best way, to keep the host always healthy. Therefore, looking from the other perspectives such as granuloma formation, it will allow us to study the concept of swarm not only associated with systems capable of carrying out not just useful tasks but also intelligent tasks (Şahin, 2005).

In the case of granuloma formation described in section 3.5.1, the term swarm refers to a large group of cells of immune system that interact during the formation, working together to achieve a goal and produce significant results, that is protecting the host through out the formation. In granuloma formation, cells that act as the control mechanisms try to isolates the pathogen and promotes the development of protective immunity by allowing cross-talk between T-cells and macrophages. Furthermore, granuloma formation involves the movement and interactions of various cell types, including bacteria,macrophages, T cells, cytokines, and chemokine. By having a model to simulate granuloma formation under various conditions, we can identify key control mechanism associated with successful granuloma formation. We also can explore what conditions contribute to the breakdown of a granuloma, corresponding to reactivation of latent infection. We can also study on how cells in granuloma formation enhanced their communication and robustness through chemokines and cytokines signalling.

Taking into account our initial aim of developing an immune-inspired algorithm based on novel immunology, we have identified the the patterns between the process of granuloma formation as well as swarm robotics. During the formation of granuloma, it is interesting to note in the case of 'anchoring' issue in swarm beacon taxis that has been previously described in section 2.5. We propose that there is a natural analogy between

the potential repair of a swarm of robots, as in the situation of swarm beacon taxis, and the formation of a granuloma and removal of bacterial infections to the cells. This is summarised in table 3.1.

Table 3.1: Properties of swarm robotics and granuloma formation

Properties of swarm robotics	Properties of granuloma formation
Large number of robots	Large number of cells
Few homogeneous groups of robots	Few homogeneous cells
Relatively incapable or inefficient robots	Each cell needs to perform the desired task
Robots with local sensing and communication capabilities	Chemokines and cytokines

As mentioned in chapter 2, accurate self-healing needs to involve more complicated computation, analysis and decision processes. Such analysis needs system-level models and makes decisions in a holistic manner. Suspect events lead to the analysis, including the abnormal cases that are detected by the virtual neurons. The self-healing needs to implement two important functions: autonomous diagnosis and autonomous repair. One of the strongest and most important reasons for applying swarms is the potential for the system to be robust. The idea is that a swarm with many robots working together will continue to operate even if some of the robots fail. The technical term for this is reliability. One can think of a swarm as a massive parallel system with a high degree of redundancy. Thus, in general, we might be able to use the granuloma idea in swarm beacon taxis issue as follows:

- If a robot fails, this becomes like a macrophage and begins to emit signals.
- Other robots now become 'T-cells' and are attracted to the failed robot.
- They surround the robot (form a granuloma) to isolate the robot from the rest of the swarm.
- The rest of the swarm is no longer effected by the failing robot(s).

3.6 Conclusions

In this chapter, we presented the applied Artificial Immune Systems (AIS) by providing the general overview of the AIS in section 3.1. We also described the AIS algorithms, which are categorised as: non-swarm immune-inspired algorithms and swarm-like immune-inspired algorithms in this section. We then discussed on the immune algorithms for swarm robotic systems in section 3.2. We then described the suitable design

methodology: the conceptual framework, CoSMoS process and immuno-engineering in section 3.3 that act as guidelines during the development of our AIS in chapter 4. We also described the work on modelling and simulation of immune systems, where we gave some examples of modelling and simulation techniques in biological systems such as diagrammatic and agent-based simulation approaches in section 3.4. In addition to this, we also described the engineering-informed modelling approach that is discussed in Hart & Davoudani (2011) as our guidelines in developing the immune-inspired self-healing swarm robotic systems. In accordance to our engineering problems and constraint we studied the process of granuloma formation as the inspiration to our AIS in section 3.5. Studies on granuloma formation as a subset of studying immune systems as a whole, will allow us to understand how the cells in the systems react against many types of viruses to keep the host continually healthy. Therefore, looking from the other perspectives such as granuloma formation, will allow us to study the concept of swarm not only associated with systems capable of carrying out just useful tasks but also intelligent tasks. Having explained the problem domain in section 2.5, which we are working on, and described the process of granuloma formation in section 3.5, we now are able to model and simulate the process of granuloma formation and take that forward towards the development of an immune-inspired algorithm for swarm robotic systems, which will be described in chapter 4. We end this chapter in section 3.6 where we discussed the importance of immune-inspired solution in engineering swarm robotic systems.

CHAPTER 4

A Model and Simulation of Granuloma Formation

The preceding chapter identified our intention to model and simulate the process of granuloma formation to establish design principles as inspiration for the development of novel Artificial Immune Systems (AIS). Within the context of the conceptual framework approach (CFA) of Stepney et al. (2005), our next step, which will be discussed in this chapter is to build a simplifying abstract representation in Unified Modelling Language (UML) of the model and agent-based simulation, that will capture the properties of granuloma formation. We first identify the research context in section 4.1, which is based on our previous discussion on granuloma formation in section 3.5, where we highlight the scope, purpose and the goal of the modelling and simulation work. In establishing our model and simulation, in accordance with the CoSMoS process (Andrews et al., 2010), we prepare our domain model in section 4.2 to encapsulate our understanding of appropriate aspects of the domain (granuloma formation) into explicit domain understanding, where we define the boundary of our model, and we prepare UML diagrams that model the behaviour and the regulation of granuloma formation. Next, in section 4.3, we detail the implementation behaviour and interactions of the agents and the environment from the domain model, followed by presenting the agent-based model of granuloma formation in section 4.4. We then perform sets of experiments based on the agent-based simulation and present the results of the experiments in section 4.5. Within the context of our goal to develop a novel AIS following the CFA and CoSMoS process described in section 3.3, we end this chapter with a conclusion in section 4.7, which will work as the basis of the work presented in chapter 5.

4.1 Research Context

In section 3.3, we described approaches to the development of immune algorithms: the conceptual framework (Stepney et al., 2005) and CoSMoS process (Andrews et al., 2010). Both approaches emphasise the needs of modelling and simulating the immune systems to assist understanding before the development of immune inspired algorithms. The conceptual framework is based on the following processes: 1)probes, observation and experiments, 2)abstract representation, based on a model of biological systems. 3)analytical computational frameworks, 4)the bio-inspired algorithms. However in CFA it does not really describe the modelling stages rather only highlight the needs and the benefits that we can obtain from these stage. Therefore, in exploring the modelling and simulation process in the conceptual framework (Stepney et al., 2005), Andrews et al. (2010) proposed the CoSMoS process and identified five distinct products that need to be accomplished during the modelling and simulation process. They are: 1)research context, 2.)domain model, 3)platform model, 4)simulation platform and 5)results model. Having described the five products in detail, the CoSMoS process (Andrews et al., 2010) is able to distinguish what do we want to achieve for each modelling and simulation stages.

Following the CoSMoS process Andrews et al. (2010), we developed a model and simulation of the general formation and progression of granuloma formation, rather than in the case of a specific disease. This is due to the fact that we did not wish to model the formation to provide insight from a biological perspective, but understand the dynamics of a general model to allow us to distill a series of design principles that we can use to create a novel AIS algorithm for swarm robotic systems. In accordance with the CoSMoS process (Andrews et al., 2010), the research context defines the fundamental purpose and goal of a project. Thus, the goal of this chapter is:

> *to develop a UML model and a simplified agent-based simulation to assist understanding of the interactions of cells during the development of granuloma formation that can be incorporated into a novel AIS that can be applied to the issues of fault tolerance in swarm robotic systems.*

This chapter is therefore concerned with the development of an abstract model and simulation that can be used to aid the design of our novel AIS. In achieving the purpose, we have identified in our literature survey in section 3.5, during the development of granuloma formation there exist lots of interaction between cells; (for examples: macrophages and T-cells) in the immune system. Before we can proceed to developing AIS, we seek to investigate how the development of granuloma formation will emerge, based on the interactions and signals from different types of cells. Thus, we develop a model to understand the properties of granuloma formation and transform the model to a simulation based on

the interactions of agents and different types of signalling mechanism. The model and simulation presented in this chapter is therefore exploratory in the context of elaborating the ideas of interaction between cells, with signalling mechanisms leading to the formation of granuloma, which is based on the literature survey done in section 3.5. To achieve our goal, we need to:

- investigate immunology background to identify cells and signalling properties that emerge in granuloma formation (described in section 3.5).
- prepare the domain model consisting of a set of UML diagrams based on the outcome of this investigation.
- prepare the platform model to focus on the 'how' aspects of the domain model.
- prepare the simulation platform, upon which the simulation of granuloma formation can be performed.
- investigate any behaviours that emerge that can be identified as being the response of cells in the formation of granuloma based on the simulation.
- analyse the behaviours that we observe with respect to our thesis goal of the development of novel AIS inspired by granuloma formation in accordance with CFA.

4.2 Domain Model

As indicated by the CoSMoS process (Andrews et al., 2010), we start by noting down the behaviours of the system in which we are interested. We therefore divide the behaviour of granuloma formation into main stages and describe each stage based on our literature survey. They are useful because we then know which process needs to be focused on our model and simulation. The stages are:

1. initiation of immune response: within seconds to minutes after exposure to antigen, resident cells initiate cellular recruitment. Pre-stored tumour necrosis factor (TNF) released by mast cells recruit neutrophils, which in turn signal to and activate tissue and circulating monocytes. Interferon-gamma (IFN-g) produced by local natural killer (NK) and T-cells further activates resident tissue histiocytes and dendritic cells. These latter cells release a host of chemokines and TNF.
2. formation of granuloma: 1) production of antigen specific T-cells; antigen-loaded dendritic cells travel to local lymph nodes and initiate a lymphocytic response. Dendritic cells produce interleukin-12 (IL-12) and present antigen to naive T-cells. Under the influence of IL-12, naive T-cells differentiate into T-helper-1 (Th) cells.

Activated Th-cells secrete IL-2, which promotes T-cell survival and proliferation, leading to expansion of the population of antigen-specific Th-cells. 2) within hours to days after antigen exposure, activated Th-cells preferentially traffic to sites where the microcirculation has been altered by TNF and chemokines produced by resident cells. If the source of antigen is not eradicated, inflammation persists. The interaction between Th-cells and activated macrophages leads to the production of IFN-g and TNF, which results in the further maturation of macrophages. Over the course of several days to weeks, a mature granuloma is formed. Other cells, including but not restricted to neutrophils and B-cells, are found in various proportions in the mature granuloma.

3. removal of bacteria and infected macrophages: activated macrophages and T-cells are the cells involved in the removal of the bacteria.

From the description given above, we prepare an abstract depiction of cells involved in granuloma formation showing the regulation that counters each cell and the interaction between those cells. We identify that there are two main cells; macrophages and T-cells that are involved during granuloma formation. The abstract depiction of granuloma formation is shown in figure 4.1. From this figure, uninfected macrophages will become infected macrophages if it is infected by bacteria. The infected macrophages will then emit signal *A* to the uninfected macrophages as well as to the dendritic cells that will then activate T-cells, leading to the movement of the uninfected macrophages and T-cells to the site of infection, which contains the infected macrophages. These cells form a 'wall' to isolate the infected macrophages from other cells. Once activated, T-cells then emit signal *B* that activates the infected macrophages. The activated macrophages are capable of killing bacteria and control the bacterial infections in the systems.

Based on our understanding of the properties of granuloma formation, we further delineate the information that we have by identifying the properties that are important for our model and simulation. This is because there exists a huge variety of elements interacting in the biological system and to accurately simulate all of them is impossible (Read et al., 2009a). Thus, based on the development stages of granuloma formation, we denote the observable phenomenon of the real-world domain. This is depicted in figure 4.2, where we define the system we intend to model, both the physical entities within it and the behaviours we expect them to manifest within the systems. Figure 4.2 also explicitly depicts several levels of hypothesis in which will be incorporated in the model and the simulation of the systems. There exists a transition across dotted line depicting our hypothesis concerning the abstract behaviours in which we believe would be responsible for the observable phenomenon. Our investigations are scoped within the context of these

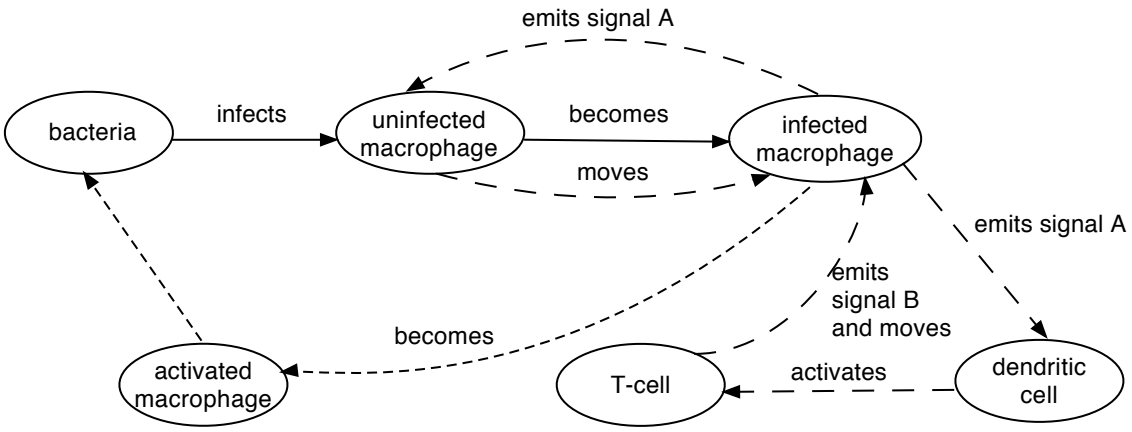


Figure 4.1: An abstract depiction of the cells involved in granuloma formation, the regulation that counters it, and the interactions between them. There are three phases in the formation of granuloma. The solid arrow shows the interactions of cells for the first phase of the formation of granuloma, the initiation of the immune response. The long dashed line shows the second phase of the formation, including the signals and interaction of cells once the macrophages have been infected by the bacteria during the formation of granuloma. The short dashed lines show the final stage of the formation of granuloma showing the removal of bacteria.

expected behaviours. We will not attempt to investigate anything outside the scope that we presented in in figure 4.2. The transitions over the dotted line indicate how this work fits into the wider context of the domain under study. Further hypotheses are detailed in the links between the expected behaviours and the elements involved in the system. These links indicate which elements in the system are responsible for manifesting the expected behaviour, and will thus find explicit representation within our system.

The hypotheses detailed in figure 4.2 provide a link between the expected behaviours of the cells interacting in the systems. The links indicate which cells and their interactions in the system we believe will be responsible for exhibiting the expected behaviours during the process of granuloma formation. In our case, the observed phenomena is a macrophage that is infected by bacteria and attracts other cells (macrophage and T-cell) to the site of infection that will be responsible for the formation of granuloma. This formation happens through the action of the cells and signalling mechanisms in immune systems. This process is further expanded and discussed in other UML diagrams in section 4.2.1. From figure 4.2, we explicitly depicts several levels of hypotheses that the model and simulation will incorporate. We hypothesised that:

- the secretion of cytokines initiated by the bacteria that infects the macrophages will attract other immune cells to the site of infections.
- the recruitment/participation of immune cells (uninfected macrophages and T-cells)

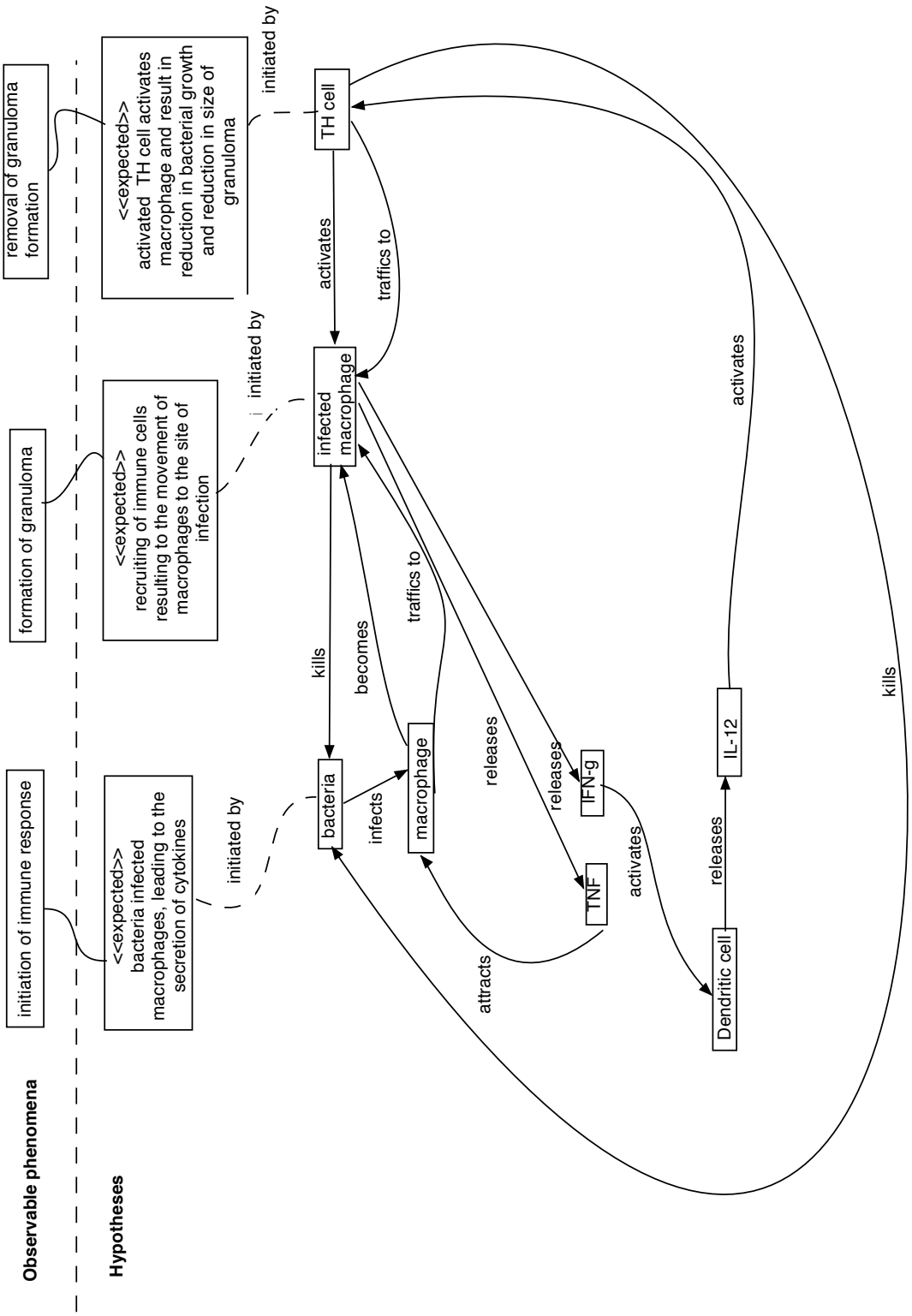


Figure 4.2: The observable phenomenon of the biological domain; the behaviours that we hypothesise to be responsible for those phenomenon at abstract level.

trafficking towards the infected. macrophages is significance for the development of granuloma formation.

- the secretion of cytokines from dendritic cells, which are activated by the infected macrophages will activate T-cells that is liable for the removal of granuloma formation and lethality of the infections.

For our model and simulation, we are only interested in the formation of granuloma; thus, we will be looking more at this stage in greater detail, leaving the initiation and removal of granuloma formation stage unexplored. By delineating our systems, we will be able to understand granuloma formation to be instantiated to an AIS algorithm.

4.2.1 Modelling the Expected Behaviours

As described in section 3.4.1, we use UML diagram as a tool to develop our model before we move to the agent-based simulation. This model serves as a way of representing our understanding of the process of granuloma formation summarising the literature of granuloma formation, where there is no technical specification of the simulation described. The expected behaviours of granuloma formation, as depicted in figure 4.2, represents the behaviour of the system that arises from the low-level interactions of cells in the systems. As mentioned in section 3.4.1, activity diagrams can be a medium to express how the scenario occurs. This is depicted in figure 4.3, which shows that the process of granuloma formation is decomposed into lower level activities as performed by cells of the system. The events depicted in this activity diagram are the abstract concepts of the process of granuloma formation, which does not specify the behavioral dynamics of individual cells. This will be accomplished with the use of state machine diagrams, as discussed in section 4.2.2. As mentioned in section 3.4.1, this diagram can be useful in showing how the individual cell-level dynamics expressed in state machine diagrams integrate to constitute a system dynamic.

4.2.2 Modelling Low-level Dynamics of Cells

In granuloma formation, cells actively change their state depending on their interactions with other cells or signals that they receive from other cells. Therefore, we will use state machine diagrams in depicting the low-level behavioural dynamics of the cells in the systems to capture the change of the state of the cell in the systems. The diagrams do not require in depth textual explanation to be understood, but the general features are explained briefly, based on the literature survey conducted in section 3.5. It is quite

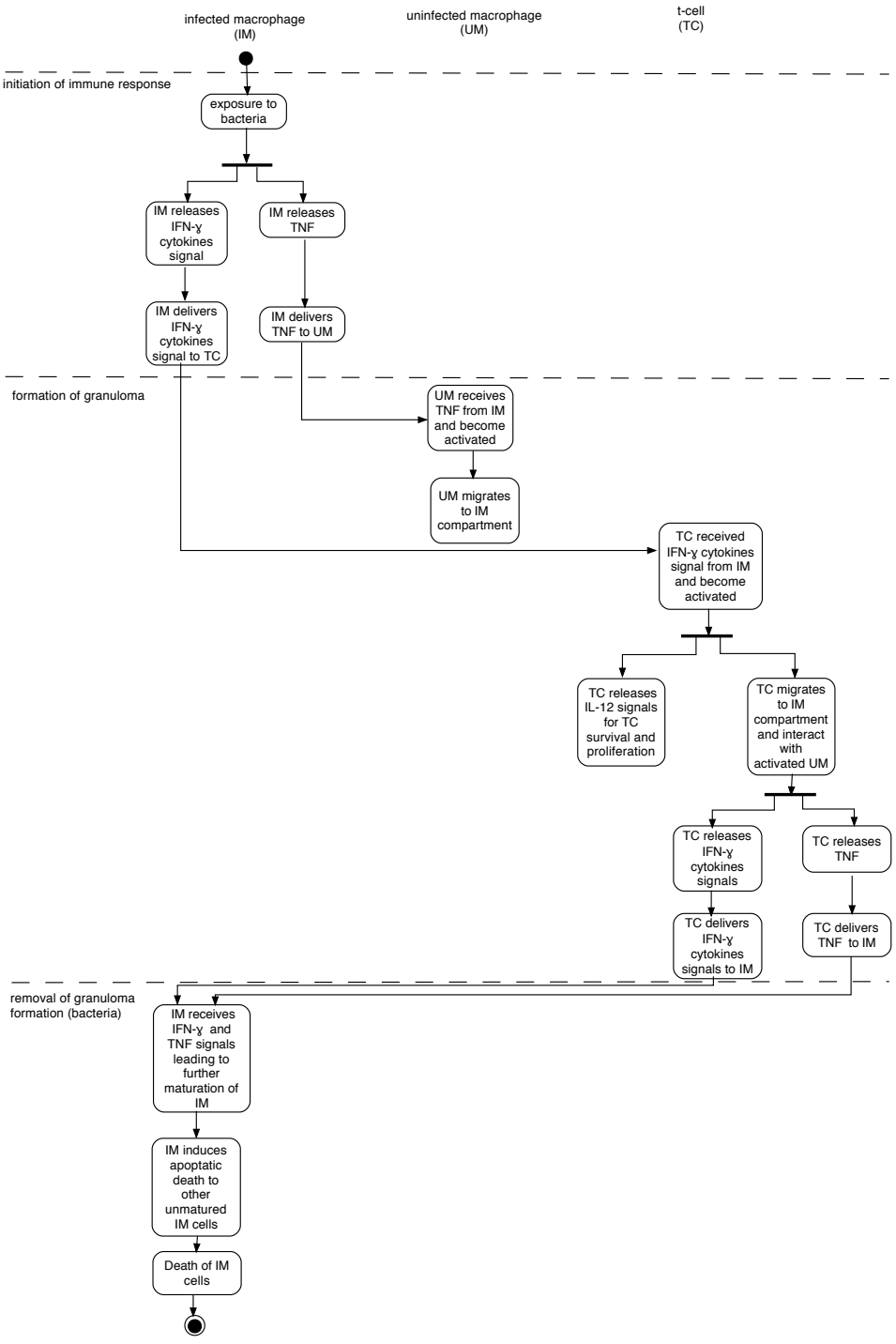


Figure 4.3: Activity diagram depicting the simplified order of events that occur for the instigation of granuloma formation. Different types of signal existed during the process.

challenging to correctly capture the dynamics of biological entities, however having this exercises will ease the understanding of the process of granuloma formation.

Figure 4.4 represents the state machine diagram for a macrophage. In this figure we show the different stages of macrophages: uninfected, infected, chronically infected and activated. Uninfected macrophages take up extra-cellular bacteria from the environment, and if not activated quickly, will become infected. An excessive exposure of a macrophage to extra-cellular bacteria can quickly transform the uninfected macrophage to a chronically infected macrophage. A sufficient bound of an infected macrophage with TNF factor will transform the macrophage from infected to activated. With enough stimulation, an activated macrophage can phagocytose and induce death to the infected cells; however their ability to function properly decreases with increasing intra-cellular bacterial load. In figure 4.4, we also show different types of signals that will be generated and secreted by macrophages. Based on our literature survey, there are two types of signal that will be generated and secreted by activated macrophages; the IL-2 cytokines and TNF factor. These signals are shown in figure 4.4 as a parallel states when the macrophages are activated. Once activated, macrophages generate the IL-2 cytokines and TNF factor and both signals are secreted to activate other macrophages. These signalling mechanisms also help the activated macrophages to continue proliferating and induce the apoptatic death to the infected macrophages, whilst controlling the infected macrophages to further infect other uninfected macrophages and spread the infections in the systems. Macrophage dies when it has reached its aging period. However, in regards to chronically infected macrophages, they can die if the intra-cellular level in their cells are above the specific threshold value. Having the state machine diagram helps to ease our understanding and while reading the literature, we can update the diagram to add other mechanisms and processes that we encounter for a macrophage before committing ourselves in preparing the simulation.

We then establish a state machine diagram of T-cell during the process of granuloma formation. This is depicted in figure 4.5. This figure represents the different stages of T-cell: naive, partially activated and activated during the process of granuloma formation. With sufficient binding with IL-12 cytokines secreted by dendritic cell in the environment, T-cell will become activated. Once activated, T-cell generates IL-2 cytokines which is depicted as a parallel state in figure 4.5 that help the activated T-cell to differentiate and proliferate. Upon sufficient aging period, T-cell will die and will be removed from the environment. With state machine diagram for T-cell, we can demonstrate the low-level behavior of T-cell. Doing this exercise, will lead to questions that need to be answered before proceeding to the simulation development.

We then potray the state machine diagram for different types of signalling mechanism

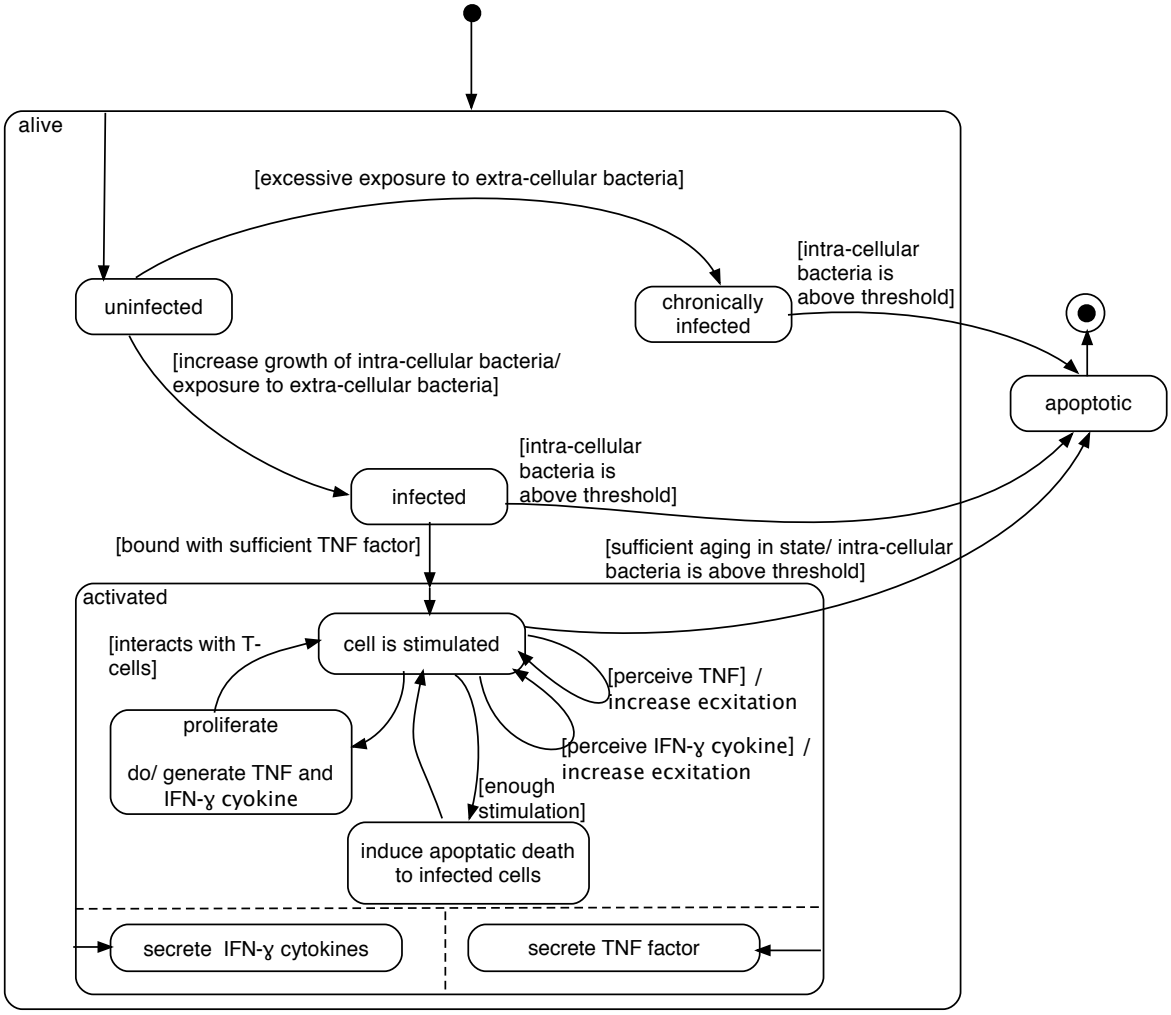


Figure 4.4: State machine diagram of a macrophage during the process of granuloma formation.

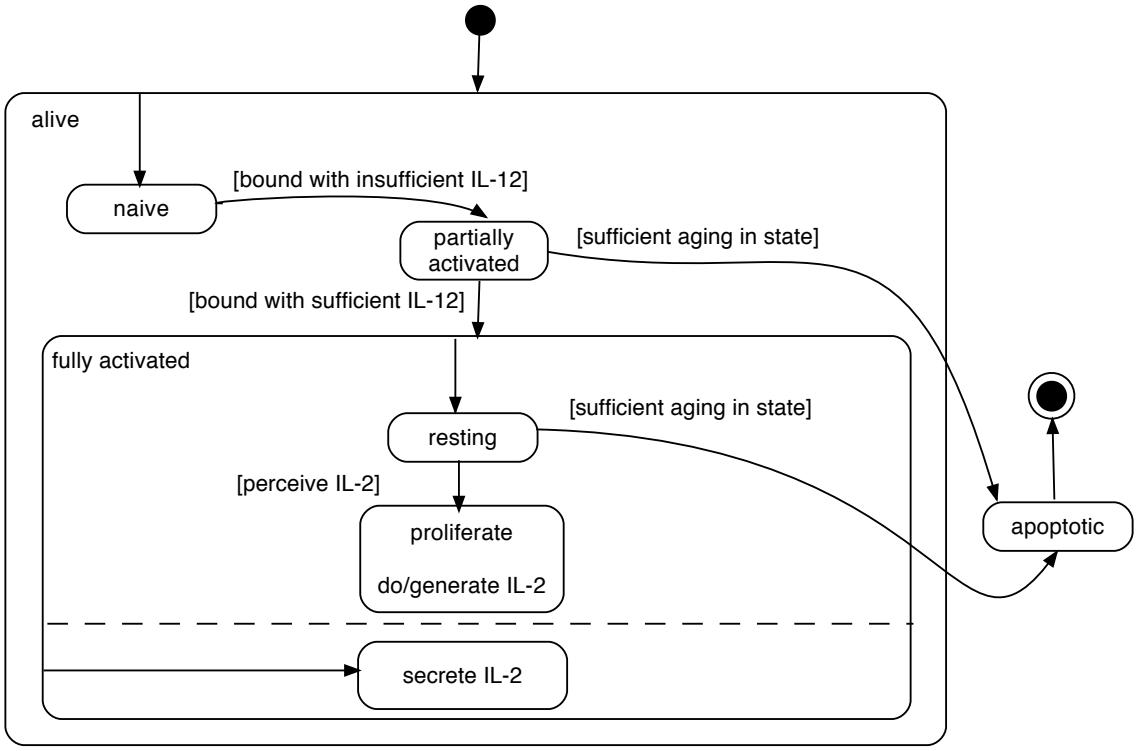


Figure 4.5: State machine diagram of a T-cell during the process of granuloma formation.

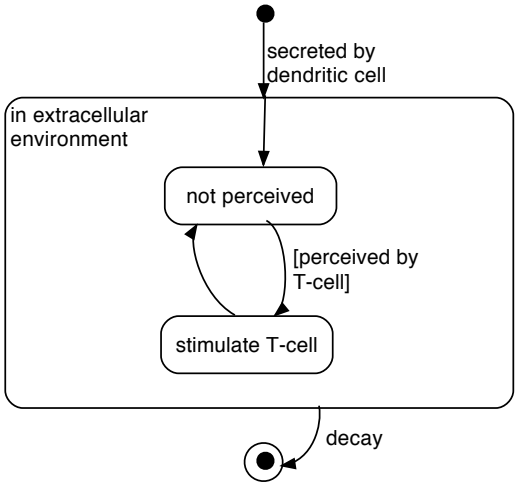


Figure 4.6: State machine diagram of the IL-12 cytokine.

during the process of granuloma formation in figure 4.6, figure 4.7 and figure 4.8. These figures do not describe the different stages of the signals, since they do not change their behavior during the process of granuloma formation. Figure 4.6, figure 4.7 only explain which cells secrete the signals and the effect on the cells that perceive the signal.

Based on the UML model, we summarised that there are four stages in granuloma formation as shown in figure 4.9. The stages are:

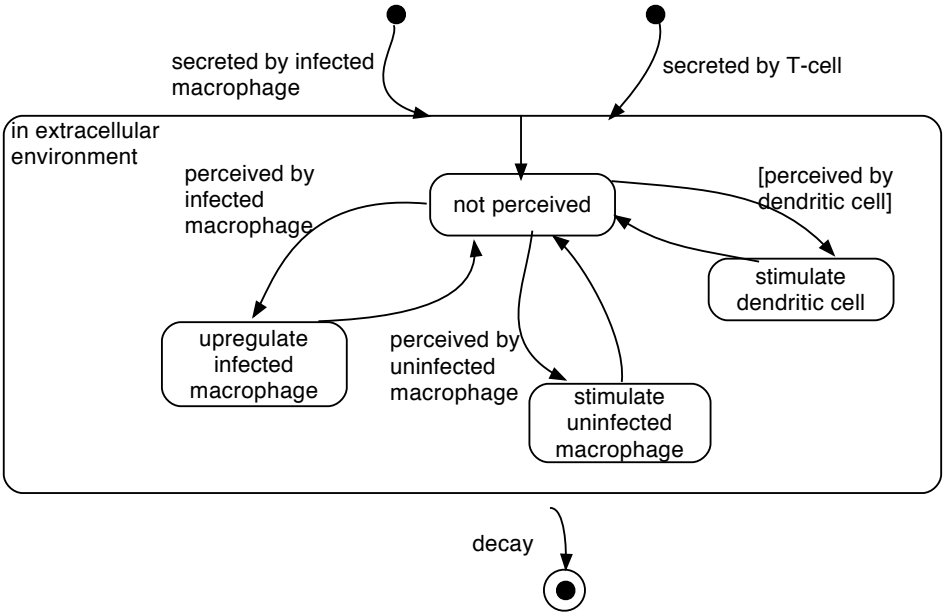


Figure 4.7: State machine diagram of the TNF factor.

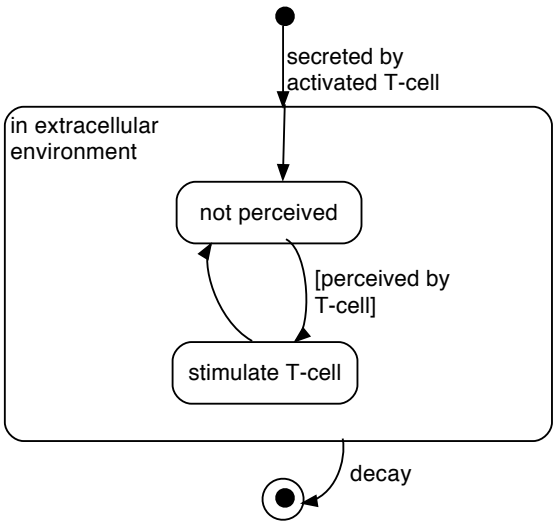


Figure 4.8: State machine diagram of the IL2 cytokine.

1. movement: movement of macrophages in the system;
2. detection: macrophage detects that there is bacterial infection exists in the system and the bacteria then starts to infect the macrophage;
3. signal propagation: the infected macrophage then starts to emit signals to attract the other macrophages;
4. granuloma formation: other macrophages move to the site of infection and try to stop the spread of the bacterial infections.

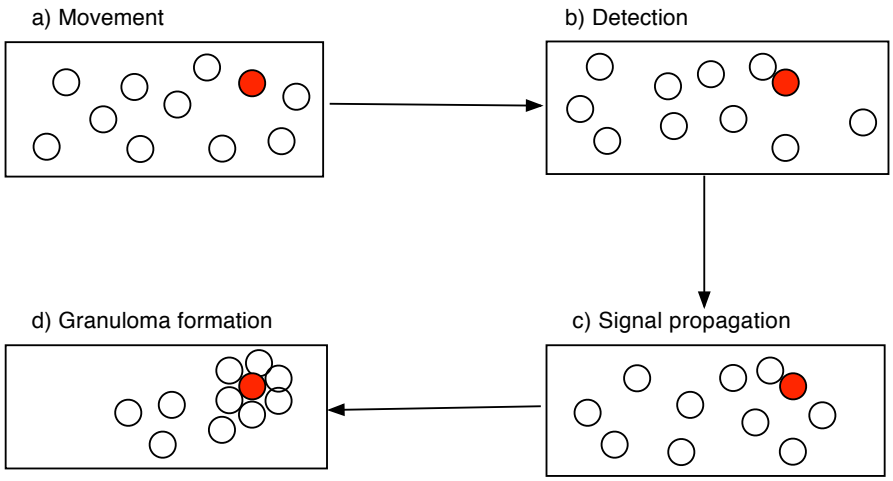


Figure 4.9: Four stages in granuloma formation based from the literature survey conducted.

During the signal propagation stage, there exists lots of signals emitted by the macrophages. This is shown in figure 4.10. In this figure, we illustrate the different stages of macrophages and shows how the secretion of chemokines attract other macrophages. The infected macrophage which is in black secretes chemokine signals to attract other uninfected macrophages to the site of infection. These uninfected cells move to the site of infections, secretes chemokines signals and form a wall around the infected macrophage to prevent the bacterial infections to infect other macrophages or other cells in the systems. The signals also attract other cells such as T-cells (in chocolate) to move to the site of infections. T-cells are the cells that will try to activate the infected macrophages and help to remove bacterial infections that infect macrophages in the system.

From the model we also identified that the main cells involved during the formation of granuloma in table 4.1. They are the macrophages, T-cells, cytokines and chemokines that act as the signalling mechanisms. Chemokines will not only attracted other macrophages to move towards the site of infection but it is also important as it will activate T-cells that

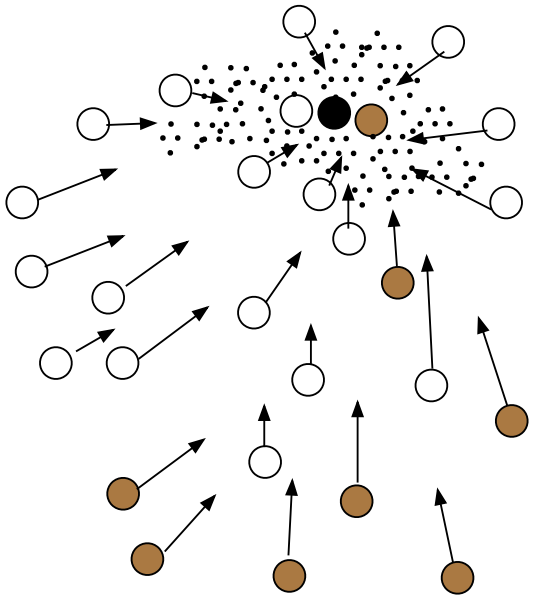


Figure 4.10: Signals propagation and cells in granuloma formation. Legend of colours: black is the infected macrophage, white is the uninfected macrophage and chocolate represents T-cells. The dots represent the chemokine signals emitted by the infected macrophages and the arrows represent the movement of the uninfected macrophages as well as T-cells to the site of infections.

Table 4.1: Cells and their functions in granuloma formation

Cells	Main function
Macrophages	Changes its behaviour once infected by bacteria
Cytokines, chemokines	signalling mechanisms
T-cell	Activate the infected macrophages and controlling the dissemination of bacterial growth

will secrete cytokines' signals that is useful in activation of macrophages. T-cells and activated macrophages are able to kill extra cellular bacteria that will control infections in immune systems.

We have presented here the domain model summarising the literature of granuloma formation. The model is expressed with UML diagrams: the activity diagram in representing the regulation and the state machine diagram in expressing the low-level dynamics of the cells involved during this process. Since the primary purpose of preparing the domain model is to help our understanding of the process of granuloma formation, we have found that it is a very useful exercise, even though we do not capture all interactions and cells during the process. This is because it is difficult to capture all the interactions as most are still being explored by biologists and the diagrams will become cluttered. For example, the dendritic cells are not thought to be altered by the granuloma formation process, but

merely consume and produce signals. They are therefore not important in subsequent simulation for our engineering work. We believe that modelling the process of granuloma formation with UML diagrams is useful before the implementation of the simulation.

With engineering functionality and constraints clearly identified in section 2.5 the abstract model of the immunology that we developed in this section is meant to represent our understanding based on the literature survey that we have conducted, and translated into computational model (in our case an agent based simulation) in section 4.3 and section 4.4. The models are not been validated by the domain experts as the models serve as the conceptual model summarising the literature on granuloma formation and as proposed by Hart & Davoudani (2011) the validation process can be shifted to validating functionality which we present in chapter 6. This is due to the fact that we are proposing and immune-inspired solution for swarm robotic systems and the model and the simulation prepared need to be consistent with the engineering constraints rather than to be validated by the domain experts. In this respect, modelling can be viewed as explorative, rather than restrictive, with no requirement for the models to be biologically plausible (Hart & Davoudani, 2011).

4.3 Platform Model

Having outlined the domain model in section 4.2, we now present the platform model for the development of agent-based simulation for granuloma formation. This is the implementation of behaviour that is inspired by granuloma and is to be used in the design of fault tolerant algorithms for swarm robots. This platform model focuses on how the process is going to be implemented and executed in the simulation platform. The two main aspects that we are interested are the agents (cells involved during the simulation) and the environment in which they interact. Since we are interested in the behavior of macrophages and T-cells during the formation of granuloma, it is sensible for these components to be the agents of our systems. The TNF signals will be an element of the macrophage agent, and INF signals will be an element of T-cell agent. This is to allow us to differentiate different type of signals during the process of granuloma formation as described in section 4.2.2. The environment will consist of the communication region and the chemokine gradient.

Based on the agent and the environment aspects that have been identified, we discuss here how these have been implemented in our agent-based simulation. In section 4.3.1 we detail the environment for the simulation, followed by a description of the chemokine space residing in the environment in section 4.3.2. Finally. we outline the three agent types in section 4.3.3

4.3.1 The Environment

The approach we have taken to model the environment is based on the previous work done by Segovia-Juarez et al. (2004), who have modelled granuloma formation and the interactions of cells during the formation of granuloma due to chemical gradients. In their work, they have combined continuous representations of chemokines with discrete macrophages and T-cells agent in a cellular automata-like environment. In our model, in representing the environment, we build a two-dimensional (2-D) grid of cells where the infection and formation of granuloma occur. The environment consists of chemokine space and agent space. The agent space is where the agent can interact and communicate whilst the chemokine space models the chemokines produced by the agents when it is infected by the bacteria to attract T-cells to move to the site of infection.

Figure 4.11 depicts the environment showing the representation of the agents in the environment. The diagram indicates the initial state of the simulation, where an extracellular bacteria is introduced near the centre of the environment. The uninfected macrophages are randomly distributed in the environment. The behavior of bacteria and uninfected macrophages agents and their rules will be described in further detail in section 4.3.3. Time is represented in the simulation by discrete steps or iterations. For every time step, the chemokine space is updated and agents move according to the specified rules in the environment. This includes the movement of uninfected macrophages and T-cells towards the site of infection and the interaction between those cells in the environment.

4.3.2 The Chemokine Space

In this simulation, we represent chemokines (TNF and IL-12) as the attractors for macrophages and T-cells. Each cell in the chemokine space holds an integer value representing the chemokine concentration. By having the chemokine concentration, it allows the agents in the environment to perform chemotaxis: a directed random walk towards the production of attraction (Goldsby et al., 2003). At each iteration of the chemokine space model, the chemokine values update according to a diffusion rule, which is applied to all cells. The diffusion rule works as follows: for each cell, integer division by 9 is performed on the chemokine value and the result is shared between the nine neighbours. The remainder, r, from this division is then shared out randomly between. When applied to every cell, the effect of the diffusion rule in the space is to smooth the chemokine concentration over the entire chemokine space, whilst leaving a level of variability at the local level. A summary of the chemical space update rules is described in algorithm 3 as described by Goldsby et al. (2003). T-cells and macrophages will move towards higher concentrations of chemokine space.

Algorithm 3: Algorithm for updating chemokine values in chemokine space that contains external bacteria (Goldsby et al., 2003). *c* is the cell in the chemokine cells, *q* in the quotient, *r* is the remainder and *o* is overspill chemokine. In general, the algorithm sets infected cells at random and then models diffusion from each infected cell as described in Goldsby et al. (2003).

```
1  begin
2      Create chemokine space, to hold chemokine values
3      Populate each cell in chemokine space, with a randomly generated integer
       value between 0 and a user defined maximum (i.e denote as infected cells, each
       infected cell then has its own gradient value)
4      foreach iteration do
5          Set level of chemokine's value to 0
6          Set level of overspillchemokine, o, to 0
7          foreach cell, c in chemokines space do
8              Integer divide value in c by 9 and assign the quotient to q and remainder
               to r
9              foreach moore neighbour, n of c do
10                 if n is outside chemokines space then
11                     Increment o by q
12                 else
13                     Increment the value of cell n in chemokine space by q
14                 end
15             end
16             foreach r do
17                 Increased the associated moore neighbour by 1
18             end
19         end
20     end
21 end
```

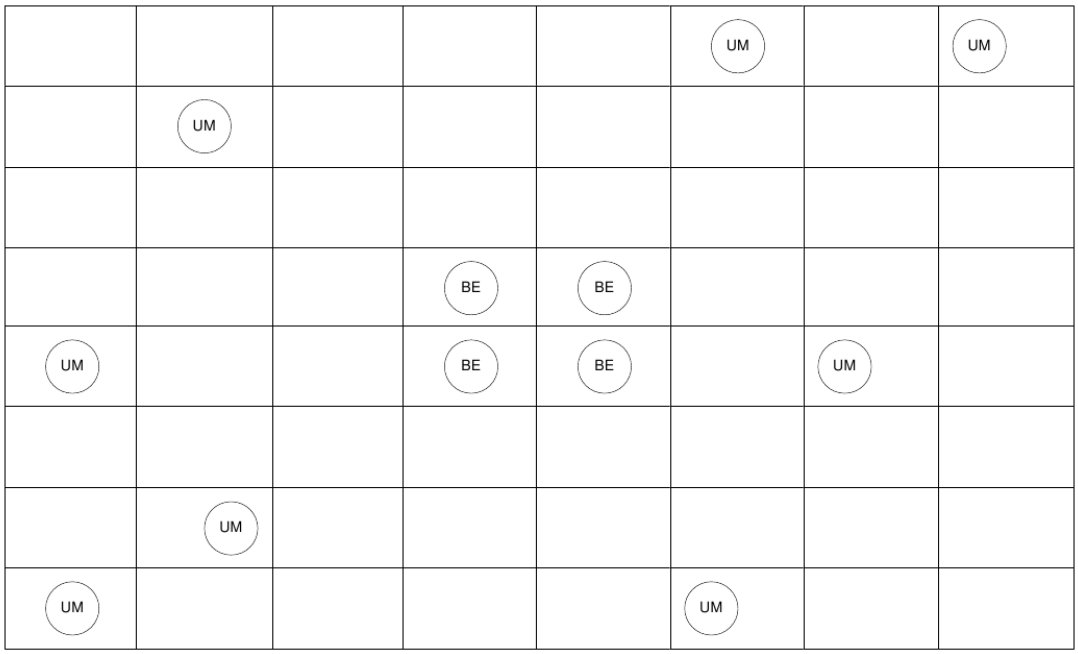


Figure 4.11: The two-dimensional tissue environment for the agent-based simulation for granuloma formation. Shown here is the initial condition of the simulation, where initial load of bacteria (Be) is placed in the center of the environment and the uninfected macrophages (UM) is located randomly in the environment.

4.3.3 Agents and Their Rules

As discussed above, the two agents in our environment are macrophage and T-cell agents. In general, macrophage and T-cells agents move randomly in the environment if there is no chemokine concentration value in the environment. However if there is a space in the environment with a higher concentration value, then there is a tendency for the agents to perform chemotactic motion (a tendency of cells to migrate toward or away from certain chemical stimuli) moving towards the infected macrophages.

Extracellular bacteria and intracellular bacteria : To simulate initial infection, we introduce an extracellular bacterial (BE) agent near the center of the environment. Extracelullar bacteria replicate at a rate α and can be phagocytosed, at a rate β, by macrophages. Essentially, this means that the macrophages stop eating bacteria when they become 'full'. Macrophages die at the rate of γ at which the intracellular bacteria are released. A macrophage is able to hold up to *N* intracellular bacteria (BI) and if any of these N replicate, the macrophages burst, releasing extracellular bacteria at the rate N_β.

Macrophage : As described in section 3.5, macrophages have essentially three states:

uninfected, infected and activated. The rules for macrophage are outlined in algorithm 4 and describe below.

Algorithm 4: Simple rules for macrophages

```
1  begin
2      if BI>0 and BE>0 then
3          return infected macrophages
4      if BE>0 and BI<0 then
5          return chronically infected macrophages
6      if BE<0 and BI>0 then
7          return chronically infected macrophages
8      else
9          return uninfected macrophages (UM)
10     end
11 end
```

T-cell agent :T-cell agents follow the chemokine gradient to the infected cells. T-cell agents each have an activation threshold that is used to determine when they change state from naive to activated. If the activation is above the activation threshold, then the T-cell agent becomes activated and can activate the infected macrophages.

4.3.4 The Simulation

We present in figure 4.12 the visual behaviour of the granuloma formation simulation with a suitable set of parameters (we will discuss the setting of parameters in section 4.5). The simulation begins with an initialisation phase where uninfected macrophages are introduced into the environment. In the absence of extracellular bacteria, the uninfected macrophages randomly change their positions and each of them will perform a random walk in the environment. Then, extracellular bacterial infections are introduced to the environment and the chemokine gradients are established, to attract other macrophages in the environment to move towards the infections. When an uninfected macrophage enters a compartment containing extracellular bacteria, it will likely become infected. This will lead to a small number of macrophages that have been infected, clustered near the centre of the simulation environment. These infected macrophages will release chemokines in the environment that direct the movement of the other cells (T-cells and uninfected macrophages). Meanwhile, the intracellular bacteria which replicate within the infected macrophages will be likely to become chronically infected macrophages when the number of intracellular bacteria residing in the infected macrophages reach a certain threshold value in the environment. This will lead to the structure of granuloma consisting of a number of chronically infected macrophages surrounded by uninfected macrophages and other

cells in the environment. As the intracellular bacteria continue to replicate itself within a chronically infected macrophages, it will eventually burst when the carrying capacity (or a threshold value) is reached. This will spread bacterial infections to the neighbouring compartments resulting in uninfected macrophages in the neighbouring compartments to be infected. In the simulation, after some times, T-cells will enter the environment and move towards higher concentration value of chemokines. We enforce such a delay, taking into account the time taken for the antigen-presenting cells (such as dendritic cells) to migrate to the lymph node and for the T-cells to move from the lymph node to the site of infection (this process is not shown in this simulation). When a T-cell enters a compartment containing infected macrophages, it will be likely to activate the infected macrophages. These activated macrophages can control the infections; meanwhile if there are no T-cells in the environment, the infected macrophages will continue to spread the infections in the environment.

4.4 Simulation Platform

The model has been implemented in simulation with Netlogo [1] which is a programmable modelling environment for simulating natural and social phenomena. NetLogo is particularly well suited for modelling complex systems developing over time. Modellers can give instructions to hundreds or thousands of agents all operating independently. This makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from the interaction of many individuals. The simulator can be run interactively via a graphical user interface (GUI) allowing for batch simulation. Using this simulation, we can observe the granuloma formation that emerges from the interaction between the agents in the environment, and test different types of hypotheses that have been described in section 4.2.

The simulation can be configured via a number of user-defined parameters that are outline in table 4.2. These parameters are related to both the configuration of the agents and the simulation. For example, in using the simulator, user can define how long the simulation will run by specifying the number of simulation's iteration. User also is able to change the configuration of the agents in the simulation. For example, to analyse the effect of the bacterial growth rate, the signalling mechanisms' diffuse rate as well as changing the number of agents in the environment. Given these sets of parameters, the simulation executes the simulation based on figure 4.12 to produce the simulation run. From here, the simulation is run for the required number of iterations and at each iteration, agents move and interact with each other. The bacterial population statistics are updated

[1] http://ccl.northwestern.edu/netlogo/

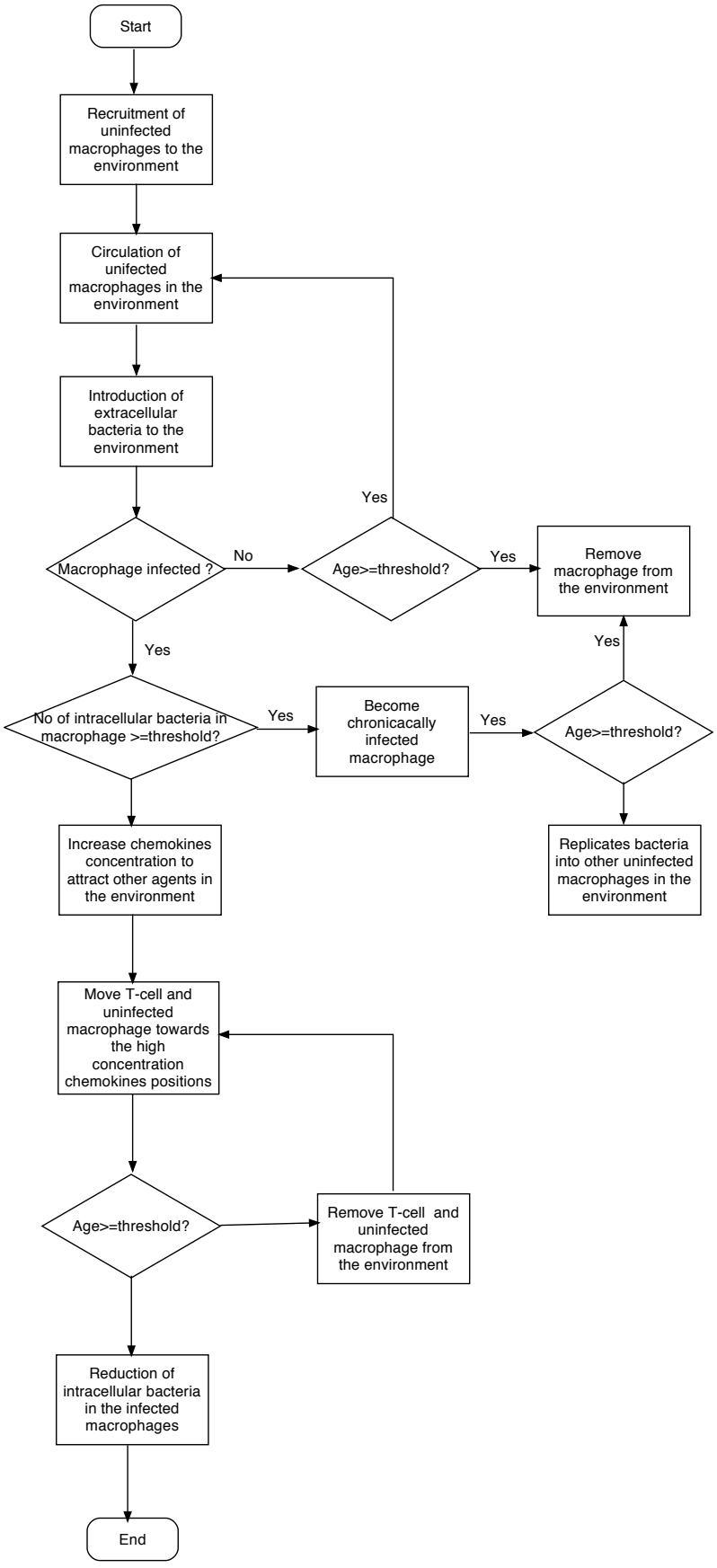


Figure 4.12: Flowchart showing the flow of the simulation for granuloma formation.

along with the display.

Table 4.2: Parameter for the Netlogo simulator for the agent-based model of granuloma formation.

Parameter	Description
Simulation time	Number of simulation iterations for the simulation
T-Cell arrival time	Number of simulation iterations to insert T-cells to the environment
TNF diffuse rate	The diffusion rate for TNF factor
IFN diffuse Rate	The diffusion rate for IL-12 cytokines
Intracellular bacterial replication value	Intracellular bacteria growth rate
Extracellular bacterial replication value	Extracellular bacteria growth rate
Number of macrophage	Number of macrophage in the environment

If we refer back to the activity diagram in section 4.3, there are three main stages in granuloma formation: 1)initiation of immune response, 2)formation of granuloma, 3) removal of granuloma formation (bacteria). In representing this stages in simulation, as depicted in figure 4.12, we represent each stages as follow:

- initiation of immune response: we represent this stage by recruiting the uninfected macrophages in the simulation. The uninfected macrophages moves randomly in the Netlogo simulator. Then, a bacterial infection is introduced in the centre of the simulator to initiate an initial immune response. Once the bacteria starts to infect a macrophage, then the level of chemokines in chemokines space is increased as described in section 4.3.2. This is in accordance to the description in figure 4.3 and figure 4.4, where the macrophages start to emit signals once been infected by bacteria. The signal is meant to attract other uninfected macrophages to move to the site of infection, so that they can start forming a 'wall' around the infected cell to prevent the infection to infect other cells in the environment. Using Netlogo, we differentiate two space for the agent and the chemokines signal. They are the agent space and the chemokines spaces. The chemokines spaces will be activated when there is an agent in the space been infected by the bacteria. If there is no infection, then the chemokines spaces will be updated. The chemokines spaces are updated as described in section 4.3.2.

- formation of granuloma: in Netlogo, once a macrophage is infected by a bacteria, we then increase the value of intracellular bacteria parameter. If the value is more

that the threshold value specified, then the chemokines concentration value will also be increased, leading to the movement of other uninfected macrophages as well as T-cells to the site of infections. This is reflecting the domain model for macrophages in figure 4.4 and figure 4.5.

- removal of granuloma formation (bacteria): in Netlogo,, the removal of granuloma formation is done when there is T-cell agent available in the neighbourhood of infected macrophages agent. The removal of granuloma formation is done by reducing the number of intracellular bacteria in macrophages.

Table 4.3: Mapping from the domain model to Netlogo.

Domain Model	Netlogo
Bacterial infection	bacteria agent
Uninfected macrophages	uninfected macrophages agent
TNF signals	TNF chemokines spaces
IL-2 signal	IL-2 chemokines spaces
Death of macrophage	intracellular bacteria level greater then threshold and age is greater then age threshold
Death of T-cell	age is greater then age threshold
Bacterial infections	intracellular bacteria value

4.5 Results Model

The previous sections have been concerned with developing the domain model, platform model and agent-based simulation for granuloma formation. In accordance with our goal as outlined in section 4.1, in this section, we describe the results that we obtain with the simulation described above. We first define initial conditions used for our simulations in section 4.5.1, and then outline different infection outcomes that the simulations reproduce. To better understand the dynamics leading to these outcomes, we describe in detail the early dynamics of the process of granuloma formation and how this contributes to determining infection outcome, as well as the later states of the infections in granuloma formation in section 4.5.2. This leads us to examine the effects of certain key parameters such as the delay of arrival time of T-cells in section 4.5.3 and the effect of variations in parameters on growth of granuloma size in section 4.5.4.

4.5.1 An Example Run

Figure 4.13 depicts the initial conditions for all simulation runs. A number of macrophages are randomly placed in the simulation, and these macrophages are in their uninfected state. As mentioned in section 4.3.1, a small initial infection of extracellular bacteria in introduced near the centre of the environment. There are no T-cells, TNF chemokines and IL-12 chemokines in the environment initially. We first run the simulation to look at the outcome of the simulation by varying some of the parameters to explore the outcome of the simulation. The example of the parameter is displayed in table 4.4. Here we compare the formation of granuloma with different TNF and IFN diffusion rates with other parameters remain the same. We also change the intracellular and extracellular replication rate. The processes that we wish to observed in the context of our simplified agent-based simulation for granuloma formation are: 1) the recruiting of immune cells to the site of infections and 2) the growth of bacteria: small or slow growing granulomas leading to the containment or large growing of granulomas leading to dissemination of the infections.

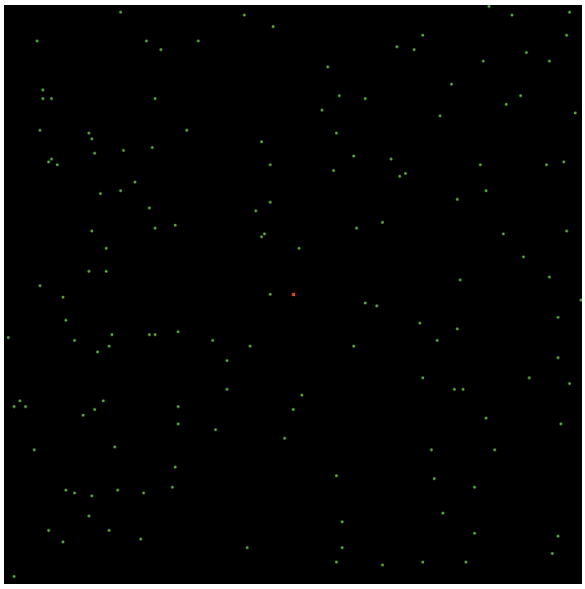


Figure 4.13: The initial condition of agents during the simulation of granuloma formation in Netlogo. The red pixel represents the bacterial infection in the simulation and the green pixels represent the uninfected macrophages.

Through out the experiments, we see the containment and the dissemination of bacteria in the environment. As described by Segovia-Juarez et al. (2004), containment is characterised by the survival of extracellular bacteria in regions surrounded by other cells (macrophages, T-cells and other cells in the systems) and/or slow growth of bacteria within the infected macrophages. With dissemination, on the other hand, there is large and increasing amount of infected cells and the extracellular bacteria can spread around

Table 4.4: Initial parameter definitions of the simulation.

Parameter	**A**	**B**	**C**
Simulation time (seconds)	2000	2000	2000
T-Cell arrival time (seconds)	500	500	1000
TNF diffuse rate	0.1	0.5	0.1
IFN diffuse Rate	0.1	0.5	0.1
Intracellular bacterial replication value	4	8	4
Extracellular bacterial replication value	4	8	4

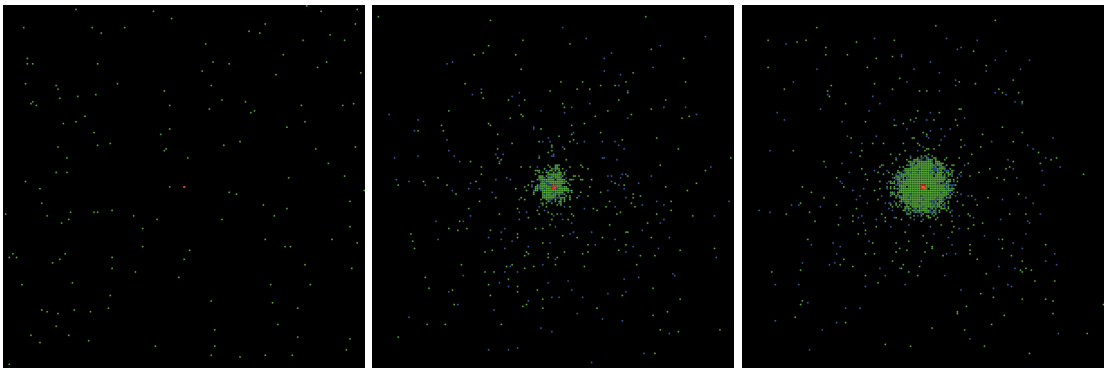
the environment. Figure 4.14 and figure 4.15 show simulations representing the containment of intracellular bacteria and the dissemination of extracellular bacteria, via a series of snapshot during the simulation process. Figure 4.14 shows an example of a containment of intracellular bacteria during the formation of granuloma obtained with parameters from table 4.4 column A. Meanwhile figure 4.15 shows an example of containment and dissemination intracellular and extracellular bacteria during the formation of granuloma, with parameters from table 4.4 column B. In the following sections, we will analyse the observed simulation results further.

Figure 4.16 and 4.17 depict the formation of granuloma at the end of some of the simulation runs. From both figures we can see that there are uninfected macrophages, infected macrophages, chronically infected macrophages as well as T-cells. In we refer to figure 4.16, when T-cells are introduced early in the simulation, the infections are able to be controlled and the formation is not as big as depicted in figure 4.17.

4.5.2 Early and Later Stages of Infections

To better understand the containment and dissemination of intracellular and extracellular bacteria during the formation of granuloma, we compare the dynamics observed in figure 4.18 and figure 4.19 showing, respectively, containment and dissemination scenarios during the early stages if infections. From the initial state described above, we observe that the system typically evolves as follows.

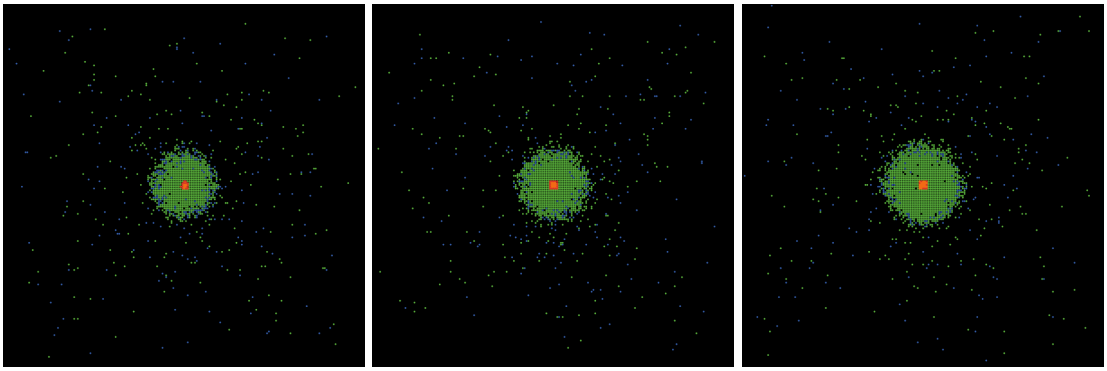
During the early stages of the simulation, the uninfected macrophages randomly change position, performing a random walk in the absence of chemokine as well as the extracellular bacteria in the environment. When extracellular bacteria is introduced to the environment, the chemokines gradient starts to direct the movement of the uninfected macrophages. When the uninfected macrophages enters a compartment containing the extracellular bacteria, it is likely to become infected. This leads to a small number of infected macrophages, clustered near the centre of the environment where the extracellular bacteria is located. These infected macrophages then release large amounts of



(a) Formation or granuloma upon initialisation, where t= 200 and be=4

(b) Formation or granuloma after 500 iterations, where t= 500 and be=4

(c) Formation or granuloma after 1000 iterations, where t= 1000 and be=4

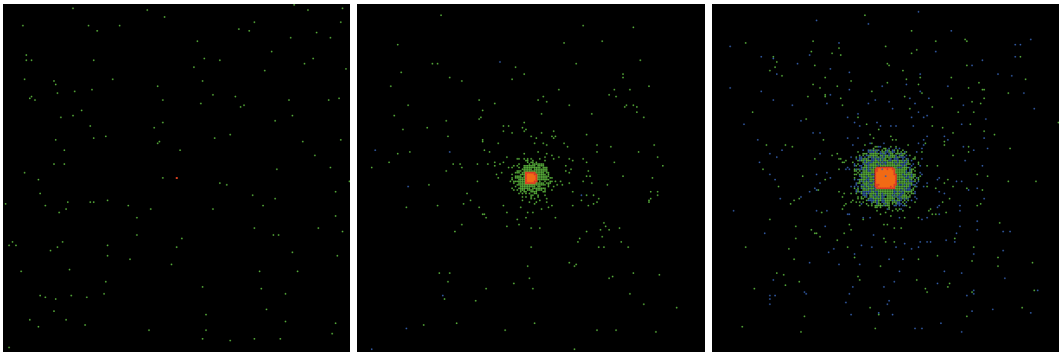


(d) Formation or granuloma after 1250 iterations, where t= 1250 and be=4

(e) Formation or granuloma after 1500 iterations, where t= 1500 and be=4

(f) Formation or granuloma after 2000 iterations, where t= 2000 and be=4

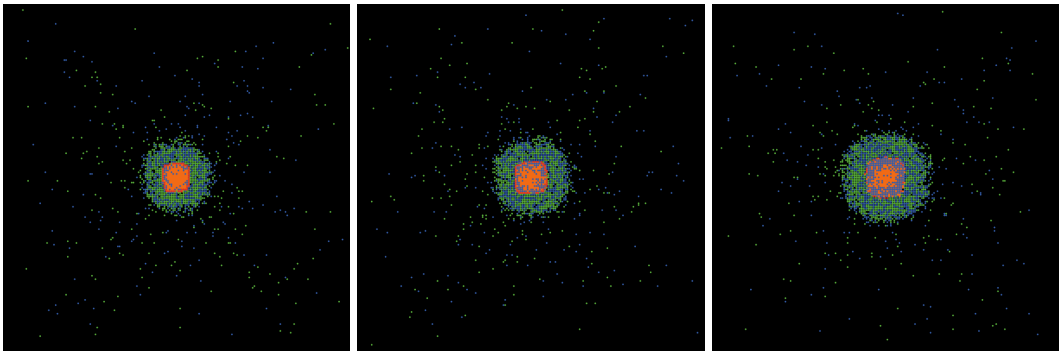
Figure 4.14: The formation of granuloma. Simulation parameters in Table 1, column A. The colours represent resting macrophages (green), infected macrophages (orange), chronically infected macrophages (red), T-cells (blue), and extracellular bacteria (yellow).



(a) Formation or granuloma upon initialisation, where t= 200 and be=4

(b) Formation or granuloma after 500 iterations, where t= 500 and be=4

(c) Formation or granuloma after 1000 iterations, where t= 1000 and be=4



(d) Formation or granuloma after 1250 iterations, where t= 1250 and be=4

(e) Formation or granuloma after 1500 iterations, where t= 1500 and be=4

(f) Formation or granuloma after 2000 iterations, where t= 2000 and be=4

Figure 4.15: The formation of granuloma. Simulation parameters in Table 1, column B. The colours represent resting macrophages (green), infected macrophages (orange), chronically infected macrophages (red), T-cells (blue), and extracellular bacteria (yellow).

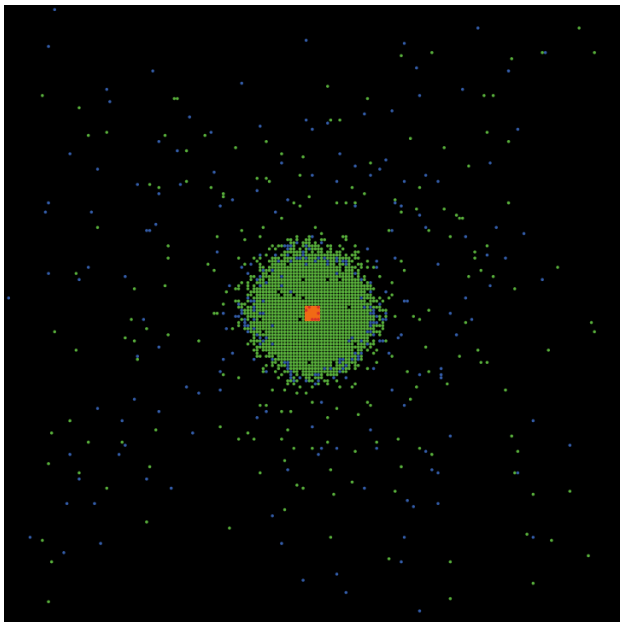


Figure 4.16: One of the end condition of agents during the simulation of granuloma formation in Netlogo. The red pixel represents the bacterial infection (infected macrophages) in the simulation and the green pixels represent the uninfected macrophages and the blue pixels are the T-cells.

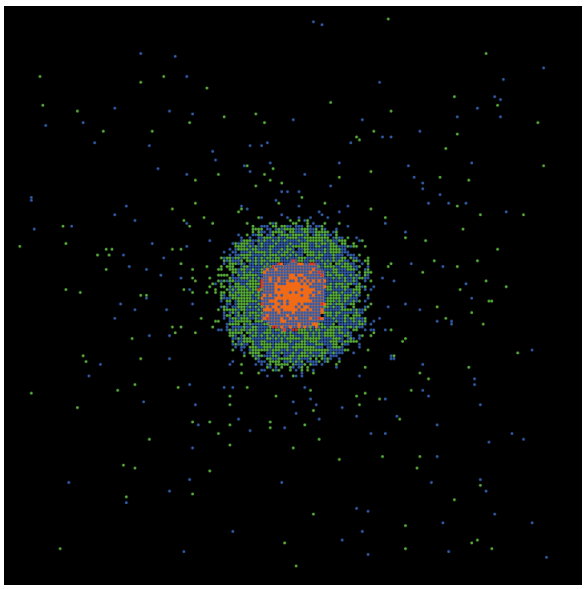
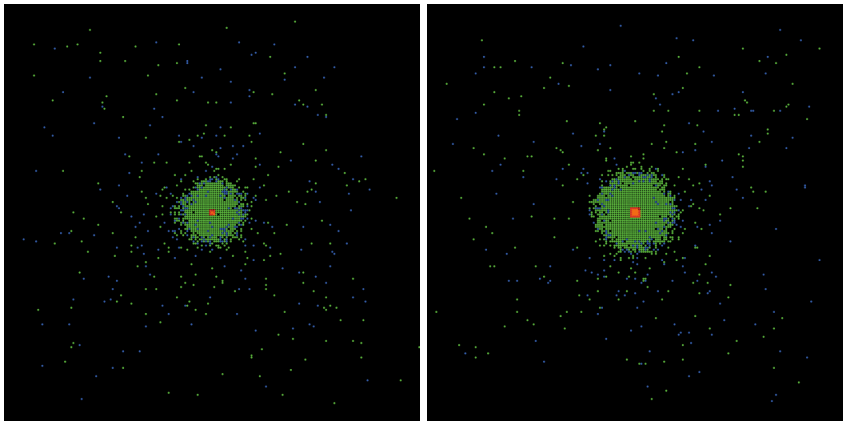


Figure 4.17: One of the end condition of agents during the simulation of granuloma formation in Netlogo. The red pixel represents the bacterial infection (infected macrophages) in the simulation and the green pixels represent the uninfected macrophages and the blue pixels are the T-cells.



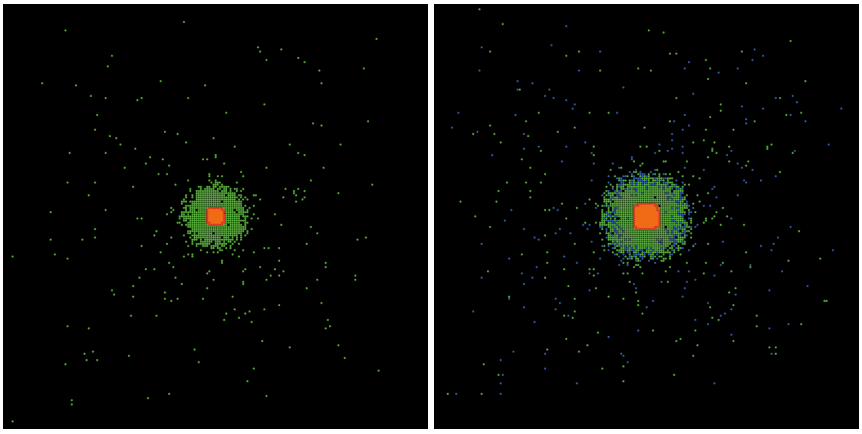
(a) Formation or granuloma after 500 iterations

(b) Formation or granuloma after 1500 iterations

Figure 4.18: Granuloma showing containment at early stages of infections. Simulation parameters in Table 1, column A. (a)The granuloma showing containment at t = 500. (b)The granuloma showing containment and dissemination at t = 1500. The colours represent resting macrophages (green), infected macrophages (orange), chronically infected macrophages (red), T cells (blue), and extracellular bacteria (yellow).

chemokines, not shown in the simulation environment, that create a chemokine gradient on the environment. These gradients direct the movement of other remaining uninfected macrophages that already in the environment, since their random walks are biased based on the higher chemokines concentration levels. Furthermore, the additional uninfected macrophages recruited in the environment also migrate towards the source of chemokines that is released by the infected macrophages. Meanwhile, intracellular bacteria begin to replicate within the infected macrophages, which will then become chronically infected macrophages that will form the initial structure of granuloma, consisting of a small number of chronically infected macrophages surrounded by the uninfected macrophages. This can be seen in both snapshots in figure 4.18(a) and figure 4.19(a). As intracellular bacteria continue to replicate within chronically infected macrophages, they eventually burst when their threshold is reached, which will spread the extracellular bacteria to neighbouring compartments in the environment. This results in a new round and a bigger number of infected macrophages, as shown in figure 4.18(b) and figure 4.19(b).

In both settings, as described in table 4.4, after a delay of 500 simulation seconds, T-cells enter the environment. We enforce a such delay to take into account the time taken for antigen-presenting cells (such as dendritic cells) to migrate to the lymph node, and for T cells to migrate from the lymph node to the site of infection (this is not shown in the simulation). The length of this delay affects the growing granuloma structure of



(a) Formation or granuloma after 500 iterations

(b) Formation or granuloma after 1500 iterations

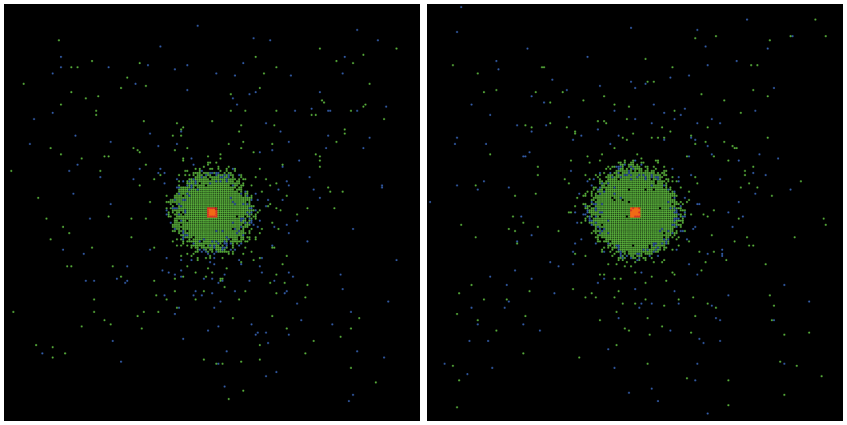
Figure 4.19: Granuloma showing containment at early stages of infections. Simulation parameters in Table 1, column C. (a)The granuloma showing containment at t = 500. (b)The granuloma showing containment and dissemination at t = 1500. The colours represent resting macrophages (green), infected macrophages (orange), chronically infected macrophages (red), T cells (blue), and extracellular bacteria (yellow).

infected and uninfected macrophages, and we also study the effects of variations to the delay length in section 4.5.3. During the movement of T-cells, like macrophages, they tend to be recruited and move toward the edges of granuloma. Therefore, as depicted in figure 4.18(b) and figure 4.19(b), there are some T-cells surrounding chronically infected macrophages as well as infected macrophages. The process of T-cell migration in the environment will influence the spread or containment of infection. For example, when a chronically infected macrophage bursts and causes an infections in neighbouring uninfected macrophages, there are two possibilities for the process of granuloma formation. The possibilities are:

1. if T-cells have migrated to the immediate neighbourhood of a chronically infected macrophage when bursting occurs, these T-cells may activate the infected macrophage(s), which will contribute to the control of infection.

2. if there are no T-cell migrating to the nearby neighbourhood, the infected macrophages may progress to a chronic infection state, and will cause the infections to spread out.

Figure 4.20 and figure 4.21 show the containment and dissemination of extracellular bacteria during the later stages of the process of granuloma formation. Figure 4.20(a), shows the containment of extracellular bacteria and the T-cells moves towards the centre of granuloma, that will activate the infected macrophages that will control the spread



(a) Formation or granuloma after 1500 iterations

(b) Formation or granuloma after 2000 iterations, where t= 200 and be=4

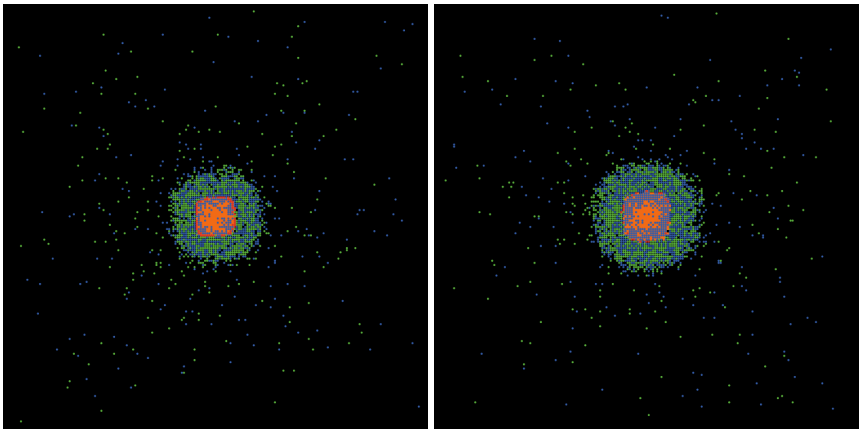
Figure 4.20: Granuloma showing containment at early stages of infections. Simulation parameters in Table 1, column A. (a)The granuloma showing containment at t = 500. (b)The granuloma showing containment and dissemination at t = 1500. The colours represent resting macrophages (green), infected macrophages (orange), chronically infected macrophages (red), T cells (blue), and extracellular bacteria (yellow).

of infections. Figure 4.21(a) shows the containment of extracellular bacteria, with the infected macrophages surrounded by the uninfected macrophages, with extracellular bacteria in the centre of the granuloma. Figure 4.21(a) clearly shows a larger granuloma with more infected and activated macrophages at the edges of the granuloma; where bacteria are not being contained and the granuloma continues to grow. Figure 4.20(b) shows the containment of extracellular bacteria with the infected macrophages at the centre of the granuloma and activated macrophages, containing the spread of the infection. Conversely, figure 4.21(b) shows the dissemination of extracellular bacteria, where the bacteria are not being contained, inducing spread of bacteria outside the granuloma. Many activated macrophages and T-cells are found in the environment.

4.5.3 Varying T-cells Arrival Time

To explore the role of parameter values during the formation of granuloma, we perform analyses for the parameters available in our simulation. We first discuss the influence of varying T-cells arrival time in this section. Secondly, we explain the effects of intracellular bacterial growth rate in section 4.5.4.

In the absence of prior infection or vaccination, the adaptive immune response takes anywhere from five to ten days to develop activated immune cells specific for respond-



(a) Formation or granuloma after 1500 iterations, where t= 200 and be=4

(b) Formation or granuloma after 2000 iterations, where t= 200 and be=4

Figure 4.21: Granuloma showing containment at 2000 simulation seconds. Simulation parameters in Table 1, column C. (a)The granuloma showing containment at t = 500. (b)The granuloma showing containment and dissemination at t = 1500. The colours represent resting macrophages (green), infected macrophages (orange), chronically infected macrophages (red), T cells (blue), and extracellular bacteria (yellow)

ing to the pathogen (Janeway & Medzhitov, 2002; Janeway et al., 2005). Therefore, macrophage activation is important in controlling granuloma size. To capture this effect, we simulate a constant time delay before T-cells begin arriving at the infection site. During this delay we vary the time for initial arrival of T-cells to the site between 500 to 2000 simulation time steps.

Figure 4.22 shows that the dissemination of extracellular bacteria at different arrival times of T-cell. From this figure, the dissemination of extracellular bacteria is affected on the arrival time of T-cell in the environment. When T-cells arrive in the first seconds of infection (i.e. no delay), we observe that the dissemination of extracellular bacteria is low. Even though there is no complete clearance of the bacteria, we observe that T-cells that move towards the site of infection will immediately activate the infected macrophages, leading to the ability of macrophages to control the infections to infect other cells in the environment and completely surround the extracellular bacteria. Thus, it can be assumed that at this stage, the macrophages are containing or trying to clear the infections that occur.

This is further explained in figure 4.23. From figure 4.23, we show the granuloma size in the simulation, showing the dissemination of the infections in the environment when T-cells are introduced to the simulation at different time step (from t=100 to t=2000). From

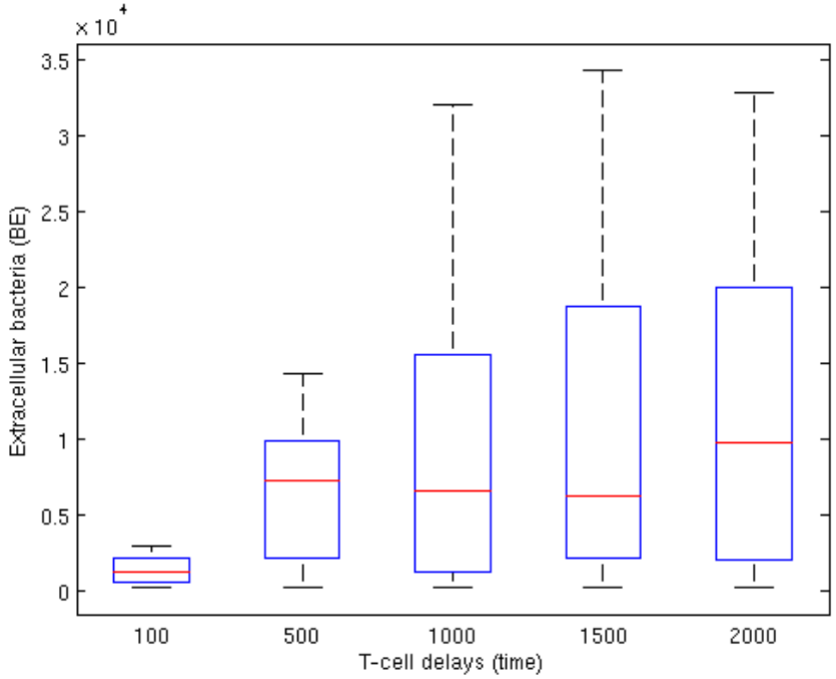
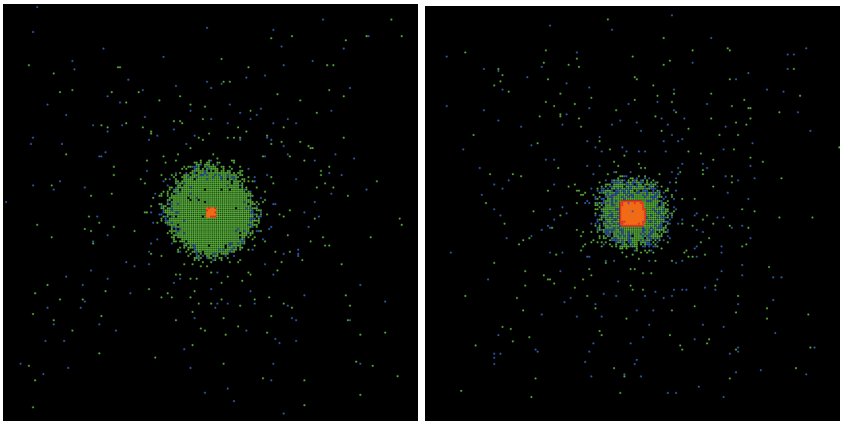


Figure 4.22: Box plot of the effect of varying T-cell arrival time on the simulation of granuloma formation for 2000 simulation time for 10 simulation runs.

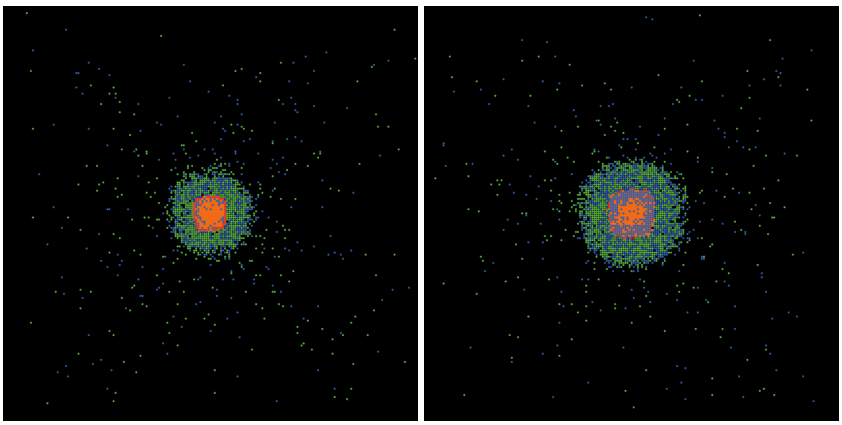
this figure we can see that if T-cells are introduced during early during the simulation, the dissemination of bacteria can be controlled. For example when T-cells are introduced at t=100, the formation of granuloma is small. When T-cells are introduced at T=500, the formation is not as big as when T-cells are introduced later during the simulation run. This can be seen when T-cells are introduced at t=1000, t=1500 and t=2000. For these cases, the dissemination of bacteria cannot be controlled and the formation of granuloma is big.

From the analyses above, we observe that the parameter governing T-cell arrival time to the site of infections is important to granuloma growth (or extracellular bacterial numbers).The figure implies that an important role could be played by successful vaccination of granuloma formation by ensuring an early response from T-cells upon infection. However, this suggest that for a given parameter set, T-cell immunity is crucial at the beginning of the infection. T-cells arriving at later times (as depicted in figure 4.22) do not have the same effect as when they arrive earlier. At the later time, other mechanisms might be also important in controlling infection such as the bacterial growth rate, number of macrophages or other parameters that work to control the infections.



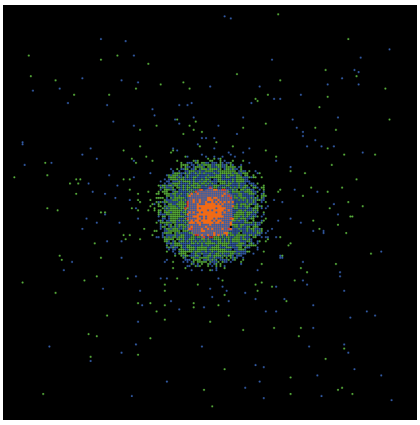
(a) Formation or granuloma upon initialisation, where t= 100 and be=4

(b) Formation or granuloma after 2000 iterations, where t= 500 and be=4



(c) Formation or granuloma after 2000 iterations, where t= 1000 and be=4

(d) Formation or granuloma after 2000 iterations, where t= 1500 and be=4



(e) Formation or granuloma after 2000 iterations, where t= 2000 and be=4

Figure 4.23: The formation of granuloma when T-cells are introduced at different time step The colours represent resting macrophages (green), infected macrophages (orange), chronically infected macrophages (red), T-cells (blue), and extracellular bacteria (yellow).

Table 4.5: Parameter definitions of the simulation.

Parameter	**A**	**B**	**C**	**D**
Simulation time (seconds)	2000	2000	2000	2000
T-Cell arrival time (seconds)	800	800	800	800
TNF diffuse rate	0.1	0.1	0.1	0.1
IFN diffuse Rate	0.1	0.1	0.1	0.1
Intracellular bacterial growth rate	2	4	6	8
Extracellular bacterial growth rate	4	4	4	4

4.5.4 Intracellular Bacterial Growth Rate

To study effect of bacterial growth during the formation of granuloma, we explore the role of intracellular bacterial growth rate in our simulation. All other parameters will remain the same and we increase the intracellular bacterial rate accordingly. The parameters are shown in table 4.5.

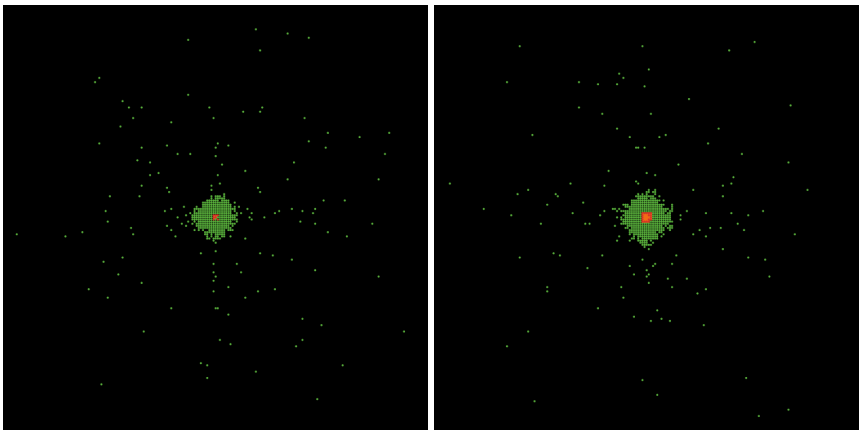
Our results show that during the early time of infections, large intracellular bacterial growth rate is correlated with the extracellular bacterial load, allowing the granuloma to grow fast, leading to a large numbers of cells trafficking to the site of infections as shown in figure 4.24. Later during the simulation, the larger intracellular bacterial growth rate will become larger as shown in figure 4.25.

Figure 4.26 shows the dissemination of extracellular bacteria at different extracellular bacterial growth rate. From this figure, the dissemination of extracellular bacteria is affected on the bacterial growth rate. We observe that the dissemination of extracellular bacteria is low when the intracellular bacterial growth rate is low and as the growth rate increases, then the dissemination of extracellular bacteria is high.

4.6 Validation

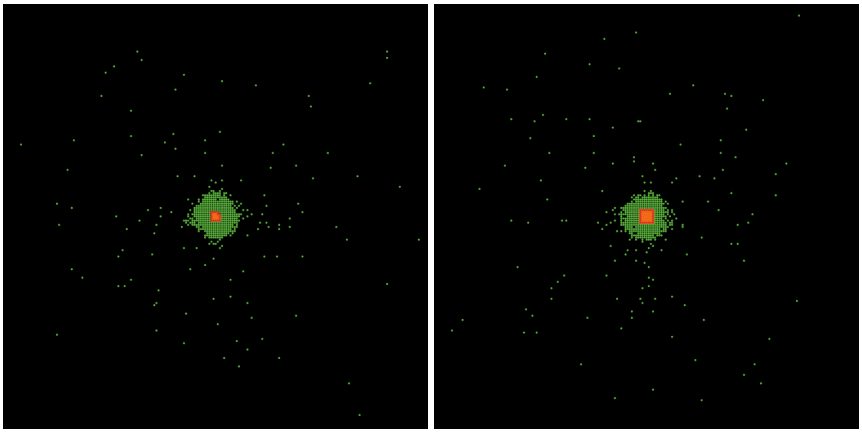
As described in section 4.1, the purpose of the model and simulation of granuloma formation in the thesis is to assist understanding as well as represents the literature survey that we have conducted. Thus, even though validation is important when we model and simulate biological systems, but in our case, the model and simulation of granuloma formation is an exploratory implementation of behaviour that is used in the design of fault tolerant algorithm for swarm robots. In assisting our algorithm design, which is described in chapter 5, we prepare the mapping as well as the analogy from the simulation of granuloma formation with the design of our algorithm in chapter 5.

In section 3.5.3, we have discussed on the properties of swarm robotics and how it can be related to the process granuloma formation as shown in table 4.6. We also provide



(a) Formation or granuloma during 500 simulation seconds, where bacterial growth rate= 2

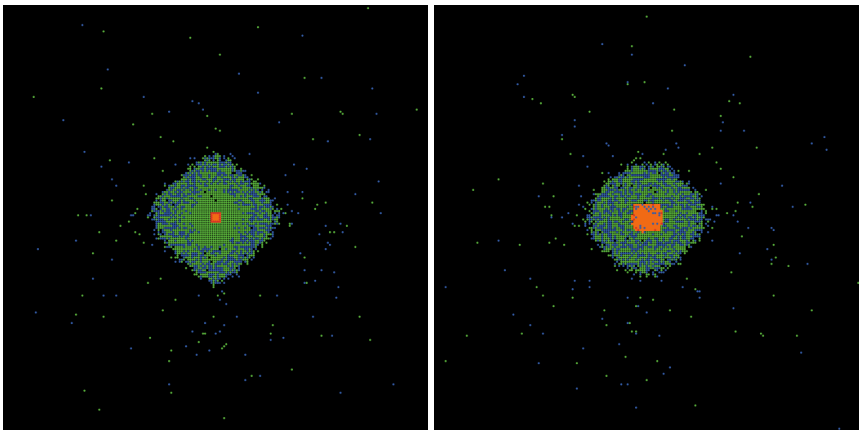
(b) Formation or granuloma during 500 simulation seconds, where bacterial growth rate= 4



(c) Formation or granuloma during 500 simulation seconds, where bacterial growth rate= 6

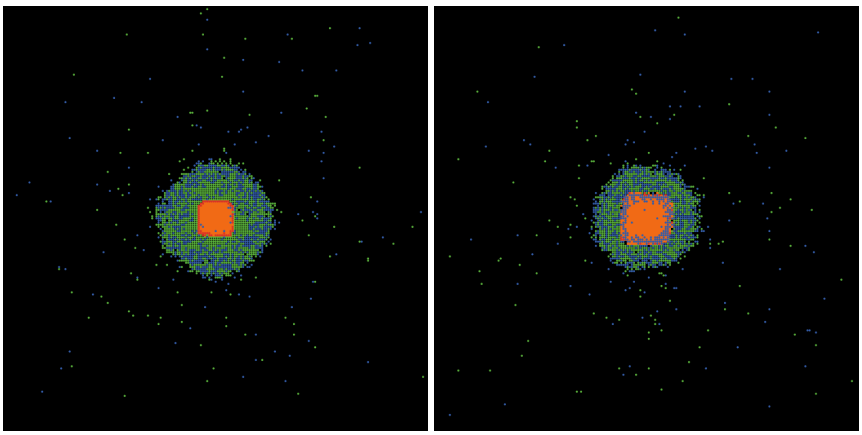
(d) Formation or granuloma during 500 simulation seconds, where bacterial growth rate= 8

Figure 4.24: Granuloma showing containment at the early stages of infections. Simulation parameters in table 4.5. (a)The granuloma showing containment for bacterial growth rate=2. (b) The granuloma showing containment and dissemination for bacterial growth rate=4. (c) The granuloma showing containment and dissemination for bacterial growth rate=6. (d) The granuloma showing containment and dissemination for bacterial growth rate=8. The colours represent resting macrophages (green), infected macrophages (orange), chronically infected macrophages (red), T cells (blue), and extracellular bacteria (yellow).



(a) Formation or granuloma during 2000 simulation seconds, where bacterial growth rate= 2

(b) Formation or granuloma during 2000 simulation seconds, where bacterial growth rate= 4



(c) Formation or granuloma during 2000 simulation seconds, where bacterial growth rate= 6

(d) Formation or granuloma during 2000 simulation seconds, where bacterial growth rate= 8

Figure 4.25: Granuloma showing containment at later stages of infections. Simulation parameters in table 4.5. (a)The granuloma showing containment for bacterial growth rate=2. (b) The granuloma showing containment and dissemination for bacterial growth rate=4. (c) The granuloma showing containment and dissemination for bacterial growth rate=6. (d) The granuloma showing containment and dissemination for bacterial growth rate=8. The colours represent resting macrophages (green), infected macrophages (orange), chronically infected macrophages (red), T cells (blue), and extracellular bacteria (yellow).

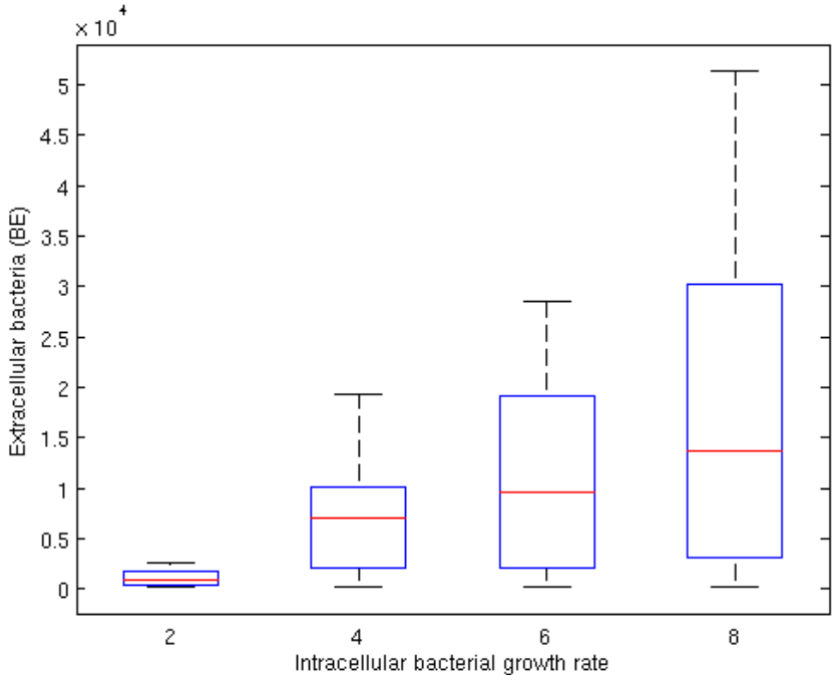


Figure 4.26: The result obtain for ten simulation runs for agent-based simulation for granuloma formation when different intracellular bacterial growth rate are introduced to the simulation.

a general discussion on how granuloma formation can be used in swarm robotic systems for 'anchoring' issue in swarm beacon taxis. Based on our description in table 4.6, we present here the explicit mapping between the agents in the granuloma model and the swarm robots.

Table 4.6: Agents in granuloma formation and swarm robots

Agents in granuloma formation	Agents in swarm robots
uninfected macrophages	robot with enough energy to move
infected macrophages	robot with energy below the threshold value, which is not able to move
T-cells	robot that is assigned to do the healing (sharing) of energy

Based on the experiment conducted with our simulation, we observed few results that relate back to our chapter goal. During the simulation runs, we observe the dynamics behaviour during the formation of granuloma:

1. the timing of recruitment of T-cells can influence the timing of bacterial control. The actual role of T-cells in activating both IFN secretion as well as cell-cell interactions.

2. macrophages in their resting stage take up extracellular bacteria and provide an ideal growth environment, and yet during their activated state state they are able to take up and destroy bacteria.

3. increase in intracellular bacterial rate and chemokine diffusion rate increases the granuloma size, allowing the dissemination of extracellular bacteria in the environment.

Based on this simulation we identified that the properties that we investigated (T-cells arrival time, bacterial growth rate) are relevant to the robotics problem as follows:

1. T-cells arrival time: as described in section 4.5.3, T-cells arrival time plays an important role in controlling the granuloma size as well as the spread of the bacterial infections to the macrophages. In robotic context, T-cells are mapped as robot that is assign to do the healing (sharing) of energy with the faulty robots with less energy in the system. Thus, it is important in the robotic systems to make sure that the robots that can perform the healing process come to the site of infections (in swarm robotic systems the site of infection may refer to the position of the faulty robot(s)) as early as possible so that it can perform the healing process as early as possible. If the healing robot can arrive early, than faulty robots can continue their operation as soon as they have enough energy. Therefore, in designing the algorithm for self-healing swarm robotic systems, we take into consideration this idea and described the algorithm in chapter 5.

2. bacterial growth rate: as described in section 4.5.4, the intracellular bacterial growth rate is important in granuloma formation. As the intracellular bacterial growth rate increases, the granuloma grows fast leading to the large number of cells trafficking to the site of infections due to the signals send by the infected macrophages to the other cells in the systems. In swarm robotics context bacterial growth rate has no direct analogy in swarm beacon taxis work. However, we can still initiate this idea in designing a self-healing swarm robotic systems by taking into consideration on how to allow robots in the systems to move to the faulty robots. We need to identify the number and type of signals that need to be send to all robots in the system and to make sure signalling mechanisms is effectively available in systems. In designing our algorithm, we apply this idea an the discussion of the algorithm is described in chapter 5.

4.7 Conclusion

In section 4.1, we have identified the research context for the model and simulation work presented in this chapter. This goal was to develop a UML model and agent-based simulation to to assist understanding on the interactions of cells during the process of granuloma formation how they could be exploited with regards to an AIS. Our first step was to prepare the domain model of granuloma formation in section 4.2, where we highlighted the boundary of the work and presented the UML model of the behaviour and regulation of the process in granuloma formation. This resulted in identification of the main components that involved in granuloma formation: macrophages, T-cells, and various signalling mechanisms. We then described in section 4.3 and section 4.4 the development of agent-based simulation to enable us on exploring the process of granuloma formation via simulation that worked based on the domain that we prepared earlier. Based on work from this chapter, we will present the granuloma formation algorithm in chapter 5, which is inspired by the our modelling and simulation work in this chapter.

CHAPTER 5

Granuloma Formation Algorithm

In a robotic context, robots will move according to the algorithm in Nembrini (2005), which makes use of local wireless connectivity information alone to achieve swarm aggregation, which make the use of situated communications, where the connectivity information is linked to robot motion so that robots within the swarm are wirelessly glued together. We assume the presence of an anomaly detection system on each robot. Consider the case when a permanent fault is located in a robot, and the robot ceases to move. We can assume that certain visual signals can be sent by the robot which other functional robots nearby can recognise. These functional robots are then attracted towards the faulty robot, akin to how T-cells are attracted by cytokines emitted by an infected macrophage. A limited number of these robots then isolate the fault robot, akin to T-cells surrounding an infected macrophage, but still move around the fault robot so that other functional robots in the swarm are no longer drawn to the 'anchor' point that could be the faulty robot. This approach is ideally used, however, when certain repairs could be initiated. Consider the case then a transient fault occurs in robot that results in a large power drain in the robot. What is required now, is for other robots to share power with each other to re-charge the 'faulty' robot. Employing the approach above allows us to surround the faulty robot with functional robots that are able to share power. Once a power share operation has been completed, then robots in artificial granuloma will then carry on with the tasks they were doing before the fault was identified.

In this chapter, we detail our work on the final two phases of the conceptual framework approach (CFA) of Stepney et al. (2005), which is to abstract principles for the development of a novel bio-inspired algorithm. We first describe in section 5.1 the design

principles for the distributed decision making to facilitate decisions during the process of self-healing in swarm robotic systems. Here, we discuss four key design principles from the models and simulation extracted from work in chapter 4. Based on these design principles, we then continue this chapter by describing our AIS algorithm, the granuloma formation algorithm in section 5.2. In this section we describe the granuloma formation algorithm, and how the design principles described in section 5.1 fit in the algorithm. This chapter ends with section 5.3, where we concluded our work in this chapter.

5.1 Design Principles for Distributed Decision Making

Based on the preceding chapters, we now continue with the development of an algorithm that we can employ in the scenario described in section 2.5. Following the CoSMoS process Andrews et al. (2010), we have developed a model and simulation of the general formation and progression of granuloma formation in chapter 4, rather than in the case of a specific disease. This is due to the fact that we do not wish to model the formation to provide an insight into a biological perspective, but to understand the dynamics of a general model to allow us to distill a series of design principles that we can use to create a novel AIS algorithm, in particular, for the development of swarm robotic systems that are able to contain certain type of errors, and initiate repair strategies to allow energy sharing between robots, when there exist the possibility of robots' energy failing in the system. As discussed in section 2.5, we are assuming that there is a case when a transient fault occurs in the robot that results in a large energy drain on the robot. What is required now is for other robots to share energy with each other to recharge the 'faulty' robot(s). From the model and simulation of granuloma formation that we have developed, we are able to instantiate the idea to this issue. The idea is to surround the faulty robot with functional robots that are able to share energy between themselves. The four key design principles from the models and simulation that we have prepared in chapter 4 are:

1. the communication between agents in the system is indirect consisting a number of signals to facilitate the coordination of agents.

2. agents in the systems react to defined failure modes in a self-organising manner.

3. agents must be able to learn and adapt by changing their role dynamically.

4. agents can initiate a self-healing process dependant to their ability and location.

We describe each of these design principles in section 5.1.1, 5.1.2, 5.1.3 and 5.1.4, which have been taken forward and used as a basis to create a self-healing swarm robotic

system where robots are able to recover from certain types of power failure and are collectively able to recharge and continue operation.

5.1.1 Design Principle I

The communication between agents in the system is indirect consisting a number of signals to facilitate the coordination of agents.

In granuloma formation the communication between macrophages is determined by the level of chemokines secretion. Chemokines not only attract other macrophages to move towards the site of infection but it is also important insofar as they activate T-cells that secrete cytokines' to activate of macrophages. T-cell and activated macrophages are able to kill extra cellular bacteria that control infections in immune systems. Thus, agents are activated by the signal and respond to it. For example, an agent that received signal *A* is activated, and it does the task that is assigned to it; meanwhile those agents that does not receive the signal continue doing their current task and remain inactivated.

In applying this idea to swarm robotic systems, we can have robot(s) that can send and receive different kinds of signal, allowing the robots to communicate with each other. For example, when a robot is 'faulty' or experiencing a large energy drain, it will release signals to the environment. The neighbouring robot(s) that obtained the signal from the environment then follow the signal towards the faulty robots, surrounding it and performing a healing process.

5.1.2 Design Principle II

Agents in the systems react to defined failure modes in a self-organising manner

During the formation of granuloma, when there is an infected cells, other cells in the systems are attracted to the site of infection by messenger molecules emitted into the local neighbourhood by the infected cells. This makes the cells constantly move and interact with each other (between the infected and uninfected cells). The various interactions between the cells in the system result in self-organised positive and negative feedback mechanisms that regulate the level of attraction of other cells in the systems to the site of infection, leading to the formation of granuloma.

In applying this idea to swarm robotic systems, we can have two types of robot. The 'faulty' and 'non-faulty' robots. The 'faulty' robots are the robots that have low energy or experience large energy drain in their body and the 'non-faulty' robots are the robots that have enough energy for themselves as well as helping the other 'faulty' robots. If

the robot is experiencing large energy drain, it emits a signal to the environment allowing other 'non-faulty' robots to move and interact with the 'faulty' robot. This results in self-organised positive and negative feedback mechanisms that regulates the level of attraction between robots in the systems.

5.1.3 Design Principle III

Agents must be able to learn and adapt by changing their role dynamically

During granuloma formation, cells in the system have the ability to learn and adapt their performance to changing environments (for example when there exist infected and uninfected cells). The cells are capable of autonomously navigating, selecting and tracking infected cells. They also change their role (for example from uninfected macrophages to activated macrophages) based on the signals they receive, and the state that they are in at any point in time. This allow cells in the system to change their role dynamically based on the information that they have, allowing them respond algorithmically to outside signals from their surrounding.

In applying this idea to swarm robotics, we can have robots in the systems to change their behaviour based on the signal and their own state, allowing them to repair other 'faulty' robots to allow the systems to continue achieving its task. For example, when a 'faulty' robot has been repaired by the other robot(s) in the systems, it must be able to change its current state from 'faulty' to 'non-faulty'. Once it has changed its own state, it is now capable of sharing its energy with other 'faulty' robots in the environment.

5.1.4 Design Principle IV

Agents can initiate a self-healing process dependant to their ability and location

During the process of granuloma formation, once the infected cell(s) emits a signal, other cells move to the sites of infections. Depending on their ability and location, if they are uninfected and able to reduce the infections, they do so. However, if they are also infected, they are not going to do anything but also emit signals to the environment. Thus, depending on their ability and location, the uninfected cells perform the self-healing process during the process of granuloma formation.

Taking this idea to swarm robotics, although we do not expect (or necessarily want) such a degree of independence in our AIS, we would like to allow robots in the system to initiate a self-healing process, depending on their ability; for example the robot(s) has enough energy to share with other 'faulty' robot(s) and their locations; whether they are

near or far from the faulty robot(s). This allows robots with enough energy who are near to a faulty robot(s) to charge the 'faulty' robot(s) before other robots with enough energy but too far from the 'faulty' robot(s) can share their energy.

Using our knowledge gained from the development of the model and simulation in chapter 4, and the subsequent derivation of the design principles outlined above, we now proceed with the discussion on the derivation of the granuloma formation algorithm for solving the *anchoring issue* in swarm beacon taxis in section 5.2. These design principles determine the local rules for each individual agent, in a principled way before the algorithm development. The goal is to make these systems adaptive and capable of changing its behaviour according to the changes of environmental circumstances, thereby ensuring the continued functionality of the system.

5.2 The Algorithm

We present here the granuloma formation algorithm that incorporates the ideas from the design principles explained in section 5.1. In this section we first present the granuloma formation algorithm and we then explain how each of the design principles is mapped to the algorithm in section 5.2.1.

As outlined in algorithm 5, in granuloma formation algorithm, robots are first deployed in the environment. Once deployed, the robots then perform the task that they need to do. During the movement, if there is any 'faulty' robot(s) in the environment, it will stop moving and emit signals that can be recognised by other 'non-faulty' robots within the pre-defined radius to the 'faulty' robot. These 'non-faulty' robots are then attracted towards the 'faulty' robot, akin to how T-cells are attracted by cytokines emitted by an infected macrophage in granuloma formation described in section 3.5. A limited number of these robots then isolate the 'faulty' robot, akin to T-cells surrounding an infected macrophage, again as described in section 3.5, and they perform repair or sharing of energy between themselves. The other robots which are not involved in isolating the 'faulty' robot ignore the failing and the robot(s) that surrounds it and treat them as if they were obstacles, in a manner similar to the standard ω-algorithm as described in section 2.3.3.

To illustrate the algorithm, figure 5.1 depicts each stages of the algorithm in accordance with the process of granuloma formation described in section 3.5. From figure 5.1, robots are first deployed in the environment(a) and start moving according to the algorithm or rule that has already been defined. In our case, the robots' movements is according to ω-algorithm (Bjerknes, 2009) as defined in section 2.3.3, which is a swarm moving towards a beacon. During the movement, each robot also checks its own status,

Algorithm 5: Overview of granuloma formation algorithm.

```
1 begin
2     Deployment: robots are deployed in the environment
3     repeat
4         Movement of robots in the environment according to ω-algorithm
          (Bjerknes, 2009)
5         Signal propagation: Faulty robots emit faulty signal, where the average
          radius of the target neighbouring robot is R
6         Protection and rescue: Healthy robots identify how many of them need to
          perform protection and rescue according to algorithm 7
7         Repair: Sharing of energy between faulty and healthy robots according to
          algorithm 6
8     until forever
9 end
```

as to whether it is in a 1)'faulty' state: a state where they are experiencing a large energy drain or 2) 'non-faulty' state: a state where it has enough energy to perform the task. If they are in a 'faulty' state, they stop moving, and remain halted in the environment, while other robots continue moving and avoid the 'faulty' robot like an obstacle in the environment (c). Once the robot has acknowledged that it is in a 'faulty' state, it starts to propagate faulty signals (d). This signal leads other 'non-faulty' robots to move towards the 'faulty' robot and surround it (e). Finally, once it has been surrounded by the 'non-faulty' robots, the 'non-faulty' robots performs the repairing or the healing process, which involves sharing energy between themselves (f).

In the real robot scenario, during the repairing phase, donor robots share their energy with the faulty robots. During this time, we can adopt the idea of the signalling mechanism in granuloma formation, as explained in section 5.1. In granuloma formation, different signals are emitted by macrophages to inform other cells of their current status. In the instantiation of the algorithm, we can use different coloured LED signals during the repairing phase. The green LED signals are emitted by the faulty and donor robots that are involved in this phase, notifying the other robots to ignore them and continue moving towards the beacon. This is illustrated in figure 5.2. Figure 5.2 shows the robotic agent signalling and interactions during the containment and repairing phase of granuloma formation algorithm. This is inspired by the signalling mechanism shown in figure 4.10. Figure 4.10 is map to figure 5.2 as explained in table 5.1.

5.2.1 Mapping the Design Principles to the Algorithm

In this section, we map the design principles stated in section 5.1 to the granuloma formation algorithm described in section 5.2. We highlight each of the design principles and

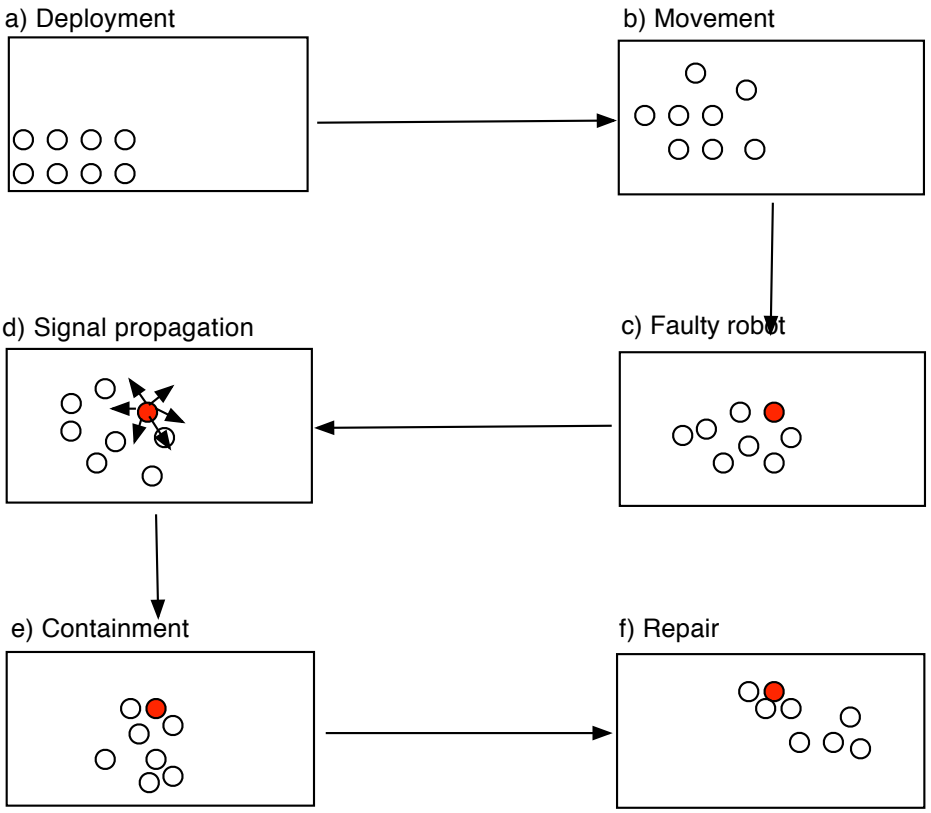


Figure 5.1: Simple Scenario: granuloma formation algorithm. In a), robots are deployed in the environment and moves or start performing their task (in b). The faulty robots stop moving and start to propagate signal as in c and d. The signals lead other macrophages to move towards the faulty robot and form an isolation between the faulty and non-faulty robots (in e). And finally the healing process between the faulty and non-faulty robots will start as in f.

Table 5.1: Mapping from granuloma formation signalling mechanism (in figure 4.10) to granuloma formation algorithm signalling and interactions (in figure 5.2).

Granuloma formation	Granuloma formation algorithm
Infected macrophage	Robot with low energy
Uninfected macrophage	Robot with enough energy
T-cell	Robot that is assigned share their energy with the robot with low energy
Chemokines signals	LED signals

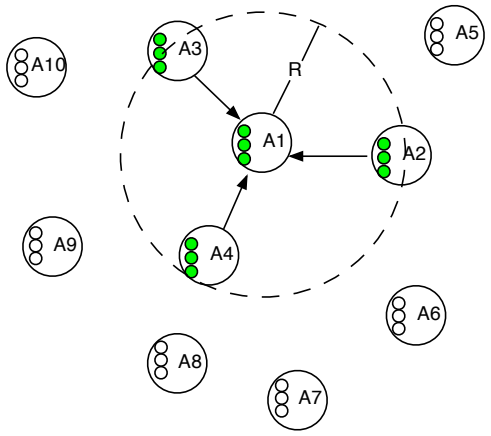


Figure 5.2: Robots agent signalling and interactions during the containment and repairing phase of granuloma formation algorithm. Robot that is involved in this phase emits the green LED signals notifying the other robots that they have low energy.

explain how it has affected the design of the granuloma formation algorithm.

As outlined in section 5.1, the first design principle states that *the communication between agents in the system is indirect consisting number of signals to facilitate coordination of agents*, which proposed to have robot(s) in the system sending and receiving different kind of signals, allowing the robots to communicate with each other. As outlined in algorithm 5, under the condition of a fault, the robots change their behaviour from 'non-faulty' to 'faulty'. During the 'faulty' stage, a robot begins to emit signals to the system. These signals become a means of communication between the robots in the system that facilitate the coordination between the robots, as discussed further below.

The second design principle states that *agents in the systems react to defined failure modes in a self-organising manner*, which is explained in section 5.1.2. This design principle proposes to have self-organised positive and negative feedback mechanisms that will regulate the level of attraction between the 'faulty' and 'non-faulty' robots in the system. In achieving this, as outlined in algorithm 6, once the 'faulty' robot emits the signal, this signal will regulate the level of attraction to the other 'non-faulty' robots in the systems to move towards the 'faulty' robot as well establishing communication between themselves. During this process, each robot evaluates its own position and identify their position as compared to the 'faulty' robot in this system as outlined in algorithm 6. As stated in the algorithm, during the communication process (in line 2 to 4), each robot evaluates its own energy and position and sends as well as receive their neighbours' energy and position within the pre-defied radius. Once a robot has received the information on their neighbours' energy, it then compares their neighbours' energy with the minimum energy that is identified as the energy threshold in the system. If the energy is less then

the energy threshold, it then assign a signature and stores the information on the energy and the position of their neighbours in an inbound queue as stated in line 5 to line 13 in algorithm 6. When an energy-low robot is found, they will branch to do algorithm 7 to identify the number of the neighbours that can share their energy. Therefore the inbound queue contains the information of a robot with low energy in the system considered as 'faulty'. Having stored the information on the faulty robot(s), each robot then evaluates its own energy by comparing with the threshold defined in the system and if their energy is more then the threshold, they are now is capable of sharing their energy and performing the healing process. They then evaluate their position in terms of whether they are the nearest robot to the 'faulty' robot; if they are the nearest, they move towards the 'faulty' robot. This is outlined in line 14 to 23 in algorithm 6.

Algorithm 6: Algorithm for containment and repair according to position of robots.

```
1  begin
2      Evaluate egy_self and pos_self
3      Send egy_self and pos_self to peers within R
4      Receive egy_peer and pos_peer from peers
5      forall egy_peer received do
6          if egy_peer < egy_min then
7              Store egy_peer in inbound queue
8          else
9              Add egy_peer to outbound queue
10         end
11     end
12     forall egy_peer in inbound queue do
13         Add signature
14         Store egy_peer in robot list
15         forall egy_peer in robot list do
16             if egy_self < egy_threshold then
17                 Do algorithm 7
18                 Evaluate pos_peer
19                 Sort pos_peer in ascending order
20                 Move to nearest pos_peer
21             end
22         end
23     end
24     forall egy_peer in outbound queue do
25         Delete signature
26     end
27 end
```

The basic terms used in the algorithm are as follow:

- pos_{self}: position of self robot
- pos_{peer}: distance of peer robots
- egy_{self}: energy of self robot
- egy_{peer}: relative energy of peer robots
- egy_{needed}: energy needed by failing robot
- $egy_{threshold}$: the limit of the energy that is needed

The third design principle in section 5.1.3 states that *agents must be able to learn and adapt by changing their role dynamically*. In adopting this design principle, we proposed that the number of functional robots that surround the faulty robots varies, and is not pre-defined. Thus, robots must be able to change their role dynamically during the communication phase, once the 'faulty' robot starts to emit faulty signals to the system. The number of robots required is determined by the amount of energy required to repair the failing robot, together with the location of the faulty robot. As outlined in algorithm 6, once the 'faulty' robot starts emitting the signal, communication is established between robots situated at the predefined radius of the 'faulty' robot. During this stage, its needed energy is evaluated, while feedback is received from the functional robot surrounding it one at a time. The 'faulty' robot evaluates each feedback from the functional robots containing the information on the energy that can be contributed by them. If the needed energy is enough, it stops emitting signals, and if the needed energy is not enough, it then continues emitting the signal until the needed energy is enough. An illustration of this process is shown in figure 5.3. From this figure, when there exists a faulty robot(s), it emits a signal, regulating some level of attraction to robots at a certain predefined radius (R). The robots that receive the signal communicate, exchange information on their locations and their current energy. This is also according to the energy transfer rules explain in algorithm 7 line 5 to line 11, where the nearest functional robots come alongside the robot, and share an amount of energy. If the energy that can be donated by the nearest functional robot is enough, the faulty robot stops sending information to the other robots. However, if the energy is not enough, the faulty robot continues to request energy from other functional robots in the environment. This process is repeated until enough energy information is obtained and then the isolating and sharing of energy take place.

The fourth design principle, which is described in section 5.1.4 states that *agents can initiate a self-healing process dependant to their ability and location*. This design principle is in line with the description of the algorithm 7 and algorithm 6 outlined and explained above, as well as the description of the mapping of second and the third design

Algorithm 7: Algorithm for containment and repair according to energy of robots.

begin
  Evaluate egy_{needed}
  Send egy_{needed} *to peers within* R
  Receive egy_{peer} *from peers within* R
  forall egy_{peer} *received* **do**
    if egy_{peer} *received* $< egy_{needed}$ **then**
      Send egy_{needed} *to peers within* R
    else
      Stop sending $egy_{needed}(t)$ *to peers within* R
    end
  end
end

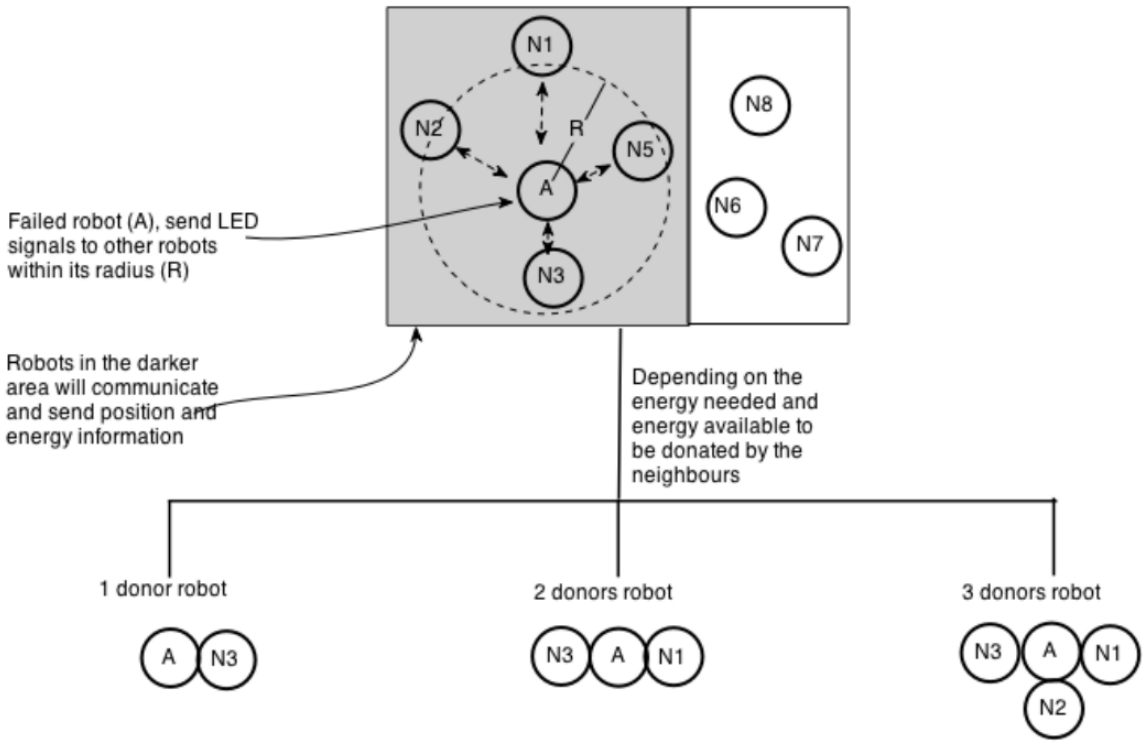


Figure 5.3: The process of signalling and communication between robots in the system.

principles. Thus, depending on the ability of the functional robot (in our case, the robot has enough energy for sharing as well continue operating), as well as their location are within the 'faulty' robot's radius, they can perform the healing process. If they are located within the radius but they do not have enough energy, they will not aid in the healing process and continue their previous operation. If for example there is no functional robots that have enough energy within the 'faulty' robot's predefined radius, the radius increases slightly, allowing other functional robots to establish communication with the 'faulty' robots.

5.3 Conclusions

In this chapter, we began by establishing that we were at a phase in the CFA where we could move towards producing the design principles and a granuloma formation algorithm for self-healing swarm robotic systems based on the biological properties of granuloma formation that were investigated in chapter 4. We first explained the design principles for swarm robotic systems in section 5.1, which are inspired by the UML model and agent-based simulation which are discussed in chapter 4. We then discussed on the development of novel AIS the granuloma formation algorithm in section 5.2, which is instantiated by the process of granuloma formation in immune systems. We also explained how each design principle that we established fits to the algorithm. In the next chapter, we will describe the series of experiments and results that we obtained from applying the granuloma formation algorithm in swarm beacon taxis. The algorithm is able to contain certain types of error and initiate repair strategies to allow energy sharing between robots, when there exist robots' energy failure in the systems. This experiment is performed in simulation, using the sensor-based simulation tool set, Player/Stage (Gerkey et al., 2003).

In summary, the granuloma formation algorithm proposed in this chapter has maintain the principles of swarm robotic systems as described below:

- no central control: in the algorithm, there is no central control or any human intervention that control the robots in the systems. The robots move according to the algorithm in performing their tasks. They change their functions in accordance to the specified rules in the algorithm. For example they change their functions from healthy to 'faulty' if the identify that their energy is getting lower and starts sending signals to the healthy robots. The healthy robots then perform the healing process or sharing of energy again according to the algorithm.

- simple agents: in swarm beacon taxis, the robots are only equipped with proximity sensors that help the robots to move towards the beacon, avoiding the obstacles

and aggregate together. They also can send infrared signals to the environment. Even though they only have certain capabilities, they can still aggregate together to achieve beacon as well as communicating to do the healing process.

- robustness and flexibility: the algorithm is meant to maintain the robustness of swarm robotic systems. In helping the 'faulty' robots to gain energy from the other healthy robots, the robustness of the systems can be achieved and the desired task can be performed. This is in line with the suggestion made by Şahin (2005) mentioning that the system must be able to function in the presence of partial failures or other abnormal conditions and they must be able to adapt to any changing requirements of the environment.

In applying the algorithm to real e-pucks robots, there are few extensions that need to be made. For example currently, the e-puck robots that we have do not have the power sharing capability. This is the main extension that need to be done in demonstrating the success of failure of the algorithm in real robots scenario. Other than the power sharing capability, the other mechanisms are already there (proximity sensors, signalling mechanisms, infrared sensors). Therefore, with power sharing mechanism designed in the e-puck robots the algorithm can be tested in real robots scenario. This will be part of the assessment of the practicality of the granuloma formation algorithm for swarm beacon taxis with tolerance to partial power failure.

CHAPTER 6

Experimental Methods and Results Analysis

In this chapter we describe the results obtained from the implementation of swarm beacon taxis algorithm, the ω-algorithm (Bjerknes, 2009) as described in chapter 2 in order to produce a baseline from which we can compare our proposed immune-inspired solution described in chapter 5. We first describe the experimental protocol for our simulation in Player/Stage in section 6.1. We then discuss the results obtained from the series of experiments that we have conducted for ω-algorithm, producing our baseline in section 6.2. This is followed by the result and analysis obtained from the single nearest charger algorithm, a trophallaxis algorithm proposed by Melhuish & Kubo (2007) for energy sharing in swarm robotic systems in section 6.3 and shared nearest charger algorithm, an extension of the trophallaxis algorithm in section 6.4. We finally explore the results from our proposed immune-inspired algorithm, the granuloma formation algorithm in section 6.5 and compare the results with the single and shared nearest charger algorithms discussed earlier. This chapter ends with section 6.6, in which we discuss on how the model and simulation described in chapter 4 has affected our immune-inspired algorithm as well as the potential drawbacks of the granuloma formation algorithm.

6.1 Experimental Protocol

The experiments presented in this section were performed in simulation using the sensor-based simulation tool set, Player/Stage (Gerkey et al., 2003)[1]. 10 e-puck robots are simulated with sized 0.12 m x 0.12 m, and equipped with 8 proximity sensors according to

[1]Player-Stage can be downloaded from http://playerstage.sourceforge.net/

the real e-puck robots' sensors; two at the front, two at the rear, two at the left and two at the right. Initially robots are randomly dispersed within a 2 m circle area with random headings. A robot polls its proximity sensors at frequency 5 Hz (1/T). Whenever one or more sensors are triggered the robot executes an avoidance behaviour, and turn away from the colliding robot or obstacles. The avoidance turn speed depends on which sensors are triggered and the robot keep turning for 1 second. The task of the swarm is to aggregate; movement together towards an infra-red beacon is an emergent property of the system. The environment is a 20 x 20 cm square arena with a beacon at position (4.0, 0.0) as shown in figure 6.1. The fixed parameters for the simulation are displayed in table 6.1. Each simulation run consisted of ten robots and repeated ten times. The analysis of why we choose ten simulation runs is presented in appendix A. The values in this table are obtained in accordance to the experiments conducted in Bjerknes (2009) with epuck robots. For each simulation run, the centroid position of the robots in the swarm was recorded. For the first experiment, we developed ω-algorithm (Bjerknes, 2009) in simulation to serve as our baseline behaviour for the remaining experiments.

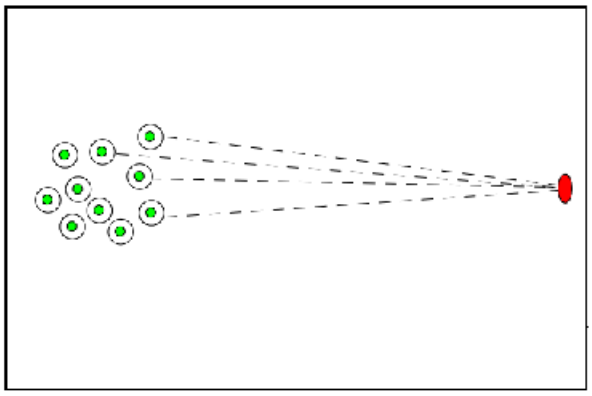


Figure 6.1: Snapshots of the simulation with Player/Stage with 10 robots and 1 beacon

In these experiments, we consider the failure mode of the failed motor, with the addition of the cause of motor failure being lack of power, but with the assumption that enough power remains for simple signalling, but not enough to power the motors. This assumption has been tested electronically using real e-puck robots with a simple obstacle avoidance task. This is to explore how the robots behave in low power situations, whether they could continue signalling even though they are no longer move. We performed a simple experiment with epuck robots where we allowed the robots to wander in an environment with a simple obstacle avoidance behaviour until the robots stopped moving. We monitored the power levels within the robots from this point (when the robots stopped

Table 6.1: Robot fixed parameters for all simulations according to experiments conducted in Bjerknes (2009) with epuck robots.

Parameter	Value
Time step duration	1 s
Robot normal speed	0.15 cm/s
Avoidance sensor range	0.4 cm
Robot body radius	0.12 cm
Robot's wireless range	2.0 cm
Minimum energy needed to move	500 j
Battery capacity	5000 j
Component fault	power drain
No of faulty robots	1 to 5 units
Simulation duration	1000 seconds

moving) and when the battery was totally discharged and the result is shown in table 6.2. We found that on average, the e-puck robots are able to send signals for 27 minutes before all the energy is lost. This failure mode allows us to construct a potential self-healing mechanism for the swarm, which involves a recharge of drained batteries.

Table 6.2: Results for difference in time between the robot's wheel and robot's led when the robot stops moving

Robots No	Difference in time between the robot's wheels stop moving and robot's led stop signalling
Robot 1	27 minutes 36 seconds
Robot 2	29 minutes 37 seconds
Robot 3	28 minutes 29 seconds
Robot 4	25 minutes 48 seconds

In the simulation, we introduce a power reduction failure to robots in the swarm: a single robot failure, two robot failures and three or more robot failures (until 5). When a single robot fails in the environment, it experienced a power reduction approximately at time=200 seconds in the simulation. With two failures, two robots are supplied with a power reduction simultaneously at t=200 seconds, whilst with three and more failing robots introduced simultaneously in the simulation. During this time, the robots are not moving and remain static in the environment. The parameter for the faulty scenario is shown in table 6.3.

Table 6.3: Variable parameters for failing scenario in the environment

Number of faults	Parameter	Time (s)
Single and multiple failure	Speed=0 m/s, energy=500 joule (J)	t=[200]

6.1.1 Simulation Platform

Player/Stage is an open source free to use software package released under a GNU Public License [2]. The software consists of *player*, a robot device interface and *stage*, a robot simulator. By combining these two parts a comprehensive simulation tool can be used; alternatively just the *player* part can be used to control physical robots. Player/Stage was the software of choice for the work undertaken by swarm robotics research (Nembrini et al., 2002; Nembrini, 2005; Liu et al., 2007). For a full list of publications, readers can refer to Player/Stage website [3].

Player provides the interface for many items of robot hardware. Operating as a robot server, the *player* software allows robot control software to be run on any machine with a network connection to the robot. Control software can be written in a variety of languages as long as it conforms with the interface. The interface should mean that simulated and real hardware is accessed in the same way; thus, the same control code should be able to control both real and simulated robots. *Stage* is a robot simulator producing an environment and robot models. Sensing data is simulated with results made available to control code just as real hardware would provide the data. Used in conjunction with the player software, *stage* represents how the real robots might act within a given environment. The most useful features in *stage* include footsteps, a history of the movement of the robot, and view of data, which allows a variety of different sensors to be displayed on the screen. Using view data, sensors can be selected to allow good customisation in terms of displaying options. Rangers can be shown as arcs coming from the robot showing how far a robot is from an object. The camera data can also be shown in the form of a blobfinder displaying the coloured blobs that the robot can see.

The lead developer of Player/Stage describes *stage* as ideal for massively multi-robot experiments, with particular emphasis given on the suitability of stage for swarm robotics (Vaughan, 2008). *Stage* is intended to allow migration of a controller from simulated to real robots (Vaughan & Gerkey, 2007). The strength of *player* is said to be its transparency, avoiding constraints being placed on the developer and providing a collection of easily accessible and commonly used devices (Vaughan & Gerkey, 2007). *Stage* is described as 'realistic enough' Vaughan (2008) for many purposes. However Vaughan (2008) points out some of the limitations of *stage*. *Stage* does not take dynamics into account, meaning actuation is not as it would be in real life. The camera data is only provided in blobfinder form and no bitmapped camera data is provided, meaning any vision based systems will be unrealistic. *Stage* also ignores sensor noise by simply relying on the low resolution ray-tracing used to construct models of robots and the arena to provide

[2] GNU General Public License: http://www.gnu.org/license/gpl.html

[3] Player/Stage publication list: http://playerstage.sourceforge.net/index.php?src=pubs

apparent noise. This means that whilst the simulation provided might give a good idea of general behaviour it may mask some potential errors based on variation in sensor readings and actuation.

6.1.2 Performance Metrics

In all the experiments, we measure the progression of the centroid of the swarm towards the beacon for every 100 seconds as per Bjerknes (2009), using equation 6.1; where *x* and *y* are the coordinates of the robots and *n* is the number of robots in the experiment.

$$\text{Centroid distance of robots to beacon} = \sum_{i=1}^{n} \frac{\sqrt{(x_{1_i} - x_{2_i})^2 + (y_{1_i} - y_{2_i})^2}}{n} \tag{6.1}$$

6.1.3 Significance Test

During the experiments, we use the non-parametric effect magnitude test (Vargha & Delaney, 2000). Non-parametric statistics are those data that do not assume a prior distribution. There are several advantages of using nonparametric statistics, as explained in Vargha & Delaney (2000). As can be expected, since there are fewer assumptions made about the sample being studied, non-parametric statistics are usually wider in scope as compared to parametric statistics that actually assume a distribution. This is mainly the case when we do not know a lot about the sample we are studying and making a priori assumptions about data distributions that might not give us accurate results and interpretations.

6.2 Experiment I: ω-algorithm

$H1_0$: *The ω-algorithm (M1) for swarm beacon taxis allows the swarm to achieve a centroid distance less than 0.5 cm away from the beacon when there are no failures introduced.*

$H2_0$: *The ω-algorithm (M1) for swarm beacon taxis allows all robots in the swarm to achieve a centroid distance less than 0.5 cm away from the beacon with one failing robot.*

$H3_0$: *The ω-algorithm (M1) for swarm beacon taxis allows all robots in the swarm to achieve a centroid distance less than 0.5 cm away from the beacon with more than two failing robots.*

6.2.1 Description

The following section investigates ω-algorithm (Bjerknes, 2009) in Player/Stage simulation that will serve as a baseline from which we calibrate our new proposed immune-inspired algorithm against.

6.2.2 Results and Analysis

Our experiments start with an investigation of the ω-algorithm for swarm beacon taxis, in effect $H1_0$, as developed by Bjerknes (2009). The swarm starts in one part of the arena and moves toward the beacon as the results of the emergence behaviour of the algorithm. The distance from the centroid of the swarm to the beacon for each run is given in figure 6.2. For each experiment the robots have a different starting position in the arena, but as the importance here is on the relative performance between different sets of runs, the starting point was set to 35 cm from the beacon. This allows for a comparison between each run. The hypotheses can be accepted if the swarm reaches a distance of less than 0.5 cm from the beacon. Based on the experiments, the swarm has a mean velocity of 1.2 cm/simulation seconds. The slowest moved at 1.52 cm/simulation seconds and the fastest had the velocity of 1.01 cm/simulation seconds. At time t=600 seconds, the swarm has reached 0.5 cm away from the beacon. From the simulation results, the experiment fails to reject hypothesis $H1_0$ at the default of α= 0.05 significance level, which is indicated by the *p* value = 0.4830, which is much greater than the α. The 95% confidence interval on the mean centroid distance of robots to the beacon less than 0.5 *cm* is obtained from this experiment.

We then introduce a partial failure to an individual robot in the swarm, given the scenario outlined above of partial power failure which is enough to stop the robot moving. This experiment is to test $H2_0$. As mentioned by Bjerknes (2009), the influence from this fail state is small and the failed robot will be avoided as if it is an obstacle in the environment, and the swarm will continue towards the beacon. However, the swarm experiences a temporary slow down as it is attempting to avoid the obstructing robot, then it will pick up its normal velocity once the failed robot has been avoided. The mean distance over time, across the ten runs, is shown in figure 6.3. The swarm has the mean velocity of 1.32 *cm/sec*, where the fastest velocity in the experiment is 1.65 *cm/sec* when the swarm is trying to free itself from the failed robot. During this scenario, the faulty robot does not move and remains static. The experiment fails to reject hypothesis $H2_0$ at the default of α= 0.05 significance level, which is indicated by the *p* value = 0.6442 that is much greater than the α. The 95% confidence interval on the mean centroid distance of robots to the beacon less than 0.5 *cm* is obtained from this experiment.

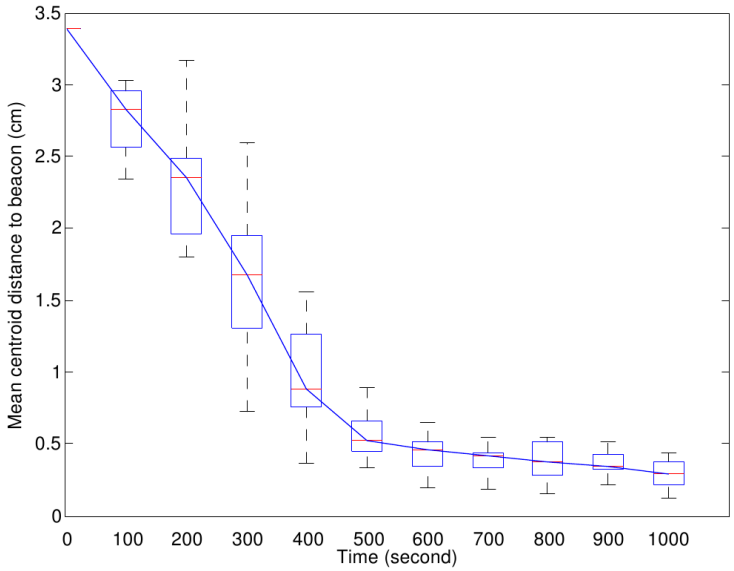


Figure 6.2: Boxplots of the mean distance between swarm centroid and beacon as a function of time for 10 experiments using ω-algorithm with no robot failure for $H1_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and beacon for each boxplot. At approximately t=650 seconds, the swarm has reached the beacon.

In the third set of experiments two and then three failing robots are introduced to the simulation. This experiment tests $H3_0$. Figure 6.4 and 6.5 show the results of multiple robot failure in swarm beacon taxis. As mentioned by Bjerknes (2009), the influence from these two failing robots is small and the failing robots will again be avoided as if they were obstacles in the environment, and the swarm will continue towards the beacon. The result obtained from the experiments fail to reject hypothesis $H3_0$, when there is two failing robots in the experiment at the default of α= 0.05 significance level, which is indicated by the p value = 0.5372 is much greater than the α. The 95% confidence interval on the mean centroid distance of robots to the beacon less than 0.5 *cm* is obtained from this experiment.

However, as more failing robots are introduced into the experiment, the swarm stops moving towards the beacon and the faulty robots will 'anchor' the robots in the swarm and they will never reach the beacon as shown in figure 6.5. From these experiments, the anchoring problem manifest as three failing robots are introduced to the swarm. With three failing robots, the experiment results allow us to rejects the hypothesis $H3_0$ at the default of α= 0.05 significance level, which is indicated by the p value = 3.6926e-14 that has fallen below α. The 95% confidence interval on the mean centroid distance of robots to the beacon is more than 0.5 *cm* in this experiment.

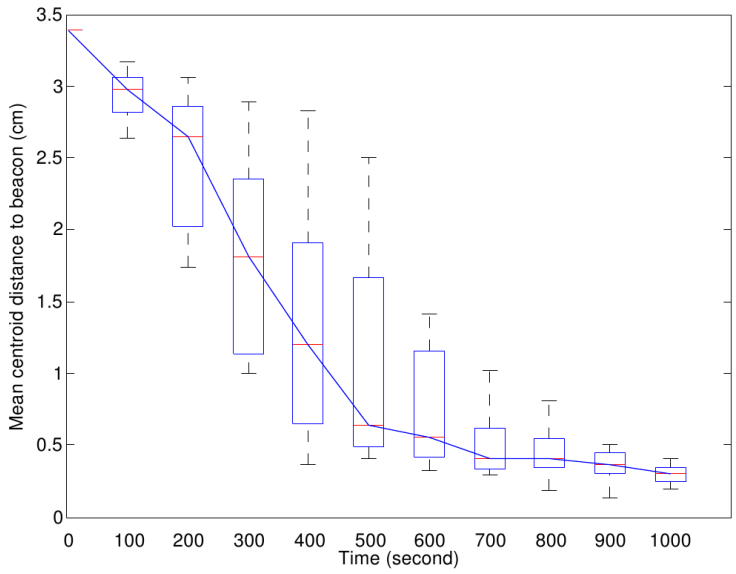


Figure 6.3: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using ω-algorithm with one robot fails at t=200 seconds for $H2_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and beacon for each boxplot. The swarm reaches the beacon at approximately t=850 seconds.

Results from the experiments show that, even with two completely failed robots the swarm always reach the beacon, and the delay is once again relatively small. However, as three faulty robots are introduced into the simulation, the swarm starts to stop moving, not reaching the beacon and stagnating around the three failing robots as shown in figure 6.5. From the experiments that have been conducted, we have confirmed the observations of (Bjerknes, 2009) that the anchoring issue, occur in the swarm beacon taxis as more failures are injected in the systems. The failing robot will 'anchor' the swarm, impeding its movement towards the beacon.

As we have reproduced the effect reported by Bjerknes (2009), we propose a number of potential solutions that might potentially mitigate the effect of the observed anchoring issues and evaluate their performances when compared with each other. These approaches are: the single nearest charger algorithm, an algorithm which is inspired by the idea of trophallaxis proposed by Melhuish & Kubo (2007), which we describe in section 6.3 and shared nearest charger algorithm, an algorithm which is an extension from the idea of trophallaxis (Melhuish & Kubo, 2007). This algorithm is explained in section 6.4.

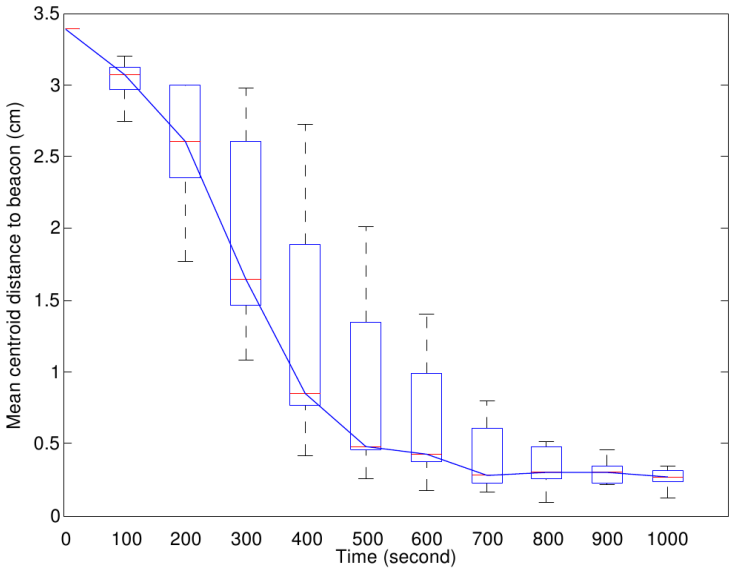


Figure 6.4: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using ω-algorithm with two robot fails simultaneously at t=200 seconds for $H3_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st}quartile. The solid line shows the median distance between swarm centroid and beacon for each boxplot. The swarm reaches the beacon approximately at t=1000 seconds.

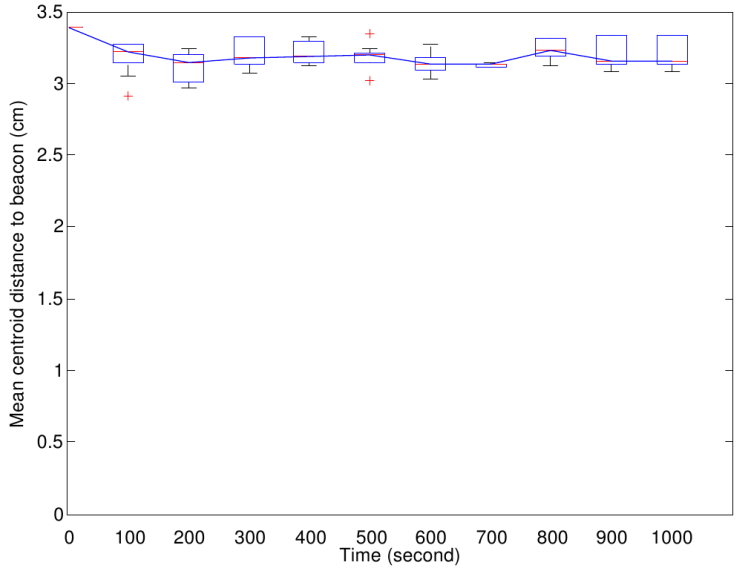


Figure 6.5: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using ω-algorithm with three robot fails simultaneously at t=200 seconds for $H3_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and beacon for each boxplot. The swarm does not reach the beacon as the failing robots anchor the swarm.

6.3 Experiment II: Single Nearest Charger Algorithm

In this section, we investigate the effect of single nearest charger algorithm for swarm beacon taxis. We first describe the single nearest charger algorithm in section 6.3.1 and discuss the results obtained from the experiments comparing the performances of the single nearest charger algorithm with ω-algorithm in section 6.3.2. The hypothesis for the experiments is:

$H4_0$: *The use of single nearest charger algorithm (M2) for swarm beacon taxis does not improve the ability of the robots in the systems to achieve a centroid distance less than 0.5 cm away from the beacon when compared with ω-algorithm (M1) with one or more failing robot(s) in the system.*

6.3.1 Description

As discussed by Melhuish & Kubo (2007), truly autonomous systems will required to be able to generate and manage their own energy. Therefore, robots must be able to distribute the collective energy resources owned by the group member in the environment. This is taken from the phenomenon of food sharing or *trophallaxis* observed in the world of social insect such as ants. Taking this idea in solving the 'anchoring' issue in ω-algorithm, we apply the energy sharing mechanism proposed by Melhuish & Kubo (2007) between robots in the simulation. We also adopt the energy transfer rule obtained from Melhuish & Kubo (2007). The rules are:

1. the energy transfer is limited to only two robots: each robot in the simulation can only receive or donate energy from one robot at a time.
2. the energy transfer begins when two robots collide, one of them requesting energy (recipient) from the other robot (donor).
3. the donor cannot reject the request (if it has enough energy) and must donate an amount energy defined by the the following rule:
    - if the donor has energy above the energy threshold, the donor will send a fixed proportional amount of energy to the recipient.
    - if the donor has energy below the energy threshold, the donor may not donate the energy to the recipient.

In our experiment, we set the energy threshold = 1500 J. This means that each donor robots must have at least 1500 J of energy before they can donate some of their energy.

If their energy is below this threshold, they cannot donate their energy and leave the recipient, allowing other robot that have a higher energy to donate some of their energy to the recipient. This help to preserve the donor's energy, allowing them to save some of their energy to complete their own task. This single nearest charger algorithm is outlined in algorithm 8.

Algorithm 8: Overview of single nearest charger algorithm

```
1  begin
2      Deployment: robots are deployed in the environment
3      repeat
4          Random movement of the robot in the environment
5          Signal propagation: Faulty robots emit distress signal
6          Rescue: One of the healthy robots with the nearest distance (earliest
           arriving robot) perform protection and rescue
7          Repair: Sharing of energy between faulty and healthy robots according to
           algorithm 9
8      until forever
9  end
```

Algorithm 9: Algorithm for containment and repair for single nearest charger algorithm

```
1  begin
2      Evaluate pos_self(t)
3      Send pos_self(t) to peers
4      Receive egy_peer(t − x) and pos_self(t − x) from peers
5      forall egy_peer(t − x) do
6          Evaluate egy_peer(t − x)
7          if egy_peer(t − x) < egy_min then
8              Evaluate pos_peer(t − x)
9              Sort pos_peer(t − x)in ascending order
10             Move to nearest pos_peer(t − x)
11         else
12             Do not move to pos_peer(t − x)
13         end
14     end
15 end
```

Algorithm 9 reflects the overview of the single nearest charger algorithm to do the repair for the faulty robot(s). The basic terms used in the algorithm are outline below:

- $pos_{self}(t)$: position of the current robot
- $pos_{peer}(t-x)$: position of peer robots

- $egy_{self}(t)$: energy of the current robot
- $egy_{peer}(t-x)$: energy of peer robots
- egy_{min}: minimum energy required
- egy_{needed}: energy needed by each robot

6.3.2 Results and Analysis

As with our baseline experiments (M1) in section 6.2, M2 is tested with the same scenarios and results are compared to M1. We look at both statistical and scientific significance in comparing the results of the experiments. Statistical significance is measured with ranksum test and scientific significance with A measure (Vargha & Delaney, 2000). A *p* value of 0.05 for ranksum test is commonly used to signify that two samples are different with the statistical significance because the medians are different at a 95% confidence level. The interpretation of A value is such that:

- a value around 0.5=*no effect*;
- a value around 0.56=*small effect*;
- a value around 0.64=*medium effect*;
- a value around 0.71=*big effect*.

We first introduce one and two failing robots to the algorithm and from the results obtained, M2 does not differ greatly from M1. The results are shown in figure 6.6 and figure 6.7. Based on figure 6.6 and figure 6.7, M2 works well with one and two failing robots in the system, where the swarm can achieve less than 0.5 *cm* of the beacon. From the results, the experiments fail to reject hypothesis $H4_0$ at the default of α= 0.05 significance level, indicated by the *p* value=0.54 with one failing robot and *p* value=0.8150 with two failing robots, which is much greater than the α. The 95% confidence interval on the mean centroid distance of robots to the beacon of less than 0.5 *cm* is obtained from the experiments. However, as compared to M1, in M2 all robots are able to arrive at the beacon but in M1, even though the swarm can reach the beacon with one and two failing robots, they leave the failing robots behind and avoid them as obstacles. This is consistent with the observations by Bjerknes (2009) and Bjerknes & Winfield (2010) and our results in section 6.2.2.

We then continue by introducing three and four failing robots to the system. The results show that with three and four failing robots, the swarm fails to achieve less than

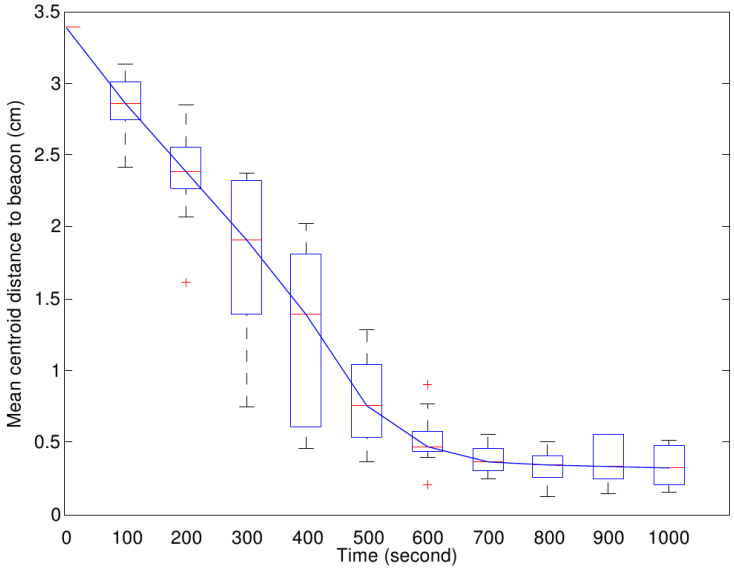


Figure 6.6: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using single charger algorithm with one robot fails for $H4_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and the beacon for each boxplot for 10 experiments. With one faulty robot, the swarm reaches the beacon approximately at t=650 seconds.

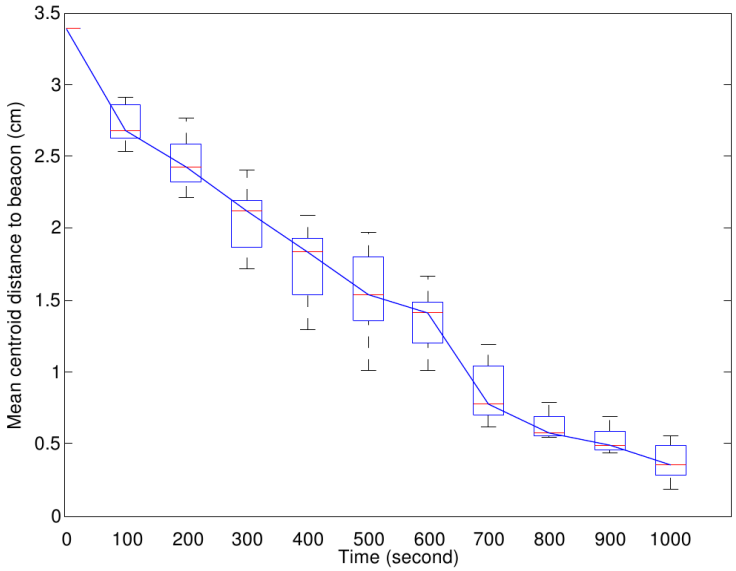


Figure 6.7: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using single charger algorithm with two robots fail for $H4_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and the beacon for each boxplot for 10 experiments. The swarm reaches the beacon at approximately t=900 seconds.

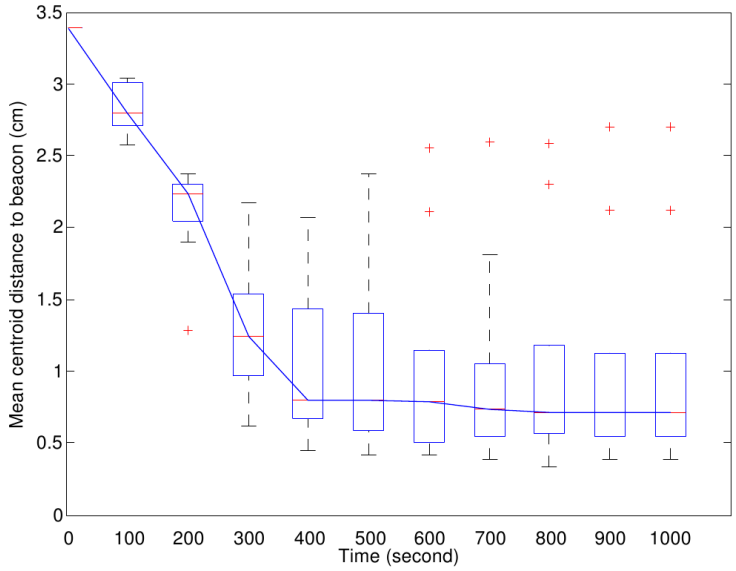


Figure 6.8: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using single charger algorithm with three robot fails for $H4_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and the beacon for each boxplot. The swarm does not reach the beacon as the failing robots 'anchor' the swarm to moves towards the beacon.

0.5 *cm* away from the beacon but can still achieve less than 1.0 *cm* away from the beacon. The swarm does not reach the beacon, as the failing robots 'anchor' the swarm to moves towards the beacon. These are shown in figure 6.8 and figure 6.9. With three and four failing robots in the system, the experiment rejects the hypothesis $H4_0$ at the default of α= 0.05 significance level, which is indicated by the p value = 1.7661e-04 and p value = 1.7562e-04 that have fallen below the α. From these results, M2 performs significantly better from M1 when there are three and four failing robots in the system.

The P and A values for mean centroid distance on robots for M1 and M2 are shown in table 6.4. From these values, both algorithms do not differ statistically with one and two failing robots are introduced in the system. However, when three and four failing robots are introduced to the system, M2 performs significantly better than M1. However, as the number of failures reaches 5 (half of the swarm failed) both methods have a small effect on the result. Based on the experiments conducted, when half of the swarm fails, robots are not able to reach the beacon and the swarm stagnates around the failing robots. For five failing robots M2 does not gives a better performance if compared with M1 as indicated by the p value = 0.3033.

In the experiments, the robots start with same levels of energy which is 5000 J. In each failing case, the energy is reduced to 500 J. The threshold value for the failing robots

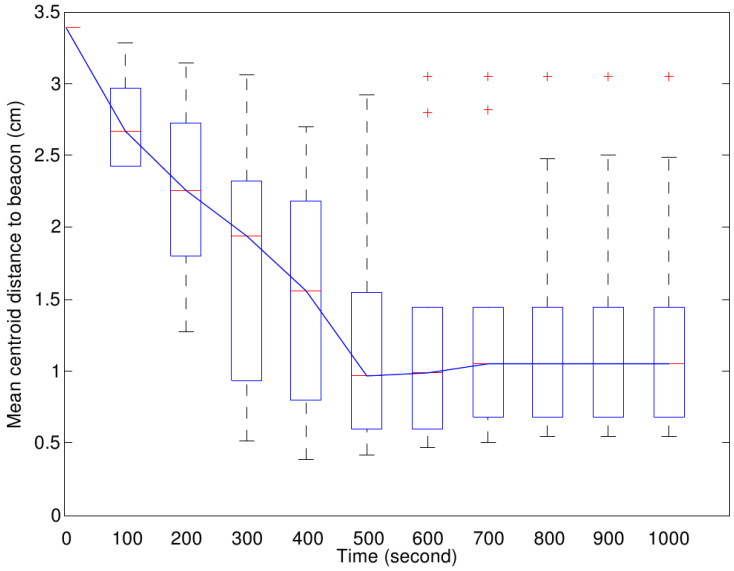


Figure 6.9: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using single charger algorithm with four robot fails for $H4_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between the swarm centroid and the beacon for each boxplot. The swarm does not reach the beacon as the failing robots 'anchor' the swarm to move towards the beacon.

Table 6.4: P and A values for mean centroid distance on robots between M1 and M2 in 1100 simulation seconds.

M2/M1	**ranksum P**	**A measure**
One fail	0.5400	-
Two fails	0.8150	-
Three fails	*1.7661e-04*	1
Four fails	*1.7562e-04*	1
Five fails	0.3033	-

to start triggering a distress signal is 500 J. Figure 6.10, 6.11 and 6.12 show the energy of each robot during 1000 simulation seconds with two, three and four failing robots in the system. In figure 6.11, three failing robots are introduced into the systems. When there are three failing robots in the system, half of the swarm reach the minimum energy threshold, which is 500 J approximately at t = 900 seconds. From figure 6.12, when there are four failing robots, almost all robots in the swarm start to reach the minimum energy threshold during 900 simulation seconds. This trend continues for five failing robots in the environment, where all the robots have energy less than 500 J approximately at t = 900 seconds.

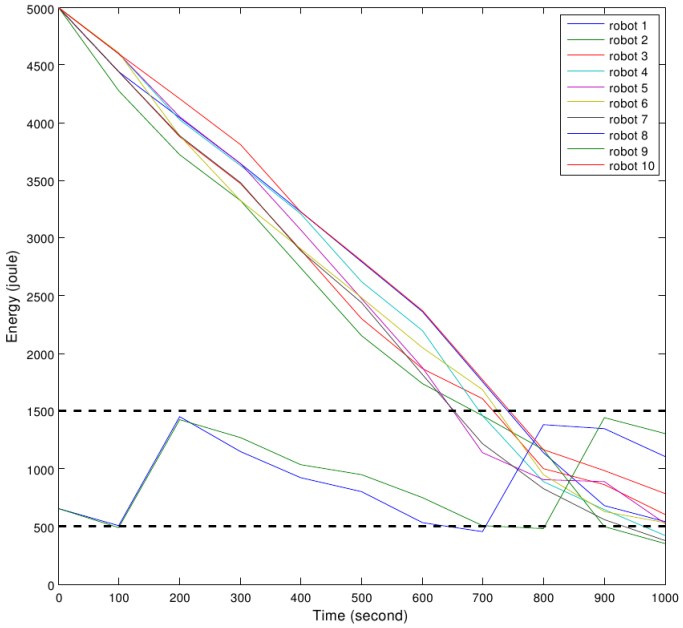


Figure 6.10: The energy of 10 robots with 2 robots fail using single nearest charger algorithm. The energy of half of the swarm has reached the minimum energy threshold during 900 simulation seconds

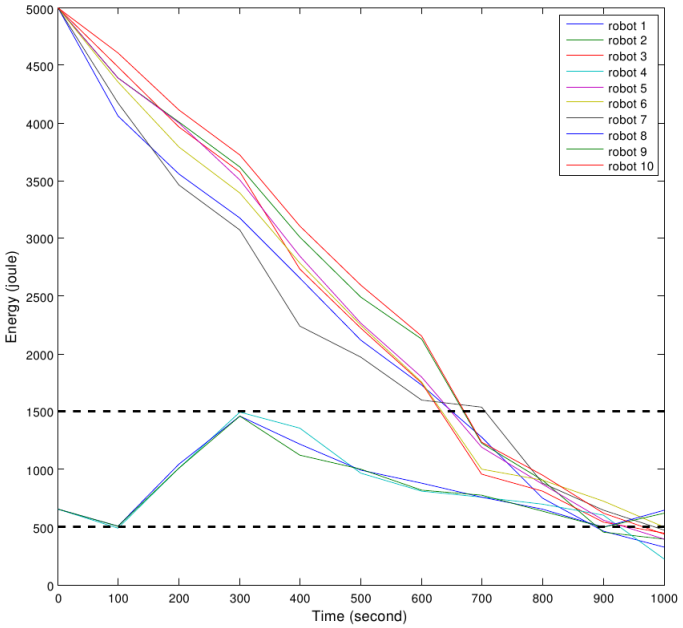


Figure 6.11: The energy of 10 robots with 3 robots fail using single nearest charger algorithm. The energy of the swarm has reached the minimum energy threshold during 900 simulation

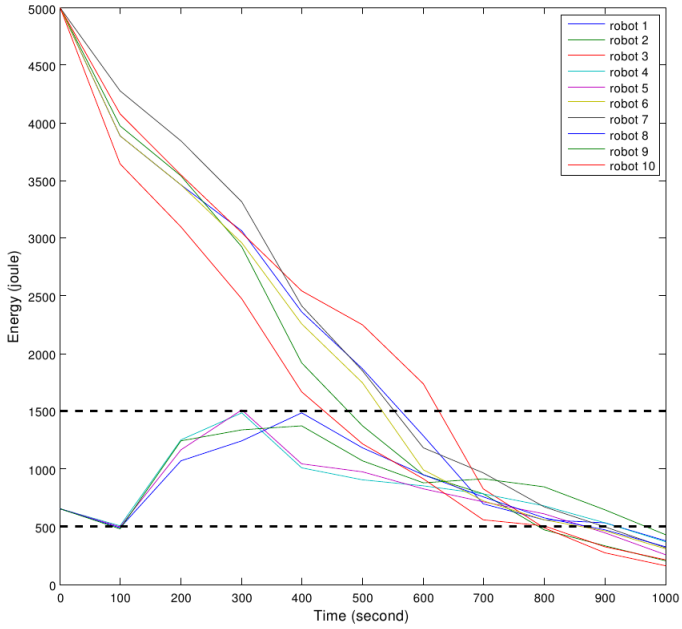


Figure 6.12: The energy of 10 robots with 4 robots fail using single nearest charger algorithm. The energy of the swarm has reached the minimum energy threshold during 900 simulation

In summary, the results of the experiments show that M2 performs slightly better than M1 but the swarm still cannot reach the beacon if the number of failing robots is more than three. During the experiments, we also observe that the nearest charging robot may not have sufficient energy to donate to the faulty robots, leading to the failure of the charging robots. This method would also be applicable if the number of failing unit is small (less than half of the swarm fails). However as more robots fail, they then act as 'anchor' the swarm leading to impeding the swarm to move towards the beacon. In conclusion, although M2 managed to achieve a desirable solution when there are one and two failing robots in the system, it still cannot tolerate with the presence of three and more failing robots in the system. Thus, the next set of experiments in section 6.4 investigates an enhanced method and introduces the shared nearest charger algorithm in solving 'anchoring' issue for swarm beacon taxis.

6.4 Experiment III: Shared Nearest Charger Algorithm

In this section, we investigate the effect of the shared nearest charger algorithm for swarm beacon taxis. We first describe the algorithm in section 6.4.1 and discuss the results obtained from the experiments comparing the performances of shared nearest charger algorithm with single charger algorithm and ω-algorithm in section 6.4.2. The hypothesis

for the experiments is:

$H5_0$: *The use of a shared nearest charger algorithm (M3) does not improve the ability of the robots in the systems to achieve a centroid distance less than 0.5 cm away from the beacon when compared to ω-algorithm (M1) and single nearest charging algorithm (M2) with one or more failing robot(s) in the system.*

6.4.1 Description

Based on the idea of the single nearest charger algorithm explained in section 6.3.1, we extend the algorithm by increasing the number of donors to each faulty robot. The extension is based on our evaluation done in section 6.3.1, where we believe that by increasing the number of donors assist the recharging process when there is faulty robot(s) in the system. This algorithm is considered as a novel contribution prior to the development of the immune-inspired algorithm described in chapter 5 The general algorithm of shared nearest charger algorithm is presented in algorithm 10, which is also illustrated in figure 6.13.

Algorithm 10: Overview of Shared Nearest Charger Algorithm

```
1 begin
2     Deployment: robots are deployed in the environment
3     repeat
4         Random movement of the robot in the environment
5         Signal propagation: Faulty robots emit distress signal
6         Rescue: n number of healthy robots with the nearest distance (earliest
          arriving robot) and highest energy perform protection and rescue according
          to algorithm 11
7         Repair: Sharing of energy between faulty and healthy robots according to
          algorithm 11
8     until forever
9 end
```

From figure 6.13, we can see that three donor robots share their energy with the failed robot. In this algorithm, the robots can transfer the minimum amount of energy from each of the neighbouring/charging robots taking into account that priority is given to the robots with higher energy and near to the failing robots. In the algorithm, we limit the energy transfer between three robots, which means that each faulty robot can only receive energy simultaneously from the three nearest robots. We choose three donors as we believe that it is possible to have two or three donor robots in the system but it will be too many for four or more donor robots in the system. In real e-puck robots, it is possible to have one,

two or three refuelling interactions but having more than three will add complication to the interactions between robots in the systems. Based on this reason, we choose to have only three donor robots for this algorithm. Depending on the needs and ability of the robot(s), the donor robots can be increased or decreased accordingly. The donor robots in shared nearest charger algorithm must donate the amount of energy defined by the energy transfer rule in algorithm 11, which are described as follows:

1. the energy transfer is limited to only four robots (including the faulty robot): each robot in the simulation can only receive energy from three robots at a time, but can only donate energy to one robot at a time.

2. the energy transfer begins when two robots collide, one of them requesting energy (recipient) from the other robot(s) (donor).

3. the donor cannot reject the request (if it has enough energy) and must donate an amount energy defined by the the following rule:

   - if the donor has energy above the energy threshold, the donor send a fixed proportional amount of energy, which is one third of the needed energy to the recipient.
   - if the donor has energy below the energy threshold, the donor may not donate the energy to the recipient.

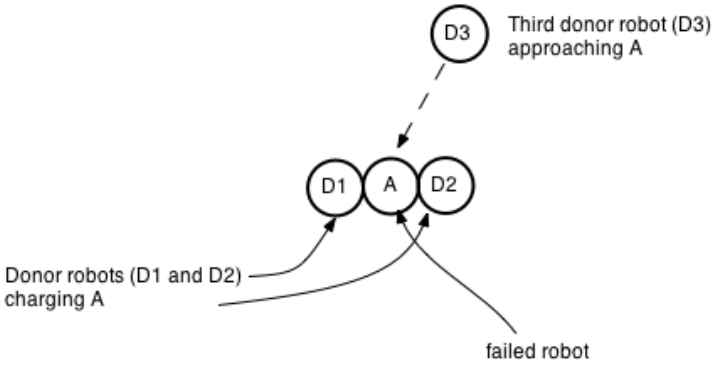


Figure 6.13: The illustration of shared nearest charger algorithm.

The basic terms used in the algorithm are as follow:

- $pos_{self}(t)$: position of current robot.
- $pos_{peer}(t-x)$: position of peer robots.

Algorithm 11: Algorithm for containment and repair according to energy and position of robots.

```
1  begin
2      Evaluate pos_self(t)
3      Send pos_self(t) to peers
4      Receive egy_peer(t − x)/ and pos_self(t − x) from peers
5      forall egy_peer(t − x) do
6          Evaluate egy_needed(t)
7          Divide egy_needed(t) with n
8          Send egy_needed(t) to peers
9          if egy_peer(t − x) < egy_min then
10             Evaluate pos_peer(t − x)
11             Sort pos_peer(t − x)in ascending order
12             Move to nearest pos_peer(t − x)
13         else
14             Do not move to pos_peer(t − x)
15         end
16     end
17 end
```

- $egy_{self}(t)$: energy of current robot.
- $egy_{peer}(t-x)$: energy of peer robots.
- egy_{min}: minimum energy required.
- egy_{needed}: energy needed by each robot.
- n: number of donor robot, in this experiment n = 3.

6.4.2 Results and Analysis

As with previous experiments in section 6.2 and section 6.3, M3 is tested in the same scenarios and the results are compared with M2 and M1. We look at both the statistical and scientific significance. Statistical significance is measured with ranksum test and scientific significance with A measure (Vargha & Delaney, 2000).

We begin the experiment by introducing one and two failing robots in the system. With one and two failing robots, the results of M3 do not differ greatly from M2 and M1 as in all methods the swarm can still reach the beacon if there is only one or two failing robots in the system. The result for M3 with one and two failing robots in the system is shown in figure 6.14 and figure 6.15. Based on figure 6.14 and figure 6.15, M3 works well with one and two failing robots in the systems where the swarm can reach the beacon. From the results, the experiments fail to reject hypothesis $H5_0$ if compared with M1 at

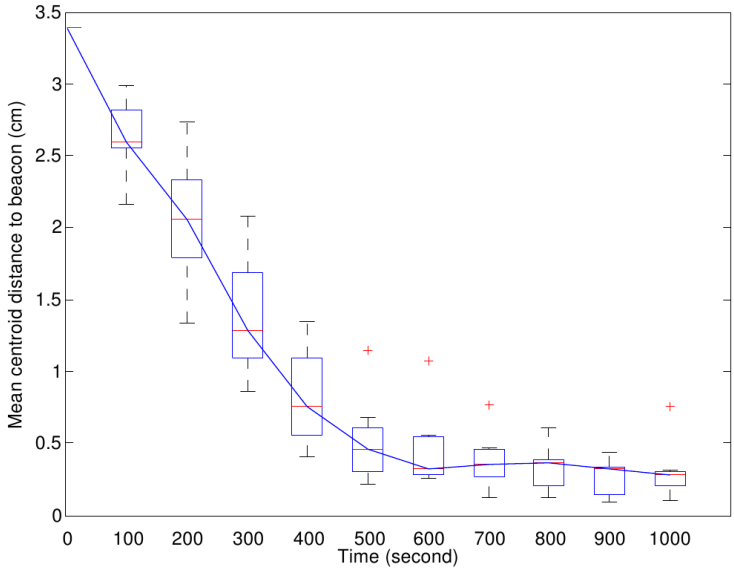


Figure 6.14: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using shared nearest charger algorithm with one robot fails for $H5_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and beacon for each boxplot for 10 experiments. With one failing robot in the system the swarm is able to reach the beacon.

the default of α= 0.05 significance level, indicated by the *p* value=0.65 with one failing robot and *p* value=0.57 with two failing robots, which is much greater than the α. The experiments again fail to reject hypothesis $H5_0$ if compared with M2, indicated by the *p* value=0.3631 with one failing robot and *p* value=0.2404 with two failing robots, which is much greater than the α. The 95% confidence interval on the mean centroid distance of robots to the beacon less than 0.5 *cm* is obtained from the experiments.

However, M3 starts to suffer when the failures increases. With three, four and five failing robots in the system, the swarm fails to reach the beacon. These are shown in figure 6.16, figure 6.17 and figure 6.18. As depicted in these figures, the swarm can only reach around 1 to 1.5 *cm* away from the beacon with three, four and five failing robots in the system. Based on the results, the experiments fail to reject hypothesis $H5_0$ if compared with M2 at the default of α= 0.05 significance level, indicated by the *p* value=0.4261 with three failing robot, a *p* value=0.0930 with four failing robots and a *p* value=0.8785 with five failing robots, which is much greater than the α. However, the experiments reject hypothesis $H5_0$ if compared with M1 with three and four failing robots in the system. This is indicated by the *p* value=1.7462-e04 with three failing robots and *p* value=1.0650e-04 with four failing robots, which is much greater than the α. However, as the number of failures reaches five (half of the swarm), it again fails to reject $H5_0$ as both methods show

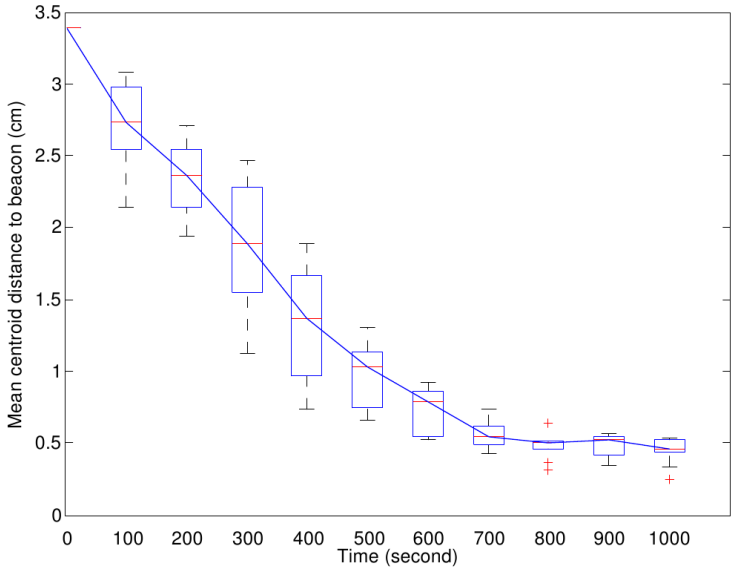


Figure 6.15: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using shared nearest charger algorithm with two robots fail for $H5_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and beacon for each boxplot. With two failing robots in the system the swarm is able to reach the beacon.

Table 6.5: P and A values for mean centroid distance on robots between M3 and M2 in 1100 simulation seconds.

M3/M2	**ranksum P**	**A measure**
One fail	0.3631	-
Two fails	0.2404	-
Three fails	0.4261	-
Four fails	0.0930	-
Five fails	0.8785	-

no difference in performance, with a p value=0.4226.

The P and A values for mean centroid distance on robots comparing M3 and M2 are shown in table 6.5 and comparing M3 and M1 are shown in table 6.6. From these values, M3 does not differ statistically with M2 when failing robots are introduced in the system. When we compared M3 with M1, even though it does not differ statistically when one, two and five failing robots are introduced into the system, it does give a better performance when there are three and four failing robots in the system.

As in previous experiments, all robots start with equal energy which is 5000 J. Figure 6.19, 6.20, 6.21 and 6.22 explains the energy of 10 robots during 1000 simulation seconds with two, three and four failing robots in the system with the shared nearest charging

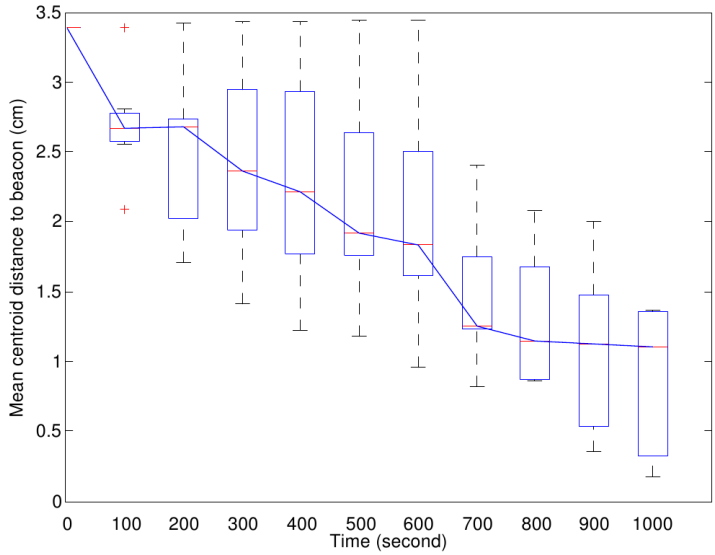


Figure 6.16: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using shared nearest charger algorithm with three robot fails for $H5_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and beacon for each boxplot. With three failing robots in the system the swarm is unable to reach the beacon.

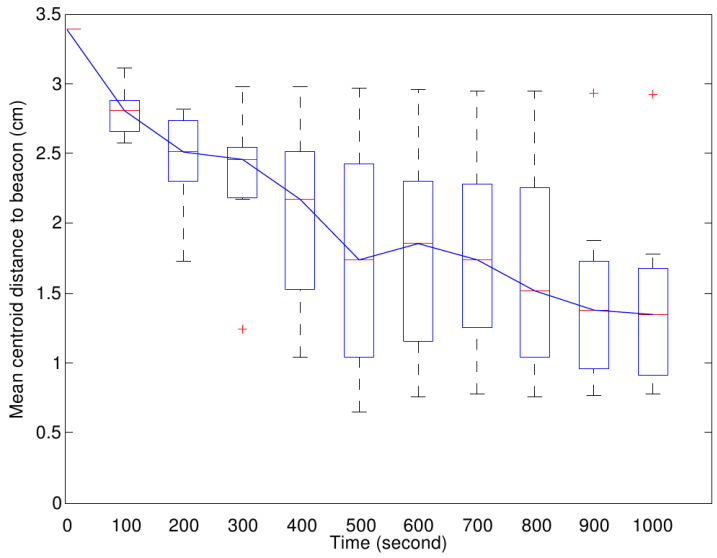


Figure 6.17: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using shared nearest charger algorithm with four robot fails. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and beacon for each boxplot. With four failing robots in the system the swarm is unable to reach the beacon.

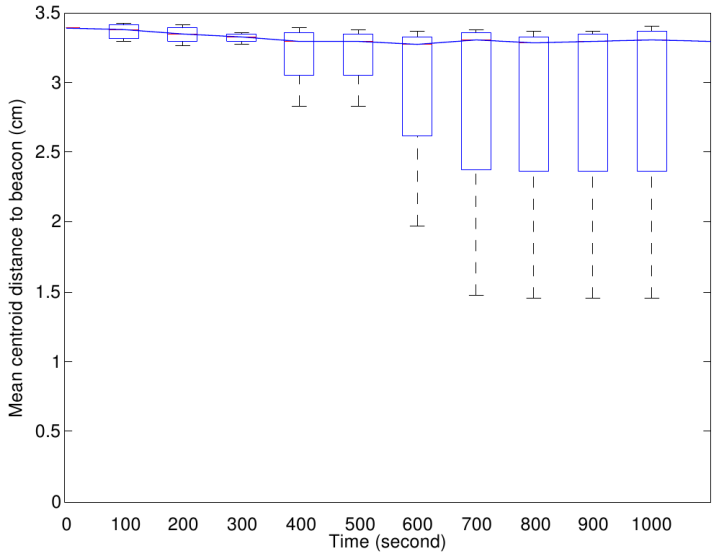


Figure 6.18: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using shared nearest granuloma formation algorithm with five robots fail for $H5_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The solid line shows the median distance between swarm centroid and beacon for each boxplot for 10 experiments. With five failing robots in the system the swarm is unable to reach the beacon.

Table 6.6: P and A value for mean centroid distance on robots between M3 and M1 in 1100 simulation seconds.

M3/M1	**ranksum P**	**A measure**
One fail	0.650	-
Two fails	0.57	-
Three fails	*1.7462e-04*	1
Four fails	*1.0650e-04*	1
Five fails	0.4226	-

algorithm. For two and three robot failures, the robots do not suffer from minimum energy level. The charging robots have enough energy to recharge the faulty robots and maintain a significant amount of energy to move towards the beacon. However, with four failing robots in the system, the robots start to lose a significant amount of energy and towards the end of the simulation, half of the swarm has energy less than the threshold value, which is 500 J as shown in figure 6.21.

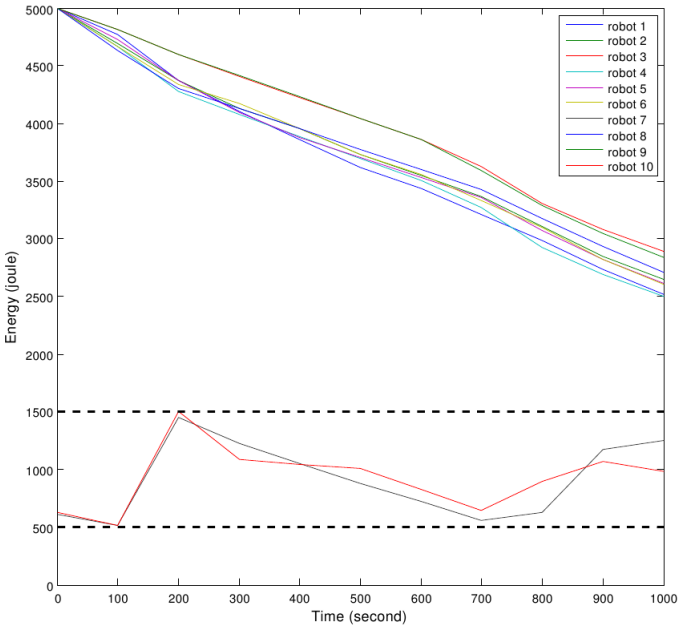


Figure 6.19: The energy autonomy of ten robots with two failing robots in the system using the shared nearest charger algorithm.

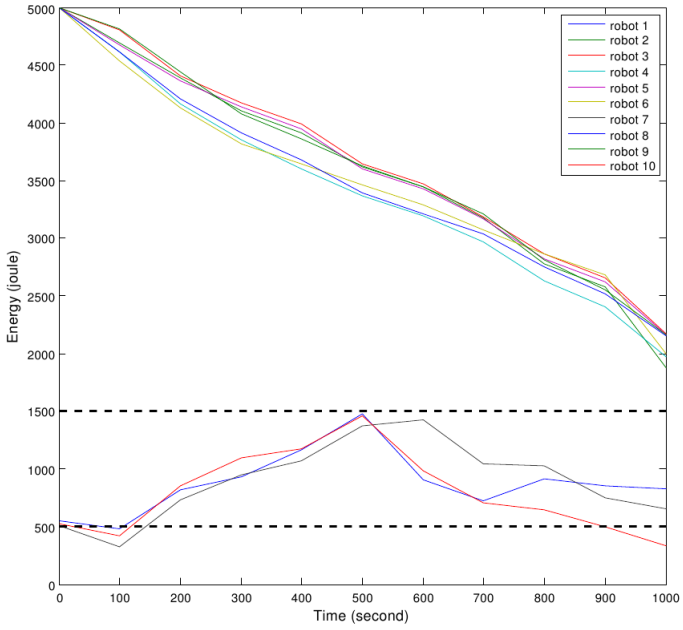


Figure 6.20: The energy autonomy of ten robots with three failing robots using the shared nearest charger algorithm.

6.4.3 Experimental Findings

So far, we have studied the effect of single nearest charger and shared nearest charger algorithms in an attempt to resolve the potential anchoring issues for power failure in the

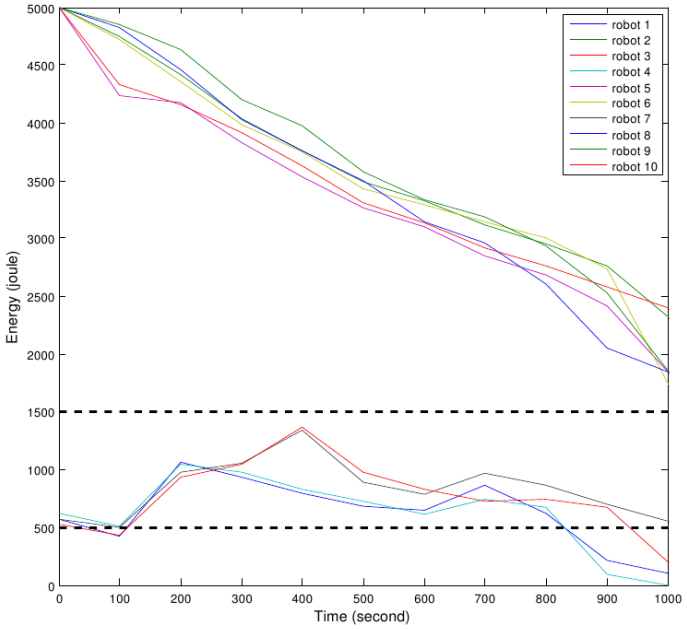


Figure 6.21: The energy autonomy of ten robots with four failing robots using the shared nearest charger algorithm.

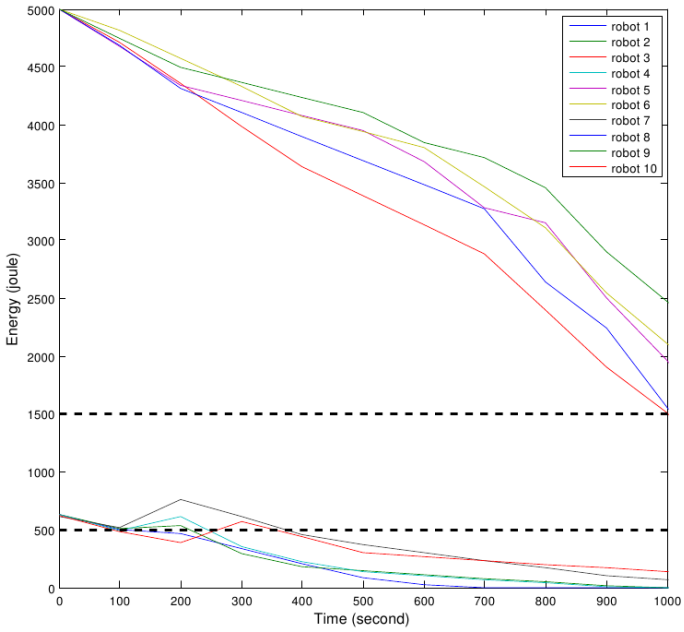


Figure 6.22: The energy autonomy of ten robots with five failing robots using shared nearest charger algorithm.

context of swarm beacon taxis. Based on these results, we observe issues with swarm taxis in line with Bjerknes & Winfield (2010) and with simple repair strategies are unable to mitigate these issues. With a single nearest charger algorithm, only one robot needs to share its energy with a faulty robot. However, since only one robot is sharing the energy, it has to give a large proportion of the required energy to the faulty robot, resulting in a major reduction in its own energy. This issue is crucial, and potentially not scalable, when the number of faulty robots increases. In an attempt to resolve this issue, we proposed an enhancement by increasing the number of simultaneous chargers. Here, we proposed a shared nearest charger algorithm, with three charging robots for each failing robot in the environment. Compared with the single nearest charger algorithm, the robots no longer suffer from major energy reduction. However, when too many robots try to reach the faulty robots, docking and navigation is clearly an issue, as robots interfere with each other to recharge the failing robots. Thus, a design of the docking and signalling algorithms than can improve the navigation is clearly needed. Due to these issues, we propose an immune-inspired solution for the anchoring issues inspired by the process of granuloma formation in immune systems as was described in chapter 5.

6.5 Experiment V: Granuloma Formation Algorithm

In this section, we investigate the effect of granuloma formation algorithm for swarm beacon taxis. We first describe the algorithm in section 6.5.1 and discuss the results obtained from the experiments comparing the performances of granuloma formation algorithm with single nearest charger and shared nearest charger algorithms in section 6.5.2. The hypotheses for the experiments are:

$H6_0$: *The use of a granuloma formation algorithm (M4) does not improve the ability of the robots in the system to achieve a centroid distance less than 0.5 cm away from the beacon when compared to single nearest charger algorithm (M2) when more than two faulty robots are introduced to the systems.*

$H7_0$: *The use of a granuloma formation algorithm (M4) does not improve the ability of the robots in the system to achieve a centroid distance less than 0.5 cm away from the beacon when compared to shared nearest charger algorithm (M3) when more than two faulty robots are introduced to the systems.*

6.5.1 Description

Having identified and presented the idea of granuloma formation algorithm in chapter 5, we now instantiate the algorithm for anchoring issues in swarm beacon taxis in ω-

algorithm and explain the results obtained from instantiating the algorithm in swarm beacon taxis.

6.5.2 Results and Analysis

The experiment tests hypothesis $H6_0$ and $H7_0$ to determine the performance of the granuloma formation algorithm. We first test the granuloma formation algorithm on swarm beacon taxis with one and two failing robots in the system. We then continue to test the algorithm with three and more failing robots in the system. M4 is assessed with the same scenario as the single nearest charger algorithm (M2) and the shared nearest charger algorithm (M3). We then compare the algorithms and examine the statistical and scientific differences with ranksum and A measure tests.

During the experiments, we first introduce one and two failing robots in the system. In both experiments, the swarm is able to reach less than 0.5 *cm* away from the beacon approximately at t = 800 seconds and t = 850 seconds. These are illustrated in figure 6.23 and figure 6.24. When introducing three, four and five failing robots in the systems, the swarm is still able to reach the beacon and the failing robots are able to be charged by the other functional robots in the system, allowing the failing robots to continue moving. The results are shown in figure 6.25, figure 6.26 and figure 6.29. From these figures, when there are three, four and five failing robots in the system, the swarm is able to reach less than 0.5 *cm* away from the beacon approximately at t = 850 seconds, t = 950 seconds and t = 1000 seconds, which is towards the end of the simulation time. However, when we compared M4 with M3 and M2 reported in section 6.3.2 and section 6.4.2, we can say that even though the swarm can reach the beacon towards the end of the simulation time it is still a promising result, as all the failing robots are able to be charged by the other functional robots in the system and they can continue moving towards the beacon.

We continue the experiments by measuring the significant difference between M4 with single nearest charger algorithm (M2) and shared charger algorithm (M3). All results are shown in table 6.7 and 6.8. We first highlight the significant difference between M4 and M2. With one, two and three failing robots in the system, there is no significant difference between M4 and M2. This explains that by having a small number of failing robots in the environment, most of the mechanisms are able to make the systems recover and operate to perform the needed task. Therefore, based on the results obtain from the experiments, with one, two and three failing robots, we fail to reject $H6_0$ if compared with M2 at the default of α= 0.05 significance level, indicated by the p value=0.7903 with one failing robot, a p value=0.4722 with two failing robots and a p value=0.0350 with three failing robots, which is much greater than the α. However with four and five failing robots in the

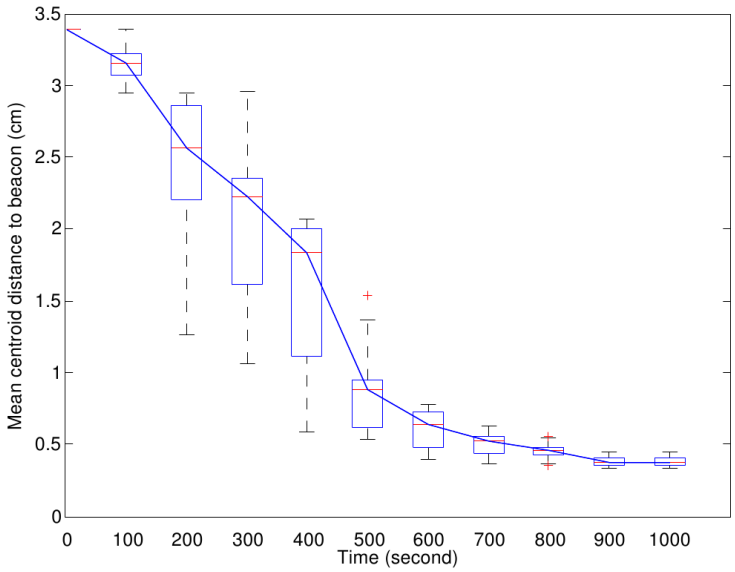


Figure 6.23: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using granuloma formation algorithm with one robot fails at t=100 for $H6_0$ and $H7_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The swarm reaches the beacon approximately at t=650 with one failing robot in the system.

system, we reject $H6_0$ if compared with M2 at the default of α= 0.05 significance level, indicated by the p value=1.0650e-04 with four failing robot and a p value=1.7861e-04 with five failing robots in the system.

When comparing M4 with M3, the experiments again fail to reject $H7_0$ at the default of α= 0.05 significance level with one and two failing robots in the system. This is indicated by p value=0.2563 and p value=0.0307 that are greater then the α. The experiments reject the hypothesis $H7_0$ at the default of α= 0.05 significance level with two, three, four and five failing robots in the system, which is indicated by the p value = 0.0307, p value = 0.0081, p value = 1.0650e-04 and p value = 1.7856e-04 that has fallen below the α. The 95% confidence interval on the mean centroid distance of robots to the beacon is less than 0.5 *cm* is obtained from this experiment. In summary, from the P and A values shown in table 6.7 and table 6.8, M4 does not differ statistically from M2 and M3 when one and two failing robots are introduced into the system. However, when there are three, four and five failing robots in the system, it is statistically proven that M4 performs significantly better if compared with M2 and M3.

In the experiments, we also evaluate the robot's energy autonomy in order to study the distribution of robots' energies in the swarm. We compare the energy distributions with the results that we obtained from previous experiment done for the single nearest

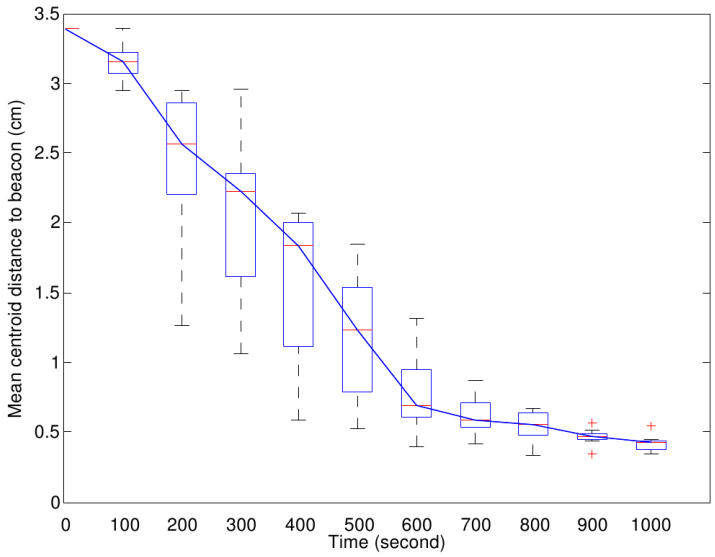


Figure 6.24: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using granuloma formation algorithm with two robot fails simultaneously at t=100 for $H6_0$ and $H7_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The swarm reaches the beacon approximately at t=800 with two failing robots in the system.

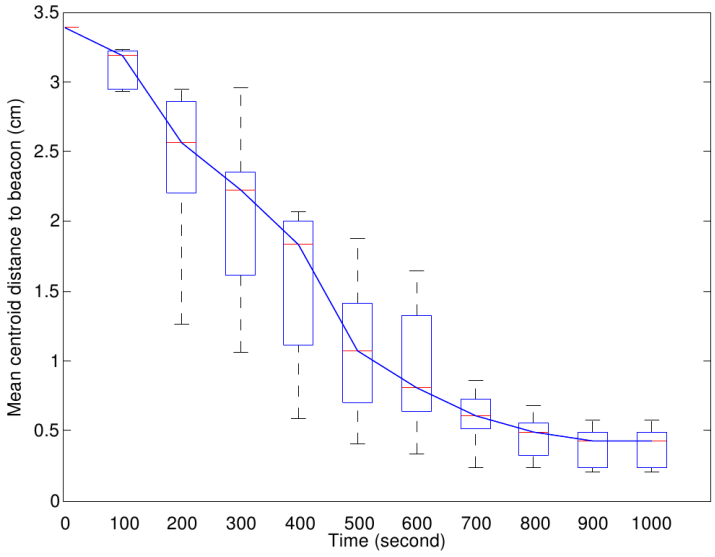


Figure 6.25: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using granuloma formation algorithm with three robot fails simultaneously at t=100 for $H6_0$ and $H7_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The swarm reaches the beacon approximately at t=850 with three failing robots in the system.

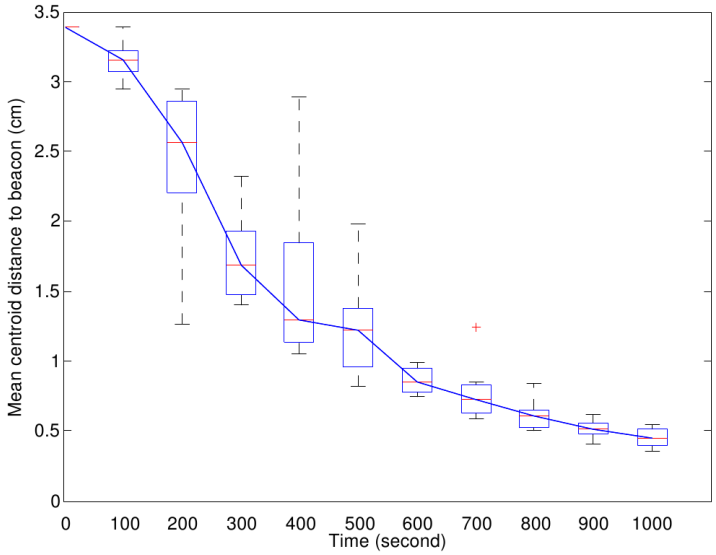


Figure 6.26: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using granuloma formation algorithm with four robot fails simultaneously at t=100 for $H6_0$ and $H7_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The swarm reaches the beacon approximately at t=950 with four failing robots in the system.

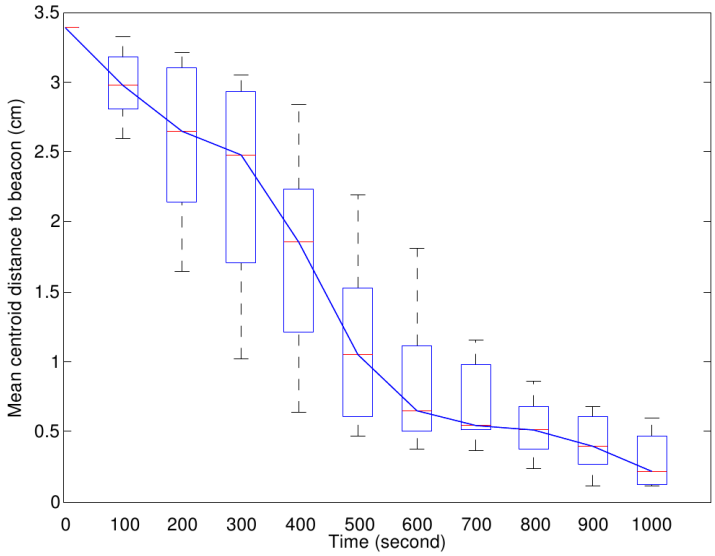


Figure 6.27: Boxplots of the distance between swarm centroid and beacon as a function of time for 10 experiments using granuloma formation algorithm with five robot fails simultaneously at t=100 for $H6_0$ and $H7_0$. The centre line of the box is the median while the upper edge of the box is the 3^{rd} quartile and the lower edge of the box is the 1^{st} quartile. The swarm reaches the beacon approximately at t=1000 with five failing robots in the system.

Table 6.7: P and A value for mean centroid distance on robots between M4 and M3 in 1000 simulation seconds.

M4/M2	**ranksum P**	**A measure**
One fail	0.7903	-
Two fails	0.4722	-
Three fails	*0.0350*	0.8900
Four fails	*1.0650e-04*	1
Five fails	*1.7861e-04*	1

Table 6.8: P and A value for mean centroid distance on robots between M4 and M2 in 1000 simulation seconds.

M4/M3	**ranksum P**	**A measure**
One fail	0.2563	-
Two fails	*0.0307*	0.8900
Three fails	*0.0081*	0.8550
Four fails	*1.0650e-04*	1
Five fails	*1.7856e-04*	1

charger algorithm (M2) in section 6.3.2 and the shared nearest charger algorithm (M3) in section 6.4.2. Here, we describe the results obtain from the experiments highlighting the distribution of energy of the three algorithms with three, four and failing robots in the system. The results showing the energy of ten robots during simulation run with three, four and five failing robots in swarm beacon taxis with granuloma formation are illustrated in figure 6.28, figure 6.29 and figure 6.30. From these figures, we can see that with three, four and five failing robots in the system, the average energy for each robot in the system is still above the minimum energy threshold, which is 500 J. The following results are observed during the experiments:

1. with one and two failing robots: all three algorithms (M4, M3 and M2) instantiated to ω-algorithm are able to charge the failing robots and the swarm is able to move towards the beacon. Since the number of failing robot in the system is low, this issue is not crucial and we observe that all robots in the system with M4 and M3 have energy above the energy threshold until the end of the simulation runs. However with M2 approximately three to four robots in the system have their energies fall below the energy threshold value at approximately t = 900 seconds.

2. with three failing robots: with M4, as shown in figure 6.28 we observe that all robots in the system have energy above the minimum threshold value (500 J) during the simulation run. With M2, approximately eight robots have reached the minimum energy threshold whilst with M3, around three to four robots have reached the minimum energy threshold value at approximately t = 900-950 seconds.

3. with four failing robots: as illustrated in figure 6.29, we observe that with M4, all robots in the system still have energy above the minimum threshold value. With M2, all robots in the system have suffered a low energy level approximately at t = 850-900 seconds whilst with M3, around three to four robots have reached the minimum energy threshold value approximately at t = 900-950 seconds.

4. with five failing robots: as shown in figure 6.30, we can see that with M4 all robots still have sufficient energy to move towards the beacon with all the robots having their energy above the threshold value. However, with M2 we observe that all robots have start to suffer with low energy at approximately t = 800 seconds. Whilst with M3 around five robots in the system suffer with energy below the threshold value.

Based on the results reported above, we can summarise that with granuloma formation algorithm instantiated to ω-algorithm, we can provide a feasible solution for the 'anchoring' issue in swarm beacon taxis. This is because rather than having a fixed number of functional robots that can charge the faulty robot(s), in granuloma formation algorithm the number is not fixed and it is rather dynamic, as has been described in chapter 5. The functional robot(s) that can charge the faulty robot(s) will depend on the position as well as the energy transfer rule that has been described in section 5.2. From the experiments that we have conducted, this algorithm may be one of the potential solutions to solve the 'anchoring' issue in swarm beacon taxis.

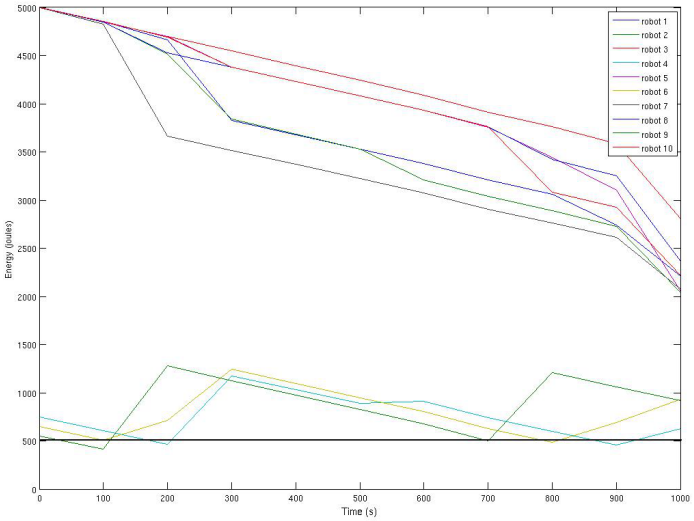


Figure 6.28: The energy autonomy of 10 robots with 3 robots fail using granuloma formation algorithm.

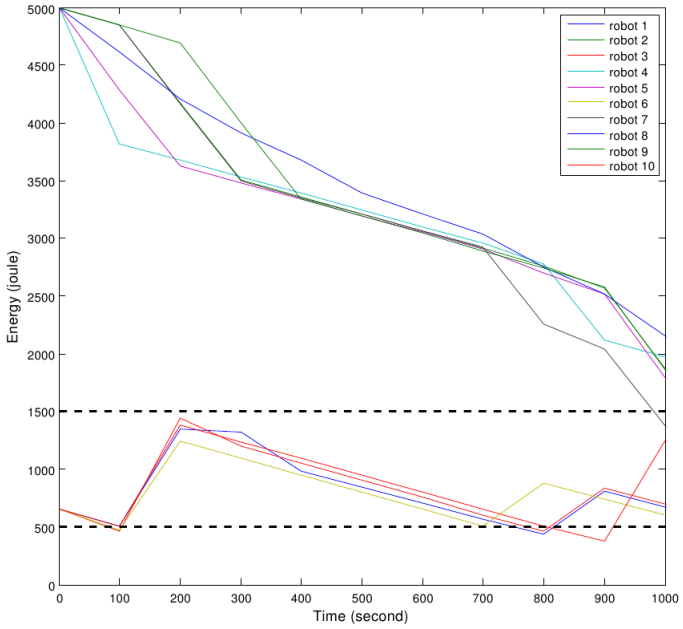


Figure 6.29: The energy autonomy of 10 robots with 4 robots fail using granuloma formation algorithm.

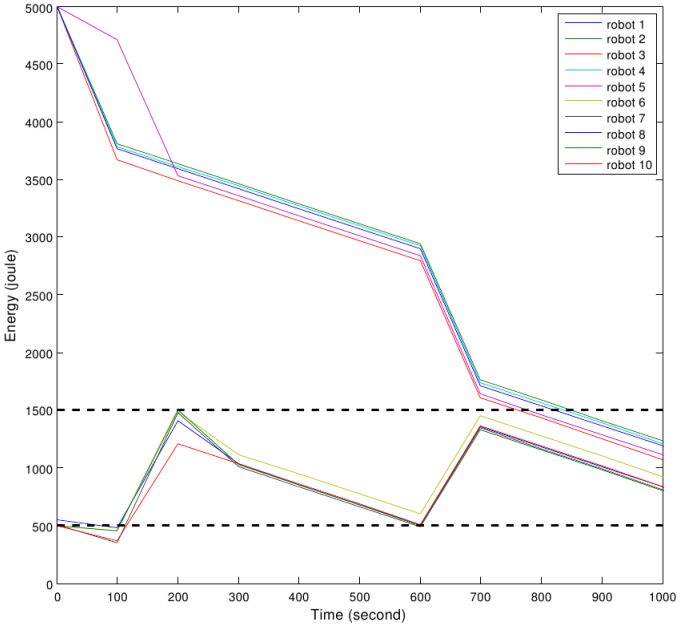


Figure 6.30: The energy autonomy of 10 robots with 5 robots fail using granuloma formation algorithm.

6.6 Conclusions

In this chapter, we began by discussing the experimental protocol in player/stage simulation in section 6.1. We then run series of experiment to assess the performance of the swarm in dealing with the partial failure of robot(s). We began our experiment with the investigation of ω-algorithm (Bjerknes, 2009) in section 6.2 for swarm beacon taxis and the possible 'anchoring' issue that may occur due to the failure of robots in the systems. The result of ω-algorithm will work as a baseline throughout the progression of the experiments. We then proposed a number of solutions that might potentially mitigate the effect of observed 'anchoring' issue called the single nearest charger algorithm discussed in section 6.3 and the shared nearest charger algorithm described in section 6.4. From the proposed solution, we studied the effects of applying both algorithms to swarm beacon taxis. We finally described the experimental findings and investigated the performance of granuloma formation algorithm inspired by the granuloma formation in section 6.5, which we proposed in chapter 5 based on our model and simulation in chapter 4. Based on the results and analysis obtained in section 6.5.2, we observed that the algorithm that we developed based on our model and simulation of granuloma formation can be one of the solutions for dealing with the 'anchoring' issue described in section 2.5. However, in dealing with a low number of failing robots in the system, we can say that single and shared nearest charger algorithms can also solve the 'anchoring' issue and the swarm can still moves towards the beacon. As the number of failing robot increases, we observed that single and shared nearest charger algorithms cannot solve the 'anchoring' issue in swarm beacon taxis. In the single nearest charger algorithm, when there exist three, four and five failing robots in the system, the robots start to experience a large energy drain leading to half of the swarm having their energy below the energy threshold value (500 Joules). Thus, when there are three, four and five failing robots in the system the single nearest charger algorithm is not a feasible solution for the 'anchoring' issue in swarm beacon taxis. With shared nearest charger algorithm, since the energy is distributed evenly with three nearest functional robots, it can still be a solution if there are three and four failing robots in the system. However, with five failing robots, the algorithm needs to take into consideration some issues such as the docking and the movements of the robots in the system. Having too much time to move towards the faulty robots will lead to a large energy drain for the functional robots and again the 'anchoring' issue cannot be solve.

Having single and shared nearest charger algorithms as an option in solving the 'anchoring' issue in swarm beacon taxis, based on the results obtained in section 6.5.2, we summarised that the granuloma formation algorithm can be a possible solution in dealing with failing robots in the system. Based on the results from the experiments, we observed that the algorithm is able to solve up to five failing robots in the system. The energy

of each of the robot is still above the minimum threshold value. This indicates that the robots in the swarm can still move towards the beacon. Even though the granuloma formation algorithm can be a possible solution, we believe we can improve the algorithm by looking into other energy transfer rules and extending the algorithm. At the moment, the algorithm just takes into considerations the functional robots that is located at a predetermined radius to make sure that the other robots can continue moving towards the beacon. The number of the functional robot that charge the failing robot(s) is also not fixed. Only one until three functional robots can charge the failing robot(s) at the moment.

During the experiments with granuloma formation algorithm, we observe that as the number of failing robots increase, the functional robot can still charge the failing robots if the distance is more that 2 *cm* away from the beacon as most of the robots still have enough energy in charging the failing robots. However, as the distance between the swarm and the beacon is less than 0.5 *cm* , there is a possibility that the functional robots' energy is getting lower. Therefore, more functional robots are needed in charging the faulty robots and if most of the functional robots' energy is getting lower, they will not have the ability in charging the faulty robots, leaving them as an obstacles in the systems. In applying the algorithm to the real e-puck robots, we do believe that the results similar to the simulation can be obtained depending on the design of the hardware according to the design of the charging mechanism in our simulation. However, since the work in this thesis is intended to propose an immune-inspired algorithm based on the model and simulation of immune systems applying to the issue in the simulation of swarm beacon taxis , we believe that this can be part of our work in the future.

In term of robotics literature, we considered that the algorithm is novel as there is less work that emphasised on the self-healing mechanism in swarm robotic systems. Even though in swarm robotic systems, there is the need for the robots in the system to have characteristics such as robustness, flexibility and scalability, but there is less work demonstrating this characteristics. Based on the literature survey conducted, we found out that the idea of trophallaxis, which we described as single nearest charger algorithm in section 6.3, is the only mechanism that allow robot to self-charge the other robots in the system. Other than this, most of the solution will involve a static or a removable charger that needs human intervention. Therefore, we do believe that our early work can be a possible novel contribution to the field of swarm robotic systems, which can be useful in the future. As a conclusion, we do believe that the model and simulation for granuloma formation that we undertook in chapter 4 has assisted us throughout the development of our immune-inspired algorithm. Having described experimental results in this chapter, we will discuss our conclusions and future work in chapter 7.

CHAPTER 7

Reflections on the Development of Immune-inspired Solution for Swarm Robotic Systems

This chapter reflects on the experience of developing an Artificial Immune Systems (AIS) for self-healing swarm robotic systems. In particular, this thesis focuses on the development of swarm robotic systems that are able to contain certain types of error and initiate repair strategies to allow energy sharing between robots, when there exist failures of robots' energy in the systems. This chapter summarises work presented in chapter 2, 3, 4, 5 and 6 within the swarm robotics context in section 7.1. We provide a summary of our work at each of the stages of the conceptual framework and how this principle method has affected our work in developing AIS. We then analyse how we have explored the problem domain in swarm robotic systems and how we have identified the biological process for inspiration that has the similar properties of the problems so that we can instantiate the idea to the problem that we have. We also briefly explained on how we probe, model and simulate the process of granuloma formation in biological systems before we come out with the AIS frameworks and algorithms. Based on the research presented, we provide reflections on following the principle method taking into account the engineering-informed modelling approach (Hart & Davoudani, 2011) to develop an immune-inspired algorithm for a self-healing swarm robotics system that is able to initiate repair strategies to allow energy sharing between robots in section 7.2. Section 7.3 provides conclusions of this thesis by summarising our work, identifying future work as well as giving feedback on our research questions described in chapter 1.

7.1 Summary of Work

In section 3.3, we discussed the CFA Stepney et al. (2005) which describes a principled method for the development of novel immune-inspired algorithms. In this section we will analyse the work presented in chapters 2, 3, 4, 5, and 6 with respect to CFA approach taking into consideration the engineering-informed modelling approach proposed by Hart & Davoudani (2011), which we described in section 3.4.3. We begin with section 7.1.1 describing our work in exploring the problem domain in swarm robotic systems specifically the 'anchoring' issue that exists in swarm beacon taxis due to partially-failing robot(s) before explaining our work in applying CFA for the development of immune-inspired algorithm in accordance with the 'anchoring' issue in swarm beacon taxis section 7.1.2.

7.1.1 Understanding the Problem Domain in Swarm Robotic Systems

As mentioned in Hart & Davoudani (2011), the constraints of the engineered problem must be informed during the model development and validation phase. The enables the development of AIS which is consistent with the application constraint and can be validated in terms of functional requirements of the application rather than towards the biological needs. In understanding the application domain and constraint, we explore the field of swarm robotic systems in chapter 2. In this chapter we emphasised the challenges and issues in swarm robotics, specifically in maintaining the robustness of the swarm robotic systems. The main application discussed in this chapter is swarm beacon taxis in section 2.3, which is an aggregation task in swarm robotic systems where the swarm collectively moves towards a beacon. In swarm beacon taxis, Nembrini et al. (2002) and Bjerknes (2009) developed a class of aggregation algorithm, which makes use of local wireless connectivity information alone to achieve swarm aggregation. They are the α algorithm (Nembrini et al., 2002), β algorithm (Nembrini et al., 2002) and ω algorithm (Bjerknes, 2009). This application is used as our experimental case study in this thesis.

In swarm robotic systems, various types of failure modes and the effect of individual robot failures on the swarm have been analysed. As stated in Bjerknes & Winfield (2010) the failure modes and effects for swarm beacon taxis are: 1) complete failures of individual robots (completely failed robots due, for instance, to a power failure) might have the effect of slowing down the swarm taxis towards the beacon, 2) failure of a robot's IR sensors and 3) failure of a robot's motors only. These failure modes, described in section 2.5, may cause the failed robot to 'anchor' the swarm, impeding its taxis toward the beacon. For our case study, we focused our attention on the effect of a partially-failed robot to 'anchor' the swarm leading to impeding its taxis toward the beacon inherent in

the ω-algorithm (Bjerknes & Winfield, 2010).

In exploring the 'anchoring' issue, we implemented the ω-algorithm (Bjerknes & Winfield, 2010) in the swarm robotics simulation tool, the Player/Stage (Vaughan & Gerkey, 2007) in chapter 6. These experiments are implemented mainly to reproduce the effect of faulty robots in swarm beacon taxis or the 'anchoring' issue reported by Bjerknes (2009). During these experiments we measure the progression of the centroid of the swarm towards the beacon when there are one to five failing robots in the simulation. These experiments served as a baseline from which we calibrated our proposed immune-inspired solution. The following set of hypotheses are tested during the experiments to show the effect of the failing robot(s) in the systems:

1. *The use of ω-algorithm (M1) for swarm beacon taxis allows the swarm to achieve a centroid distance less than 0.5 cm away from the beacon when there are no failures introduced.*

2. *The use of ω-algorithm (M1) for swarm beacon taxis allows all robots in the swarm to achieve the distance less than 0.5 cm away from the beacon with a failing robot in the environment.*

The results from the experiments conducted in chapter 6 show that, even with two partially-failed robots, the swarm will always reach the beacon, and the delay of time in reaching the beacon is relatively small. However, as three faulty robots were available in the simulation, the swarm started to experience the effect of 'anchoring' described in section 2.5. The faulty robots became an anchor and the swarm will move around these faulty robots without reaching the beacon. These experiments confirm the potential issue of 'anchoring' as has been reported by Bjerknes (2009).

These experiments allowed us to understand the application and the problems that exist in swarm robotic systems specifically in swarm beacon taxis. Based on our understanding, we have worked towards solving this issue by exploring the biological systems and proposed an immune-inspired solution specifically tailored for the effect of partially-failing robots, resulting from a large energy drain in swarm robotic systems, that is able to contain certain type of errors and initiate repair strategies to allow energy sharing between robots, when there are robots with energy failure in the system.

Having outlined how we explored the problem domain in swarm robotic systems, we will now present the summary of our work on how we apply the principle of CFA in engineering swarm robotic systems in section 7.1.2. This is in accordance with our goal of developing an the immune-inspired solution to the application in swarm robotic systems, specifically for the 'anchoring' issue that exists in swarm beacon taxis (Bjerknes, 2009), as described above.

7.1.2 Applying Conceptual Framework in Developing Immune-Inspired Solution for Swarm Robotic Systems

The CFA proposed by Stepney et al. (2005) highlighted the need for bio-inspired algorithms such as AIS need to be developed through a principled approach. Based on our description of CFA in section 3.3 of this thesis, we depict the stages of CFA again in figure 7.1 for our discussion in this section.

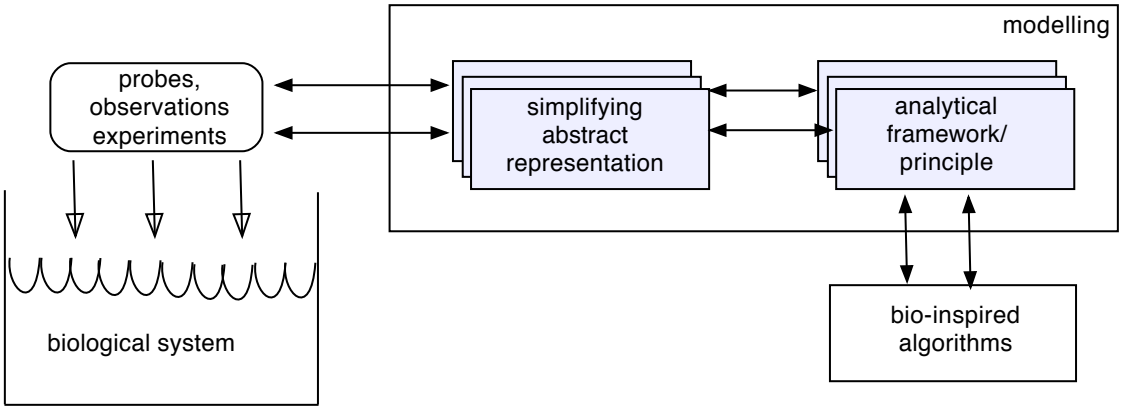


Figure 7.1: Conceptual framework of AIS (Stepney et al., 2005).

In summary, from figure 7.1, CFA is based on the following processes:

1. probes, observation and experiments.
2. abstract representation, based on a model of biological systems.
3. analytical computational frameworks.
4. the bio-inspired algorithms.

In this thesis, work presented in section 3.5, chapters 4, 5, and 6 followed different parts of the CFA to achieve our goal of developing a novel AIS for self-healing swarm robotic systems, specifically tailored for the effect of partially-failing robots resulting from a large energy drain in swarm robotic systems, that is able to contain certain types of error and initiate repair strategies to allow energy sharing between robots. For convenience, we summarise each stage of CFA, which we describe as follows:

- immunology and probes: the immunology discussed in section 3.5 of chapter 3 in this thesis is from immune systems' research literatures, which was limited to research literature presented in books, journals and articles. We study and investigate the relevant aspects of immunology in the research literature that is relevant to become a source of inspiration for self-healing mechanism in swarm robotic systems.

- simplifying computational model: we developed a computational model and simulations of granuloma formation, which is explained in chapter 4 of this thesis from the research literature. The model and simulation is simplified in accordance to the constraint that we obtained while exploring the problem domain of swarm robotic systems, specifically the 'anchoring' issue in swarm beacon taxis that we described in section 2.5.

- algorithm framework/principle: we proposed sets of design principles and algorithm framework in chapter 5 of this thesis that is instantiated from the development of the computational models and simulation.

- artificial immune systems: a novel AIS for self-healing swarm robotic systems, which is described in chapter 5 and implemented in chapter 6 of this thesis.

Based on the stages of CFA summarised above, we describe each work presented in this thesis in adapting the CFA to develop an immune-inspired solution for self-healing swarm robotic systems that is capable in sharing of energy between robots in the system. We begin by describing the immunology and probes, the computational models, algorithm framework/principle and finally the algorithm.

In section 3.5 in chapter 3, we presented the initial probes within the immunology research literature in order to identify immune mechanisms and properties that might provide inspiration for our immune-inspired solution. Here, we focused our discussion on the process of granuloma formation (Adams, 1976; Facco et al., 2007; Sneller, 2002). As described in section 3.5, the importance of the granuloma is that it provides a 'wall' of macrophages and T-cells surrounding infected cells, such that when the infected cells die, the surrounding macrophages will try to prevent the spread of infection. In the absence of these cells surrounding the infected cells, bacteria can be released, allowing bacterial replication to the other cells in the systems. Based on our probes into the immunology research literature on granuloma formation, the pathogenesis or the development of granulomatous inflammation is complex, and involves a variety of mechanisms acting in concert to bring about an inflammatory lesion that is able to contain and destroy bacterial infections. Most of the literature is specifically tailored for specific diseases and needs further elaboration from an immunologists. However, since our work relies more on understanding the process of granuloma formation not specific to any diseases, we try to understand the general process of granuloma formation based on our probes into the immunological literature which we have described in section 3.5. We also proposed that there is a natural analogy between the potential repair of a swarm of robots, as in the situation of swarm beacon taxis, and the formation of a granuloma and removal of bacterial infections to the cells. This is summarised in table 4.6, which we explained in section 4.7.

The analogy outlined in table 7.1, described the properties of both swarm robotics and granuloma formation. Based on this analogy, we then proposed how granuloma formation can act as a source of inspiration for the development of immune-inspired algorithm in section 4.7.

Table 7.1: Properties of swarm robotics and granuloma formation

Properties of swarm robotics	Properties of granuloma formation
Large number of robots	Large number of cells
Few homogeneous groups of robots	Few homogeneous cells
Relatively incapable or inefficient robots	Each cell needs to perform the desired task
Robots with local sensing and communication capabilities	Chemokines and cytokines

To recap, in section 4.7, we mentioned that granuloma formation is important in immune systems, as it acts as a healing mechanism, trying to prevent the bacterial infections from infecting other cells and to contain the infection by attracting other cells such as macrophages and T-cells to move to the site of infection. We proposed that this can be applied to solve the 'anchoring' issue described in section 2.5, that allow the robots in the system to contain certain type of errors and initiate repair strategies to allow energy sharing between robots in the system. The idea is that to allow swarm with many robots working together to continue operating even if some of the robots fail, which is previously described in section 4.7.

Further to the probes into the immunology research literatures in section 3.5 in chapter 3, we developed a computational model consisting of a Unified Modelling Language (UML) model and a simplified agent-based simulation to assist understanding of the interactions of cells during the development of granuloma formation in chapter 4. Following the CoSMoS process Andrews et al. (2010), we developed a model and simulation of the general formation and progression of granuloma formation, rather than in the case of a specific disease. This is due to the fact that we did not wish to model the formation to provide insight from a biological perspective, but understand the dynamics of a general model to allow us to distill a series of design principles that we can use to create a novel AIS algorithm. This is in line with the suggestion made by Hart & Davoudani (2011) that highlighted the development of the model and simulation tailored to the engineering domain and constraint for the development of AIS solution to engineering. This again required us to probe more of the biology to come out with the development of the model and simulation. During this stage, we first prepared the domain model, noting down the behaviours of the system in which we are interested as well as dividing the behaviour of granuloma formation into main stages and describe each stage based on our probes

and immunological literatures in section 4.2. This is in line with the suggestion made by Read et al. (2009a) that argued that the simulation needs to be to be properly delineated because there exists a huge variety of elements interacting in the biological system, and it is impossible to simulate them all. We used UML diagrams as a tool to develop our model before preparing the agent-based simulation. Having prepared the domain model in section 4.2, we developed the platform model in section 4.3, focuses on how the process are going to be implemented and executed in the simulation platform. Finally, we developed the simulation of granuloma formation with Netlogo [1], explained in section 4.4. The agent-based simulation that we produced kept the interactions of cells during the process of granuloma formation, allowing us to produce a simulation that is closer to the analogy of granuloma formation.

Based on the modelling and simulation work presented in chapter 4, in chapter 5 we detailed our work on the final two phases of the conceptual framework approach (CFA) of Stepney et al. (2005), which is to prepare framework/principle and the instantiation of the framework/principle for the development of a novel AIS. We outlined four key design principles from the models and simulation and proposed a granuloma formation algorithm for self-healing swarm robotic systems in section 5.1.

Upon the development of granuloma formation algorithm in chapter 5 we finally implemented the algorithm for swarm robotic systems using the sensor-based simulation tool set, Player/Stage (Gerkey et al., 2003) in chapter 6. This is mainly to solve the 'anchoring' issue swarm beacon taxis (Bjerknes, 2009; Bjerknes & Winfield, 2010) described in section 4.7, in particular, for the development of swarm robotic systems that are able to contain certain type of errors and initiate repair strategies to allow energy sharing between robots, when there exist robots' energy failure in the systems. The results are compared with the single nearest charge mechanism, which was part of the trophallaxis work by Melhuish & Kubo (2007) and the shared nearest charger mechanism. From the results, we have been able to show that granuloma formation algorithm is able to solve the 'anchoring' issue in swarm beacon taxis and the swarm is able to achieve the beacon even with more than three failing robots in the environment.

Based on the description of our work presented in this section, we now proceed to discussing on the reflections on the development of immune-inspired solution for swarm robotic systems, specifically to initiate repair strategies to allow energy sharing between robots due to energy failure in section 7.2.

[1] http://ccl.northwestern.edu/netlogo/

7.2 Reflections on the Development of Immune-inspired Solution for Swarm Robotic Systems

In this section we provide our feedback on following the principle approaches to developing an AIS algorithm for self-healing swarm robotics systems which is able to share energy between the faulty and unfaulty robots in the system. We begin in section by giving our feedback in developing the AIS algorithm according to CFA stages in 7.2.1 and our contributions to swarm robotic systems in section 7.2.2.

7.2.1 Reflections on the Conceptual Framework

During the early stage of this work, we have identified a specific engineering problem that we would like to deal with. In particular, we would like to have a novel immune-inspired solution for self-healing swarm robotic systems, that is capable in solving the 'anchoring' issue due to the failing robot(s) that experience large energy drain in the system in swarm beacon taxis (Bjerknes, 2009; Bjerknes & Winfield, 2010). This is in line with the suggestion made by Timmis et al. (2008b); Hart & Davoudani (2011), who suggested that to have an application's problem and constraint in mind before moving to the development of model and simulation in accordance with the CFA approach.

Having the problem domain clear in the early stage of our work helped us throughout the development of the immune-inspired algorithm following the CFA approach. Secondly, with an application-based approach to the CFA the decisions that we took throughout the entire process of following the CFA were driven by the application or algorithm-type (in our case the swarm beacon taxis). The advantage of such an approach is that we were able to tailor our probes to the immunology specifically to the problem that we were dealing with. We explored the immunology that can be a source of inspiration in solving the issues that we wished to solve, rather then studying the general immunology. In our work, once we identified the 'anchoring' issue due to the partially-failed robot that experienced a large energy drain in the system, leading to its motor-failure, we then look into potential solution that can be used from the immunology. From our probes to the immunology, we identified the process of granuloma formation, a process of the cells of immune systems trying to stop the bacterial infections to infect cells in the immune systems. It is also a response to an invader that requires recruitment of immune cells to isolate and, if possible, deal with the problem that occur. We also did not do an exhaustive analysis of immune mechanisms, but simply sought a promising approach about which there was local knowledge in the AIS research group. We also did believe that despite the process of granuloma formation, there probably other immune processes that would be

good inspiration for our problems. Based on the probes, we identified that there is a analogy between the process of granuloma formation and swarm robotics systems, which is summarised in section 7.1.2. Therefore, rather than looking into a vast array of biological inspiration, which is difficult to try, and cover every principles and process in an adequate amount of time, we specifically look and cover the principles that can be the source of inspiration to the specific issue in mind, as summarised in section 7.1.1. Thirdly, we realised that there is such a vast array of cells and interactions in granuloma formation. We found out that it was going to be very difficult to try and cover every process in granuloma formation. However, since we already knew the type of problem that we were interested on, and knowing the data that we would like to use in the final application of our work, we can determine what, and what not, to include in our model and simulation.

In general, we can provide the following advice to others when attempting to develop immune-inspired algorithm to solve engineering problems.

1. Identify the application or problem domain that you are interested to solve.
2. Investigate the problem in 1 by producing sets of experiments that show the effect of the problem.
3. Identify the possible immunological principles than can act as a source of inspiration to solve the problem by following the CFA approach.

Once the problem domain is clear, we move on to the development of AIS algorithm based on CFA approach. Even though our discussion is based on the experience gained from following the CFA according to our work, we believed that one can use it to gain some insight into the development of immune-inspired algorithm by following a principle approach like CFA.

Immunology and Probes

Probing the immunology is the first phase in CFA described in section 3.5 from which we draw inspiration. During this phase, we have to identify which types of biology should be used as our inspiration for the development of the immune-inspired algorithms with a large amount of biological resources that are available for probing. As commented in Andrews (2008), the phases of CFA as proposed by Stepney et al. (2005), do not explain in any detail how one can identify which biological systems would be suitable to provide inspiration, and how one can identify the aspects of the systems that might be a good inspiration for an algorithm. As argued in Andrews (2008), to identify the bio-inspired area of interest is quite easy to address; for example in this thesis since our bio-inspired

area of interest is AIS, our biological system is the immune system. However, in identifying which aspects in the immune system might be a good inspiration required more investigation.

In the case of this thesis, our choice of the process of granuloma formation as an interesting immune property to investigate was based on the analysis of our target application and problems that we would like to solve, as presented in chapter 2. This is also in line with the suggestion made by Hart & Davoudani (2011), that emphasised on the understanding of the problem domain and constraint before modelling and simulation of the immune systems. Once we had identified the desire to understand the process of granuloma formation, the next step was to understand how it works. However, while probing the immunology to understand the immunology, we realised that most of the literatures describes granuloma formation for specific diseases. Since we were interested in understanding the general properties of granuloma formation, we needed to identify the main cells and interactions that exist. This was achieved via the modelling stage in CFA that allowed us to extract the biological properties of granuloma formation as well as the development of model and simulation, which we explored in chapter 4.

In general we found out that since our main objective was to develop an immune-inspired algorithm, specifically for 'anchoring' issue in swarm robotic systems, rather than investigating the many properties and behaviours of the immune system, we only focused our attention on the possible properties that have the potential to become a source of inspiration to the problem that we were dealing with.

Simplifying Computational Model and Algorithm Framework/Principle

In the CFA, the aim of the modelling stages is to aid the understanding of the underlying biology by simplifying abstract representation, leading to the establishment of an analytical/framework for the development of bio-inspired algorithm. This stages will help in extracting the key properties of biology resulting to the development of bio-inspired algorithm. However in CFA it does not really describe the modelling stages rather only highlight the needs and the benefits that we can obtain from this stage. Therefore, in developing our model and simulation in a principled manner, we adopt the CoSMoS process (Andrews et al., 2010), which we described in section 3.3. The modelling and simulation of granuloma formation, which is the focus of this thesis has been reported in chapter 4 and summarised in section 7.1.2. We represented the model with Unified Modelling Language (UML) diagrams and agent-based simulation, as we believed that both techniques suited both the nature of our biological properties as well as the desired output of the model in simulation.

The modelling work that we did has enabled us to identify the main cells and sig-

nalling mechanisms exists in granuloma formation. We also dealinated the properties of granuloma formation that we would like to include in the simulation. This is because there are lots of cells and interactions in granuloma formation and it is not possible for us to simulate all the cells and their interactions as most of them are not yet well understood by the immunologist. The agent-based simulation that we presented in chapter 4 was a simplified version of the process of granuloma formation, as we only showed the interactions between the main cells and the signalling mechanisms. Thus, it was very much an exploratory simulation aimed at elaborating some simple ideas and questions on the process of granuloma formation. The simulation prepared is mainly extracted from the UML diagrams that we prepared from textual descriptions of the biology. Simplifying the model and simulation has allowed us to investigate the process and properties of granuloma formation that could be adapted to our application domain. Even though we developed the simulation based on our needs, the process of following the CFA may not end here. For example, in the future, we could return to the model and simulation of chapter 4 and look at ways of extending it. We might then be able to start asking questions regarding the more specific properties in granuloma formation such as adding more cells and interactions in the simulation. This would provide us with a way to probe back the biology and extend the model as well as the simulation. We also believe these models and simulations could be improved and developed further into a tool that could provide insights into biology. In achieving this, we need to calibrate the models to the biological system by using data from the real biological system, so that we can start generating understanding to be verified by biological experts.

In summary, with respect to the technical work presented in chapter 4 and chapter 5, we do believe that the modelling stages in CFA has assisted us in: 1) understanding the biological properties of granuloma formation based on the problem domain that we would like to solve, 2) developing a model and simulation and 3) preparing the design principles to be instantiated to the algorithm. Without the model and simulation it would not have been possible for us to develop the principles as they are mainly taken from our understanding of the process of granuloma formation observed in the model and simulation that we prepared. It may be a time consuming task, inevitably increasing the development time involved in constructing the model and simulation. However, once the models and simulations have been built, we can reduce the time spent in developing the design principles and the immune-inspired algorithm.

7.2.2 Reflections on Swarm Robotic Systems

Based on our experienced in developing immune-inspired solution for swarm robotic systems, we learnt that there are lots of properties in the immune systems that can be explored

to be instantiated for solving different types on engineering problem. In the case granuloma formation, most of the modelling and simulation work are done for understanding the behaviour of cells and the properties of the formation of granuloma. Once understood there is no further work done based on the results of the models and the simulation. However, in our case the models and simulation that we did do not only help us in understanding the properties of granuloma formation but it can become a source of inspiration for solving the 'anchoring' issues in swarm robotic systems. Therefore, we believed that there are other properties in the immune systems that are useful to be modelled and studied and become a source of inspiration for solving the engineering problems.

If we compared our work in this thesis for swarm robotic systems with other solutions as described in section 6.3 and section 6.4 that are able to initiate repair strategies to allow energy sharing between robots, when there exist failure of robots' energy in the systems, we believed that by having a CFA as a principle approach for developing immune-inspired solution has assisted in exploring the immune systems in a better way since the development of the model and simulation is tailored specifically with the problem that we have in mind. Therefore, the design principles that we obtained are very useful during the algorithm development. These design principles are also useful in solving other problems in swarm robotic systems as they can be added and amended in accordance to the need of the problems.

Even though we only did our experiment in solving the 'anchoring' issue in swarm beacon taxis, we do believe that our solution are useful in other problem relating to swarm robotic systems. related to large energy drain of the robots in the system. This is because energy is one of the major issues in swarm robotic systems and by allowing the robots to share energy between themselves will help the system to achieve its main objective. When we compared our work with other solutions such as the single nearest charger algorithm discussed in section 6.3 and the shared nearest charger algorithm described in section 6.4, we found out that our algorithm is able to solve when half of the swarm is having large energy drain in the system. Thus, the robots in the system can still achieved their task while doing repairing and sharing of energy between themselves.

From the experiment that we conducted in chapter 6, we identified that our immune-inspired algorithm can repair and share up to five faulty robots that exist in the system. However, when more failing robots were introduced into the system, then the there exist difficulties for the non-faulty robots in the systems to repair and share their energy to the faulty robots. This is mainly because, in our experiment we only tested with ten robots in the system and this work can be extended in the future work by introducing more robots to the systems. From our experiment also we found out that the proportion of the failing robot is critical to the system. For example:

- when there are five faulty robots in a system with ten robots, then the non-faulty robots might be able to repair and share their energy with the faulty robots
- when there are ten faulty robots in a system with twenty robots, then the non-faulty robots might be able to repair and share their energy with the faulty robots
- when there are fifteen faulty robots in a system with thirty robots, then the non-faulty robots might be able to repair and share their energy with the faulty robots

To summarise our work, we believed that by developing the immune-inspired self-healing swarm robotic systems in principled manner helped us in solving the 'anchoring' issues specifically allowing the robots to contain certain types of error and initiate repair strategies to allow energy sharing between robots, when there exist failure of robots' energy in the systems. We ended our discussion in this chapter by concluding our current and future work in section 7.3.

7.3 Conclusions and Future Work

In section 1.3.1, we identified that the goal of the thesis was:

> *to explore the principle development of a novel immune-inspired algorithm for self-healing swarm robotic systems that is capable of recovery from certain failure mode*

Based on the research goal, it is now appropriate for us to answer our research question according to the work that we did:

> *to what extent can an effective algorithm for achieving fault-tolerance with respect to beacon taxis in a robotic swarm could be developed using immunological inspiration?*

Based on the work presented in this thesis, we can say that we have able to develop an immune-inspired solution inspired by the process of granuloma formation by implementing each stage in CFA, specifically in solving the 'anchoring' issue in swarm in swarm beacon taxis due to the partially-failing robot(s) in swarm robotic systems that experienced a large energy drain while operating.

We have shown through the development of models, simulations and design principles in chapter 4 and 5, how this could be achieved. As our main application of granuloma formation algorithm was to allow the algorithm to be a 'self-healing' mechanism in swarm

robotic systems. The mechanism will be able to contain certain type of errors and initiate repair strategies to allow energy sharing between robots, when there exist robots' energy failure in the systems. For example, when there is a case when a transient fault occurs in robot that results in a large energy drain in the robot. What is required, is a 'self-healing' mechanism that will allow other robots to share energy with each other to repair or recharge the 'faulty' robot(s). From the model and simulation of granuloma formation described in chapter 4, and the sets of design principles and granuloma formation algorithm described in chapter 5, we have developed an AIS algorithm to be part of the solution in solving fault tolerance in swarm robotic systems. The idea is to surround the faulty robot with functional robots that are able to share energy between themselves, which is taken from the idea of the process of granuloma formation in immune systems.

- Is the immune-inspired algorithm that we develop effective in solving fault tolerance in swarm robotic systems?

  We believe that there is suitable type of solution that could be applied in solving fault tolerance in swarm robotic systems. We described the single nearest charger algorithm in section 6.3.1 that is part of the the work by Melhuish & Kubo (2007), which highlighted the idea trophallaxis. We then extended the algorithm by introducing the shared nearest charger algorithm as explained in section 6.4.1. However, based on the experiments that we did in comparing both algorithms with the granuloma formation algorithm in section 6, it can be concluded that the algorithm we developed can be one of the solution in dealing with solving the fault tolerance issue in swarm robotic systems specifically in solving the 'anchoring' problem in swarm beacon taxis.

Throughout the course of this thesis, we think that there is further future work that could take place to extend and expand the work that we have investigated. These fall into two categories: to apply the granuloma formation algorithm in real robot experiments; and further examination of the model and simulation that we have developed to further understand the mechanisms involved in granuloma formation for developing other novel AIS algorithms. The success of the algorithm that we have developed at the moment, is still at its early stage, as it has shown a promising result when we applied it to the 'anchoring' issue in swarm beacon taxis. This could be further extended by applying the algorithm to the real robots that is dealing with the same case study. Due to the time needed in extending the current robot to have a charging mechanism attached to its body, this work needs a longer time period as well as the involvement of expertise. However, this work can still be achieved and act as future work in swarm robotic systems. We also believe it would be important to try and apply the granuloma formation algorithm to another swarm robotics

case study such as foraging to show that the algorithm is generic enough to apply to more than one case study. Another important piece of future work would be to seek input from domain experts on the the model and simulation that we have developed in chapter 4, so that we might come closer to providing feedback to biology from our work. Based on the feedback, we might find other important or interesting mechanisms in granuloma formation that can be a source of inspiration for the development of AIS algorithm.

As a concluding remark, given the summaries and justifications of our contribution in this chapter, it is the author's opinion that this thesis has been a success; the challenge of the hypothesis has been met and future work is required in applying the work in real robotic systems to study the effectiveness of our AIS in swarm robotic systems.

APPENDIX A

Robustness Analysis

In this section, we present the robustness analysis, which is a methodology for accessing a significant number of simulation runs for our experiments in chapter 6. Initially, for each experiment we run the simulation ten times. However, we would like to make sure that ten simulation runs are sufficient for the experiments. Therefore, we run each algorithm fifty times and divide them into five classes. The first class which contains the ten simulation runs that serve as the baseline and be compared with other classes. In obtaining this goal, we employ the Vargha-Delaney A test (Vargha & Delaney, 2000), which is a non-parametric effect magnitude test, to determine when a simulation number adjustment has resulted in a scientifically significant change in simulation behaviour from the baseline, which is 10 simulation runs. We use the effect magnitude test to determine the scientific significance between simulation numbers in our experiment.

A.1 Description

The experiments are repeated for fifty simulation runs. The results are then divided into five classes. The first class which contain the 10 simulation runs serve as the baseline. The results for other classes are then compared with the baseline to see if there is any significant difference between each of the experiment. For each experiment, the mean centroid position of the robots during time = 1000 seconds are recorded. We then employ the A test score to evaluate the difference of each simulation run. Our hypothesis for this set of experiments is as follow:

$H8_0$: *The increasing number of simulation runs does not improve the performance of the swarm beacon taxis experiment in achieving beacon when compared to 10 simulation runs for single charger, shared charger and granuloma formation algorithms*

A.2 Results and Analysis

Table A.1 and figure A.1 show the A test score for single nearest charger algorithm for twenty to fifty simulation runs. From the figure, it can be seen that by increasing the number of simulation runs, there is no large effect on the results of the experiment. For one failing robot in the experiment, the A score values lie between 0.41 to 0.57. These values indicate that there is a small difference in the distibution. When there are two failing robots, the A test scores for single nearest charger algorithm are between 0.50 until 0.60, indicating that there are medium differences in the distribution. As the number of faulty robots increases, the value of A test scores increases from 0.64 until 0.87, showing that there are large difference in the distribution. This is mainly because with the sngle nearest charger algorithm, as the number of failing robots increases, the ability of the swarm to reach the beacon is less as compared to the low number of failing robots in the environment. All A test scores for the single nearest charger algorithm are listed in table A.1.

In accordance with the A test score, we look into the whole distribution of the single nearest charger algorithm. This is shown in figure A.2. With one and two failing robots in the system, the swarm with single charger algorithm is able to reach the beacon, leading to the small difference in the distribution. However, as the number of failing robots increases, from figure A.2 we can see that the swarm does not reach the beacon, leading to the large difference of ditribution as described earlier.

Table A.1: The magnitude of effect size indicated by A test score for different simulation runs for single nearest charger algorithm with one, two, three, four and five failing robots. A test scores with less than 0.56 (small) are marked with * which are seen in the case of one and two failing robots.

Simulation Number	20	30	40	50
A test score (1 fault)	0.51 *	0.41 *	0.54 *	0.57 *
A test score (2 fault)	0.60	0.53 *	0.50 *	0.56 *
A test score (3 fault)	0.64	0.74	0.68	0.64
A test score (4 fault)	0.87	0.71	0.72	0.90
A test score (5 fault)	0.51	0.87	0.63	0.68

Table A.2 and figure A.3 show the A test scores for shared nearest charger algorithm

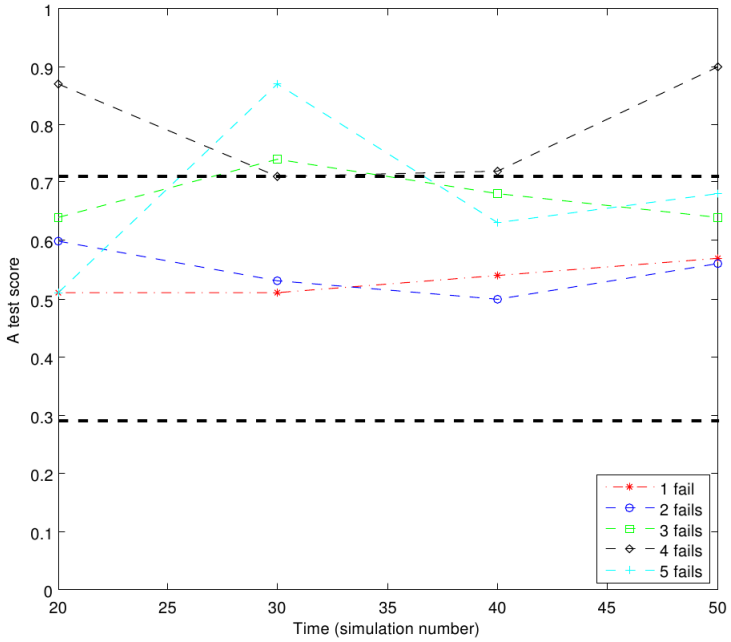


Figure A.1: A test score for the single nearest charger algorithm with different sample size (simulation runs). Each simulation run will be compared with a 10 simulation run as the baseline experiments. The dotted lines indicate the A test score of 0.29 and 0.71 where the large difference in the distribution lies (Vargha & Delaney, 2000).

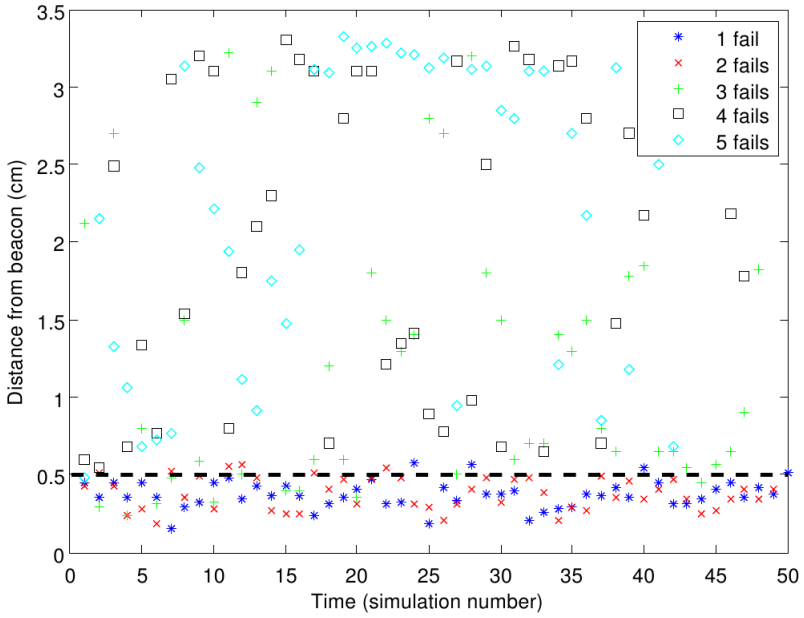


Figure A.2: The distance of swarm from the beacon with single nearest charger algorithm. With one and two failing robots the swarm can reach the beacon. Meanwhile, with three and more failing robots the swarm does not reach the beacon.

for twenty to fifty simulation runs. From figure A.3, with one failing robot in the experiment, the A scores values lies between 0.54 to 0.56. This shows that there is a small difference in the distibution for one failing robot in the system. When there are two, three and four failing robots, the A test scores for shared nearest charger algorithm are between 0.50 and 0.64, indicating that there are medium differences in the distribution. As the number of faulty robots increases to five, the value of A test scores increases from 0.58 to 0.69, showing that there are large differences in the distribution. All results for A test scores are listed in table A.2.

For the shared nearest charger algorithm, robots can reach 0.5 cm away from the beacon with one, two, three and sometimes four failing robots. Thus the magnitude effect for one, two, and three failing robots with different simulation runs are small to medium. This is further explained in figure A.4. With three and four failing robots in the environment, it has a medium magnitude effect to the results. This means that when there is three and four failing robots in the system, there is a possibility of the swarm to reach the beacon and there is also a possibility of the swarm not reaching the beacon. Meanwhile, there is again a large magnitude effect when there are five failing robots in the system showing that for all experiments with five failing robots the swarm is not reaching the beacon and stagnates at different position in the environment.

Table A.2: The magnitude of effect size indicated by A test score for different simulation runs for shared nearest charger algorithm. A test score with less than 0.56 (small) is marked with * which are seen in the case of one and two failing robots.

Simulation Number	20	30	40	50
A test score (1 fault)	0.52 *	0.54 *	0.56*	0.54 *
A test score (2 fault)	0.54 *	0.53 *	0.53 *	0.56 *
A test score (3 fault)	0.50 *	0.62 *	0.54 *	0.59
A test score (4 fault)	0.64	0.56 *	0.52	0.58
A test score (5 fault)	0.69	0.65	0.66	0.58

Table A.3 and figure A.5 show the A test scores for the granuloma formation algorithm for twenty to fifty simulation runs. From figure A.5, for one to two failing robots in the experiment, the A score values lie between 0.51 to 0.57, showing that there is a small difference in the distibution for one until four failing robots in the system. When there are five failing robots in the system, the A test scores for granuloma formation algorithm is between 0.53 and 0.61, indicating that there are medium differences in the distribution. All results for A test scores are listed in table A.3.

For granuloma formation algorithm, robots get to within 0.5 cm of the beacon with one, two, three, four and five failing robots. Thus the magnitude effect for one, two, and three, four and five failing robots with different simulation runs are small to medium. This

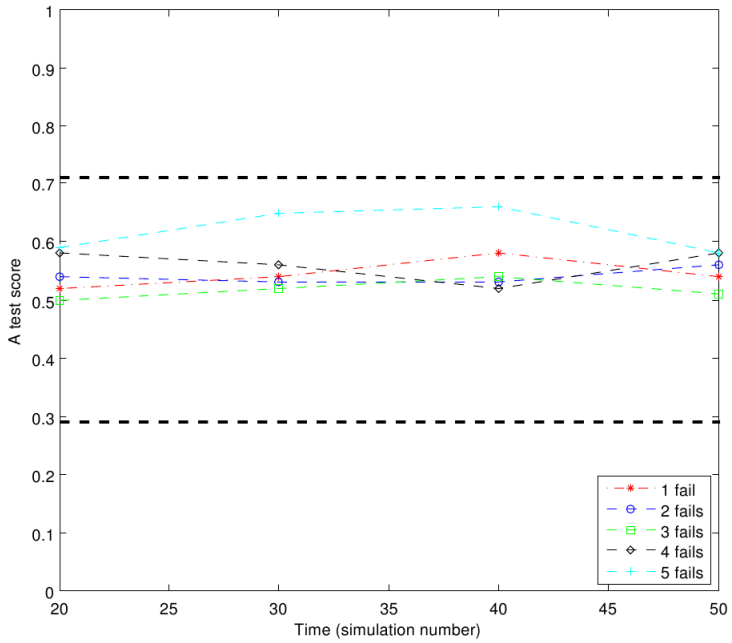


Figure A.3: A test score for the shared nearest charger algorithm with different sample size (simulation runs). Simulation will will be compared with a 10 simulation run as the baseline experiments. The dotted lines indicate the A test score of 0.29 and 0.71 where the large difference in the distribution lies (Vargha & Delaney, 2000).

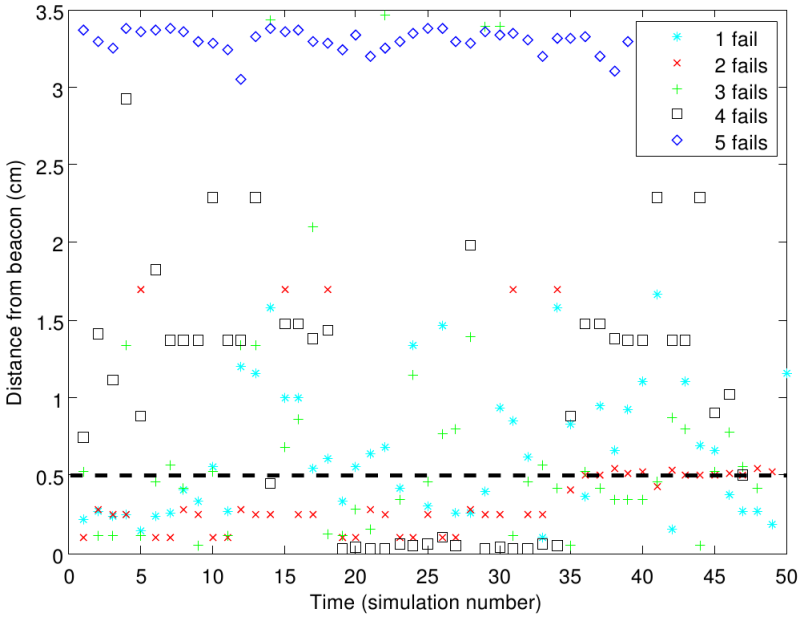


Figure A.4: The distance of swarm from the beacon with shared nearest charger algorithm. With one and two failing robots the swarm can reach the beacon. Meanwhile, with three and more failing robots the swarm does not reach the beacon.

is further explained in figure A.6. With one to five failing robots in the system, it has a small to medium magnitude effect on the results. This means that the swarm is able to reach the beacon when there are faulty robots in the environment.

Table A.3: The magnitude of effect size indicated by A test score for different simulation runs for granuloma formation algorithm. A test score with less than 0.56 (small) are marked with *

Simulation Number	20	30	40	50
A test score (1 fault)	0.56 *	0.53 *	0.59*	0.53 *
A test score (2 fault)	0.53 *	0.51 *	0.51 *	0.62
A test score (3 fault)	0.59	0.57	0.56 *	0.51 *
A test score (4 fault)	0.57 *	0.55 *	0.55 *	0.57
A test score (5 fault)	0.59	0.61	0.53	0.58

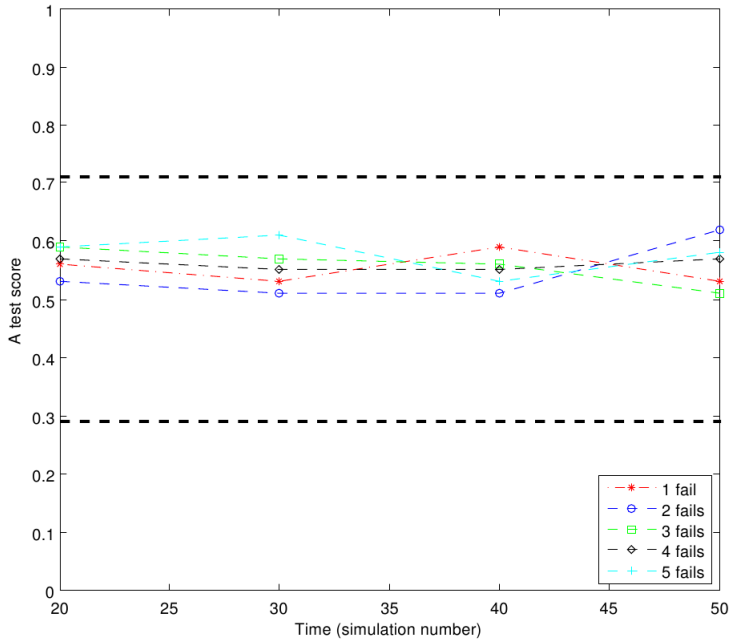


Figure A.5: A test score for the granuloma formation algorithm with different sample size (simulation runs). Each simulation runs will be compared with a 10 simulation run as the baseline.

A.3 Conclusion

The experiments presented in this chapter described the robustness analysis, which we used as a mechanism in determining that we did a sufficient number of simulation runs for our experiments in chapter 6. From the robustness analysis that we conducted in this

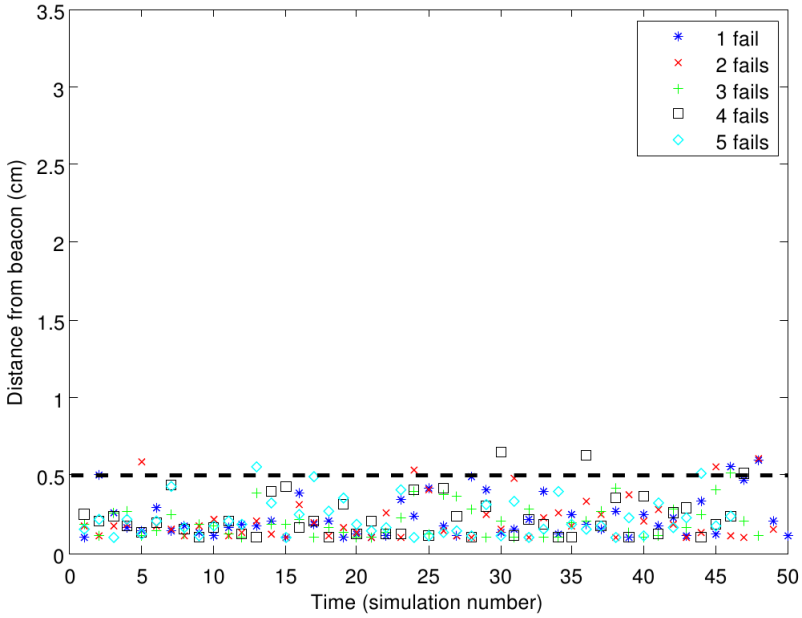


Figure A.6: The distance of swarm from the beacon with granuloma formation algorithm. In all cases with one, two, three,our and five failing robots the swarm is able to reach the beacon leading to the small to medium effect to the distribution.

chapter, we concluded that by doing ten simulation runs for all our experiments in chapter 6 the magnitude of difference was small to medium, which means that the results are significant when we compared from ten to fifty simulation runs.

Glossary

alveola macrophages an alveolar macrophage (or dust cell) is a type of macrophage found between the body and the outside world

antibody protein produced by B-cells that binds to antigens

antigen foreign substance that triggers a reaction from the immune system

bone marrow a substance in the cavities of bones where blood cells are produced

chemokines class of cytokines with various of immunoregulatory functions including attracting white blood cells to site of infections

epitheliod cells a cell derived from a macrophage often found in granulomas associated with tuberculosis

fibrosis the thickening and scarring of connective tissue usually as a result of injury

immunoglobulin protein produced by plasma cell that assist in destroying foreign substances such as bacteria

inflammation complex biological response of tissues to harmful stimuli such as pathogens (damaged cells or irritants)

lymph node	specialised immune tissue where immune response occur
macrophage	phagocytic cells that have a crucial role in host defence
monocytes	circulating (immature members) of the mononuclear phagocyte system
mononuclear phagocyte systems	a widely distributed system of free and fixed macrophages derived from bone marrow
mononuclear phagocytes	macrophages
necrosis	death of cells or tissues through injury or disease especially in a localised area of the body
neoplasms	an abnormal new growth of tissue in animals or plants such as a tumor
pathogen	microscopic organism that causes sickness such as bacteria and virus
phagocyte	a cell such as a white blood cell that engulfs and absorbs harmful microorganisms or foreign bodies in the bloodstream and tissues
primary immune organs	organs where immune cells develop
secondary immune organs	organs where immune responses occur

Bibliography

Abraham, A., Guo, H., & Liu, H. (2006). Swarm intelligence: Foundations, perspectives and applications. In N. Nedjah, & L. Mourelle (Eds.) *Studies in Computational Intelligence*, vol. 26, (pp. 3–25). Springer.

Adami, C. (1998). *Introduction to Artificial Life*. Springer.

Adams, D. O. (1974). The structure of mononuclear phagocytes differentiating in vivo. *American Journal of Pathology*, *76*, 17–48.

Adams, D. O. (1976). The granulomatous inflammatory response. a review. *American Journal of Pathology*, *84*(1), 164–191.

Aickelin, U., & Greensmith, J. (2008). Sensing danger: Innate immunology for intrusion detection. *Information Security Technical Report*, *12*(4), 218–227.

Andrews, P., & Timmis, J. (2005). On diversity and artificial immune systems: Incorporating a diversity operator into AINET. In *Proceedings of the 16th Italian Workshop on Neural Networks (WIRN 2005) and International Workshop on Natural and Artificial Immune Systems (NAIS 2005)*, vol. 3931 of *LNCS*, (pp. 293–306). Springer.

Andrews, P. S. (2008). Ph.d. thesis, An investigation of a methodhology for the development of Artificial Immune Systems: A case study in immune receptor degeneracy. Department of Computer Science. University of York, UK.

Andrews, P. S., Polack, F., Sampson, A. T., Timmis, J., Scott, L., & Coles, M. (2008). Simulating biology: towards understanding what the simulation shows. In *Proceedings of the 2008 Workshop on Complex System Modelling and Simulation*, (pp. 93–123). Luniver Press.

Andrews, P. S., Polack, F. A. C., Sampson, A. T., Stepney, S., & Timmis, J. (2010). Tech. Rep. The cosmos process, version 0.1: A process of the modelling and simulation of complex systems, University of York.

Andrews, P. S., Timmis, J., Owens, N. D. L., Aickelin, U., Hart, E., Hone, A., & Tyrrell, A. M. (Eds.) (2009). *Proceedings of the 8th International Conference on Artificial Immune Systems (ICARIS 2009)*. Springer.

Arvin, F., Samsudin, K., & Ramli, A. R. (2009). Swarm robots long term autonomy using moveable charger. In *Proceedings of the 2009 International Conference on Future Computer and Communication (ICFCC '09)*, (pp. 127–130). IEEE.

Auska, B. J., Grossa, T. S., & Srinivasan, S. (2006). An agent based model for real-time signaling induced in osteocytic networks by mechanical stimuli. *Journal of Biomechanics*, *39*(14), 2638–46.

Babaoglu, O., Canright, G., Deutsch, A., Di Caro, G. A., Ducatelle, F., Gambardella, L. M., Ganguly, N., Jelasity, M., Montemanni, R., Montresor, A., & Urnes, T. (2006). Design patterns from biology for distributed computing. *ACM Transactions on Autonomous and Adaptive Systems*, *1*(1), 26–22.

Bayindir, L., & Şahin, E. (2007). A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering and Computer Science*, *15*(2), 115–147.

Beni, G. (2005). From swarm intelligence to swarm robotics. In *International Workshop of Swarm Robotics (WS 2004)*, vol. 3342 of *LNCS*, (pp. 1–9). Springer.

Bentley, P. J., Lee, D., & Jung, S. (Eds.) (2008). *Proceedings of the 7th International Conference on Artificial Immune Systems (ICARIS 2008)*. Springer.

Bernardino, H., Barbosa, H., & Fonseca, L. (2010). A faster clonal selection algorithm for expensive optimization problems. In *Proceedings of the 9th International Conference on Artificial Immune Systems (ICARIS 2010)*, vol. 6209 of *LNCS*, (pp. 130–143). Springer.

Bersini, H. (2006). Immune system modeling: The OO way. In *Proceedings of the 5th International Conference on Artificial Immune Systems (ICARIS 2006)*, vol. 4163 of *LNCS*, (pp. 150–163). Springer.

Bersini, H., & Carneiro, J. (Eds.) (2006). *Proceedings of the 5th International Conference on Artificial Immune Systems (ICARIS 2006)*. Springer.

Bhalla, U. S., & Iyengar, R. (2001). Robustness of the bistable behavior of a biological signaling feedback loop. *Chaos*, *11*(1), 221–226.

Bjerknes, J., & Winfield, A. (2010). On fault-tolerance and scalability of swarm robotic systems. In *Proceedings of the 10th International Symposium on Distributed Autonomous Robotic (DARS 2010)*, vol. 1 of *Springer Tracts in Advanced Robotics Series*, (pp. 1–12). Springer.

Bjerknes, J. D. (2009). Ph.d. thesis, Scaling and fault tolerance in self-organized swarms of mobile robots. University of the West of England, Bristol, UK.

Bjerknes, J. D., Winfield, A. F., Melhuish, C., & Lane, C. (2007). An analysis of emergent taxis in a wireless connected swarm of mobile robots. In *Proceedings of IEEE Swarm Intelligence Symposium*, (pp. 45–52). IEEE Press.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.

Bornberg-Bauer, E., & Paton, N. W. (2002). Conceptual data modelling for bioinformatics. *Briefings in Bioinformatics*, *3*(2), 166–180.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraula, G., & Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton University Press.

Castro, L. N. d., Von Zuben, F. J., & Knidel, H. (Eds.) (2007). *Proceedings of the 6th International Conference on Artificial Immune Systems (ICARIS2007)*. Springer.

Coelho, G., & Von Zuben, F. (2006). omni-aiNet: An immune-inspired approach for omni optimization. In *Proceedings of the 5th International Conference on Artificial Immune Systems (ICARIS 2006)*, vol. 4163 of *LNCS*, (pp. 294–308). Springer.

Cohen, I. R. . (2000). *Tending Adams Garden: Evolving the Cognitive Immune Self*. Academic Press.

Cohen, I. R. (2006). Immune system computation and the immunological homunculus. In *Proceedings of the 9the International Conference on Model Driven Engineering Languages and Systems (MoDELS)*, vol. 4199 of *LNCS*, (pp. 499–512). Springer.

Cohen, I. R. (2007). Modeling immune behavior for experimentalists. *Immunological Reviews*, *216*(1), 232–236.

Cook, D., Farley, J., & Tapscott, S. (2001). A basis for a visual language for describing, archiving and analyzing functional models of complex biological systems. *Genome Biology*, *2*(4), 12.1–12.10.

Cutello, V., Narzisi, G., Nicosia, G., & Pavone, M. (2005). Clonal selection algorithms: A comparative case study using effective mutation potentials. In *Proceedings of the 4th International Conference on Artificial Immune Systems (ICARIS 2005)*, vol. 4163 of *LNCS*, (pp. 13–28). Springer.

Davoudani, D., & Hart, E. (2008). Computing the state of specknets: an immune inspired approach. In *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2008)*, (pp. 52–59). IEEE Press.

Davoudani, D., Hart, E., & Paechter, B. (2007). An immune-inspired approach to speckled computing. In *Proceedings of the 6th international conference on Artificial Immune Systems (ICARIS 2007)*, no. 4628 in LNCS, (pp. 288–299). Springer.

Davoudani, D., Hart, E., & Paechter, B. (2008). Computing the state of specknets: Further analysis of an innate immune-inspired model. In *Proceedings of the 7th international conference on Artificial Immune Systems (ICARIS 2008)*, no. 5132 in LNCS, (pp. 95–106). Springer.

de Castro, L. N., & Timmis, J. (2002). *Artificial Immune Systems: A new computational intelligence paradigm.*. Springer Verlag.

de Castro, L. N., & Zuben, F. J. V. (2000a). The clonal selection algorithm with engineering applications. In *Proceedings of the Workshop on Artificial Immune Systems and their Applications (GECCO 2000)*, (pp. 36–37). Morgan Kaufmann.

de Castro, L. N., & Zuben, F. J. V. (2000b). An evolutionary immune network for data clustering. In *Proceedings of the 6th Brazilian Symposium on Neural Networks (SBRN 2000)*, (pp. 84 – 89). IEEE.

de Castro, L. N., & Zuben, V. F. J. (2002). Learning and optimization using the clonal selection principle. *IEEE Transition on Evolutionary Computation*, *6*(3), 239–251.

de Lemos, R., Timmis, J., Ayara, M., & Forrest, S. (2007). Immune-inspired adaptable error detection for automated teller machines. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *37*(5), 873 –886.

Demir, E., Babur, O., Dogrusoz, U., Gursoy, A., Nisanci, G., Cetin-Atalay, R., & Ozturk, M. (2002). Patika: An integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics*, *18*(7), 996–1003.

Deneubourg, J., Aron, S., Goss, S., & Pasteels, J. M. (1990). The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, *3*(2), 159–168.

d'Inverno, M., & Prophet, J. (2005). Multidisciplinary investigation into adult stem cell behavior. 3737, (pp. 49–64).

Dori, D., & Choder, M. (2007). Conceptual modeling in systems biology fosters empirical findings: The MRNA lifecycle. *PLoS ONE*, *2*(9), 872.

Dorigo, M. (1991). Ph.d. thesis, Optimisation learning and natural algorithms (in italian). Dipartimento di Electronica, Politecnico di Milano.

Dorigo, M., & Birattari, M. (2007). Swarm intelligence. *Scholarpedia*, *2(9)*, 1462.

Dorigo, M., Birattari, M., & Sttzle, T. (2006). Ant colony optimization artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, *1*, 28–39.

Dorigo, M., Maniezzo, V., Colorni, A., Dorigo, M., Dorigo, M., Maniezzo, V., Maniezzo, V., Colorni, A., & Colorni, A. (1991). Tech. Rep. Positive feedback as a search strategy, Dipartimento di Elettronica, Politecnico di Milano, Milan Italy.

Eberhart, R. C., Shi, Y., & Kennedy, J. (2001). *Swarm Intelligence*. Morgan Kaufmann.

Efroni, S., Harel, D., & Cohen, I. R. (2003). Toward rigorous comprehension of biological complexity: Modeling, execution, and visualization of thymic t-cell maturation. *Genome Research*, *13*(11), 2485–2497.

Efroni, S., Harel, D., & Cohen, I. R. (2005). Reactive animation: Realistic modeling of complex dynamic systems. *Computer*, *38*(1), 38–47.

El-Ghalayini, H., Odeh, M., & McClatchey, R. (2006). Deriving conceptual data models from domain ontologies for bioinformatics. In *Proceedings of the 2nd International Conference on Information and Communication Technologies (ICTTA 2006)*, vol. 2, (pp. 3562–3567). IEEE.

Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity*, *4*(5), 41–60.

Facco, M., Miorin, M., Agostini, C., & Semenzato, G. (2007). Granuloma formation. *Diffuse Parenchymal Lung Disease*, *36*, 87–100.

Farmer, J. D., Packard, N. H., & Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Physica D*, *2*(1-3), 187–204.

Flügge, A. J., Timmis, J., Andrews, P., Moore, J., & Kaye, P. (2009). Modelling and simulation of granuloma formation in visceral leishmaniasis. In *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, (pp. 3052–3059). IEEE Press.

Forbes, N. (2000). Biologically inspired computing. *Computing in Science and Engineering*, *2*(6), 83–87.

Forrest, S., & Beauchemin, C. (2007). Computer immunology. *Immunological Reviews*, *216*(1), 176–197.

Forrest, S., Perelson, A. S., Allen, L., & Cherukuri, R. (1994). Self-nonself discrimination in a computer. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, vol. 1, (pp. 202–212). IEEE Press.

Fowler, M. (2004). *UML Distilled*. Addisson-Wesley.

Freitas, A., & Timmis, J. (2007). Revisiting the foundations of artificial immune systems for data mining. *IEEE Transaction on Evolutionary Computation*, *11*(4), 521–540.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Garrett, S. M. (2005). How do we evaluate artificial immune systems? *Evolutionary Computation*, *13*(2), 145–177.

Gerkey, B., Vaughan, R. T., & Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR 2003)*, (pp. 317–323). IEEE Press.

Gilbert, N. (2008). *Agent-based models*. SAGE Publications.

Goldsby, R., Kindt, T. J., Osborne, B. A., & Kuby, J. (2003). *Immunology*, (5th ed.). Blackwell Publishing.

González, F., Dasgupta, D., & Kozma, R. (2002). Combining Negative Selection and Classification Techniques for Anomaly Detection. In *Proceedings of the Congress on Evolutionary Computation*, (pp. 705–710). IEEE Press.

González, F. A., & Dasgupta, D. (2003). Anomaly detection using real-valued negative selection. *Genetic Programming and Evolvable Machines*, *4*(4), 383–403.

Greensmith, J. (2007). Ph.d. thesis, The dendritic cell algorithm. School of Computer Science, University of Nottingham.

Gurney, K. (1997). *An Introduction to Neural Network*. CRC Press.

Hart, E., Bersini, H., & Santos, F. C. (2007). How affinity influences tolerance in an idiotypic network. *Journal of Theoretical Biology*, *249 (3)*, 422–436.

Hart, E., Bersini, H., & Santos, F. C. (2010a). Structure versus function: a topological perspective on immune networks. *Natural Computing*, *9*(3), 603–624.

Hart, E., & Davoudani, D. (2009). Dendritic cell trafficking: From immunology to engineering. In *Proceedings of the 8th International Conference on Artificial Immune Systems (ICARIS 2009)*, vol. 5666 of *LNCS*, (pp. 11–13). Springer.

Hart, E., & Davoudani, D. (2011). An engineering-informed modelling approach to ais. In *Proceedings of the 10th International Conference of Artificial Immune Systems (ICARIS 2011)*, vol. 6825 of *LNCS*. Springer.

Hart, E., McEwan, C., Timmis, J., & Hone, A. (Eds.) (2010b). *Proceedings of the 9th International Conference on Artificial Immune Systems (ICARIS 2009)*. Springer.

Hart, E., & Timmis, J. (2008). Application areas of AIS: The past, the present and the future. *Applied Soft Computing*, *8*(1), 191–201.

Heemskerk, K. W., M., & Pavao-Zuckerman, M. (2003). Conceptual models as tools for communication across disciplines. *Conservation Ecology*, *7(3)*(8), 1–8.

Hinchey, M. G., Sterritt, R., & Rouff, C. (2007). Swarms and swarm intelligence. *Software Technologies*, *40*(4), 111–113.

Hsieh, M., Halász, Á., Berman, S., & Kumar, V. (2008). Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, *2*(2), 121–141.

Ismail, A. R., & Timmis, J. (2009). Aggregation of swarms for fault tolerance in swarm robotics using an immuno-engineering approach. In *Proceedings of the 9th Annual Workshop on Computational Intelligence (UKCI 2009)*.

Ismail, A. R., & Timmis, J. (2010). Towards self-healing swarm robotic systems inspired by granuloma formation. In *Special Session: Complex Systems Modelling and Simulation, part of ICECCS 2010*, IEEE, (pp. 313 – 314).

Ismail, A. R., & Timmis, J. (2011). Modelling containment mechanisms in the immune system for application in engineering. In *Proceedings of the 10th International Conference of Artificial Immune Systems (ICARIS 2011)*, vol. 6825 of *LNCS*, (pp. 77–80). Springer.

Ismail, A. R., Timmis, J., Bjerknes, J. D., & Winfield, A. F. T. (2011). An immune-inspired swarm aggregation algorithm for self-healing swarm robotic systems. *Swarm Intelligence*. (submitted for review).

Jackson, L. J., Trebitz, A. S., & Cottingham, K. L. (2000). An introduction to the practice of ecological modeling. *BioScience*, *50*(8), 694–706.

Jacob, C., Pilat, M., Bentley, P., & Timmis, J. (Eds.) (2005). *Proceedings of the 4th International Conference on Artificial Immune Systems (ICARIS 2005)*. Springer.

Janeway, C. A., & Medzhitov, R. (2002). Innate immune recognition. *Annual Review of Immunology*, *20*(1), 197–216.

Janeway, C. A., Travers, P., Walport, M., & Shlomchik, M. J. (2005). *Immunobiology*, (6th ed.). Garland Science Publishing.

Jevtić, A., & Andina, D. (2007). Swarm intelligence and its applications in swarm robotics. In *Proceedings of the 6th WSEAS international conference on computational intelligence, man-machine systems and cybernetics*, CIMMACS'07, (pp. 41–46).

Ji, Z., & Dasgupta, D. (2007). Revisiting negative selection algorithms. *Evolutionary Computation*, *15*(2), 223–251.

Johnson, C. G., Goldman, J. P., & Gullick, W. J. (2004). Simulating complex intracellular processes using object-oriented computational modelling. *Progress in Biophysics and Molecular Biology*, *86*(3), 1–40.

Kelsey, J., Henderson, B., Seymour, R., & Hone, A. (2008). A stochastic model of the interleukin (il)-1 network. In *Proceedings of the 7th International Conference on Artificial Immune Systems (ICARIS 2008)*, vol. 5132 of *LNCS*, (pp. 1–11). Springer.

Kelsey, J., & Timmis, J. (2003). Immune inspired somatic contiguous hypermutation for function optimisation. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2003)*, (pp. 207–218). Springer.

Kennedy, J., & Eberhart, R. C. (2001). *Swarm Intelligence*. Morgan Kaufmann.

Kernbach, S., Scholz, O., Harada, K., Popesku, S., Liedke, J., Raja, H., Liu, W., Caparrelli, F., Jemai, J., Havlik, J., Meister, E., , & Levi1, P. (2010). Multi-robot organisms: State of the art. In *IEEE International Conference on Robotics and Automation (ICRA-2010), workshop on "Modular Robots: State of the Art"*, (pp. 1–10). IEEE Press.

Kitano, H. (2003). A graphical notation for biochemical networks. *BIOSILICO*, *1*(5), 169–176.

Kitano, H. (2004). Biological robustness. *Nature Reviews Genetic*, *5*(9), 826–837.

Kitano, H. (2007). Towards a theory of biological robustness. *Molecular Sysytems Biology*, *3*(137), 1–7.

Kitano, H., Funahashi, A., Matsuoka, Y., & Kanae., O. (2005). Using process diagrams for the graphical representation of biological networks. *Nature Biotechnology*, *23*, 961–966.

Kleinstein, S. H., & Seiden, P. E. (2000). Computer simulations: Simulating the immune system. *Computing in Science and Engineering*, *2*(4), 69–77.

Klipp, E., Liebermeister, W., Helbig, A., Kowald, A., & Schaber, J. (2007). Systems biology standards-the community speaks. *Nature Biotechnology*, *25*(4), 390–391.

Kohn, K. W. (1999). Molecular interaction map of the mammalian cell cycle control and dna repair systems. *Molecular Sysytems Biology*, *10*(8), 2703–2734.

Kohn, K. W. (2001). Molecular interaction maps as information organizers and simulation guides. *Chaos*, *11*(1), 84–97.

Lehman, R. S. (1978). *Computer simulation and modeling*. John Wiley & Sons.

Liu, W., Winfield, A., Sa, J., Chen, J., & Dou, L. (2007). Strategies for energy optimisation in a swarm of foraging robots. In *Proceedings of the 2nd International Conference on Swarm Robotics (SAB 2006)*, (pp. 14–26). Springer.

Maimon, R., & Browning, S. (2001). Diagrammatic notation and computational structure of gene networks. In *Proceedings of the Second International Conference on Systems Biology*, vol. 1, (pp. 311–317). Omnipress.

Mansury, Y., Kimura, M., Lobo, J., & Deisboeck, T. S. (2002). Emerging patterns in tumor systems: Simulating the dynamics of multicellular clusters with an agent-based spatial agglomeration model. *Journal of Theoretical Biology*, *219*(3), 343–370.

Matzinger, P. (2002). The danger model: a renewed sense of self. *Science*, *296*(5566), 301–305.

McFarland, D., & Spier, E. (1997). Basic cycles, utility and opportunism in self-sufficient robots. *Robotics and Autonomous Systems*, *20*, 179–190.

Melhuish, C., & Hoddell, O. H. S. (1998). Collective sorting and segregation in robots with minimal sensing. In *Proceedings of the 5th International Conference on Simulation of Adaptive Behavior on From animals to animats 5*, (pp. 465–470). MIT Press.

Melhuish, C., Holland, O., & Hoddell, S. (1999). Convoying: using chorusing to form travelling groups of minimal agents. *Journal of Robotics and Autonomous Systems*, *28*(2-3), 206–217.

Melhuish, C., & Kubo, M. (2007). Collective energy distribution: Maintaining the energy balance in distributed autonomous robots using trophallaxis. In R. Alami, R. Chatila, & H. Asama (Eds.) *Distributed Autonomous Robotic Systems 6*, (pp. 275–284). Springer.

Merkle, D., Middendorf, M., & Schmeck, H. (2000). Ant colony optimization for resource-constrained project scheduling. In *IEEE Transactions on Evolutionary Computation*, (pp. 893–900). IEEE Press.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press.

Mokhtar, M., Bi, R., Timmis, J., & Tyrrell, A. M. (2009). A modified dendritic cell algorithm for on-line error detection in robotic system. In *Proceedings of the 11th conference on Congress on Evolutionary Computation (CEC 2009)*, (pp. 2055–2062). IEEE Press.

Muñoz Meléndez, A., Sempé, F., & Drogoul, A. (2002). Sharing a charging station without explicit communication in collective robotics. In *Proceedings of the 7th International Conference on Simulation of Adaptive Behavior on From animals to animats*, (pp. 383–384). MIT Press.

Neal, M., & Labrosse, F. (2004). Rotation-invariant appearance based maps for robot navigation using an artificial immune network algorithm. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, vol. 1, (pp. 863–870). IEEE Press.

Nembrini, J. (2005). Ph.d. thesis, Minimalist coherent swarming of wireless networked autonomous mobile robots. University of the West of England, Bristol, UK.

Nembrini, J., Winfield, A., & Melhuish, C. (2002). Minimalist coherent swarming of wireless networked autonomous mobile robots. In *Proceedings of the 7th International Conference on Simulation of Adaptive Behavior: From animals to animats (SAB 2002)*, vol. 7, (pp. 373 – 382). MIT Press.

Nicosia, G., Cutello, V., Bently, P., & Timmis, J. (Eds.) (2004). *Proceedings of the 3rd International Conference on Artificial Immune Systems (ICARIS 2004)*. Springer.

Oates, R., Greensmith, J., Aickelin, U., Garibaldi, J., & Kendall, G. (2007). The application of a dendritic cell algorithm to a robotic classifier. In *Proceedings of the 6th International Conference on Artificial Immune Systems (ICARIS 2007)*, vol. 4628 of *LNCS*, (pp. 13–28). Springer.

Owens, N., Timmis, J., Greensted, A., & Tyrell, A. (2007). On immune inspired homeostasis for electronic systems. In *Proceedings of the 6th International Conference on Artificial Immune Systems (ICARIS 2007)*, vol. 4628 of *LNCS*, (pp. 216–227). Springer.

Owens, N., Timmis, J., Greensted, A., & Tyrrell, A. (2008). Modelling the tunability of early T Cell signalling events. In *Proceedings of the 7th International Conference on Artificial Immune Systems (ICARIS 2008)*, vol. 5132, (pp. 12–23). Springer.

Paton, N. W., Khan, S. A., Hayes, A., Moussouni, F., Brass, A., Eilbeck, K., Goble, C. A., Hubbard, S. J., & Oliver, S. G. (2000). Conceptual modelling of genomic information. *Bioinformatics*, *16*(6), 548–547.

Polack, F. A. C. (2010). Self-organisation for survival in complex computer architectures. In *Proceedings of the 1st International Conference on Self-organizing Architectures (SOAR 2009)*, (pp. 66–83). Springer.

Polack, F. A. C., Hoverd, T., Sampson, A. T., Stepney, S., & Timmis, J. (2008). Complex systems models: Engineering simulations. In *Proceedings of the 11th International Conference of the Simulation and Synthesis of Living Systems (Artificial Life XI)*, (pp. 482–489). MIT Press.

Read, M., Timmis, J., & Andrews, P. S. (2008). Empirical investigation of an artificial cytokine network. In *Proceedings of the 7th International Conference on Artificial Immune Systems (ICARIS 2008)*, vol. 5132 of *LNCS*, (pp. 340–351). Springer.

Read, M., Timmis, J., Andrews, P. S., & Kumar, V. (2009a). A domain model of experimental autoimmune encephalomyelitis. In *Proceedings of the 2nd Workshop on Complex Systems Modelling and Simulation (CoSMoS 2009)*, (pp. 9–44). Luniver Press.

Read, M., Timmis, J., Andrews, P. S., & Kumar, V. (2009b). Using UML to model EAE and its regulatory network. In *Proceedings of the 8th International Conference on Artificial Immune Systems (ICARIS 2009)*, vol. 5666 of *LNCS*, (pp. 4–6). Springer.

Şahin, E. (2005). Swarm robotics: From sources of inspiration to domains of application. In *Proceedings of the International Workshop of Swarm Robotics (WS 2004)*, vol. 3342 of *LNCS*, (pp. 10–20). Springer.

Şahin, E., & Winfield, A. (2008). Special issues on swarm robotics. *Swarm Intelligence*, *2*(2-4), 69–72.

Sandor, M., Weinstock, J. V., & Wynn, T. A. (2003). Granulomas in schistosome and mycobacterial infections: a model of local immune responses. *Trends in Immunology*, *24*(1), 44–52.

Sandor, M., Weinstock, J. V., & Wynn, T. A. (2004). Response to doenhoff: Granulomas: these gizmos are cool! *Trends in immunology*, *24*(4), 169–170.

Sargent, R. G. (1985). An exposition on verification and validation of simulation models. In *Proceedings of the 17th Winter Simulation Conference*, (pp. 15–22). ACM.

Sargent, R. G. (1986). The use of graphical models in model validation. In *Proceedings of the 18th Winter Simulation Conference*, (pp. 237–241). ACM.

Sarraf, P., & Sneller, M. C. (2005). Pathogenesis of wegener's granulomatosis: current concepts. *Expert Reviews in Molecular Medicine*, *7*(8), 1–19.

Seeley, T. D. (2002). When is self-organization used in biological systems? *Biological Bulletin*, *202*, 314–318.

Segovia-Juarez, J. L., Ganguli, S., & Kirschner, D. (2004). Identifying control mechanisms of granuloma formation during m. tuberculosis infection using an agent-based model. *Theoretical Biology*, *231*, 357–376.

Shegogue, D., & Zheng, W. J. (2005). Object-oriented biological system integration: a SARS coronavirus example. *Bioinformatics*, *21*(10), 2502–2509.

Sheng, W., Yang, Q., Tan, J., & Xi, N. (2006). Distributed multi-robot coordination in area exploration. *Robotics and Autonomous Systems*, *54*, 945–955.

Signorini, J., & Greussay, P. (2004). A case study of object-oriented bio-chemistry. *Genome Informatics*, *14*(6), 14–22.

Sneller, M. C. M. (2002). Granuloma formation, implications for the pathogenesis of vasculitis. *Cleveland Clinic Journal of Medicine*, *69*(2), 40–43.

Socha, K., Knowles, J., & Sampels, M. (2002). A max-min ant system for the university course timetabling problem. In *Proceedings of the 3rd International Workshop on Ant Algorithm (ANTS 2002)*, LNCS, (pp. 1–13). Springer.

Socha, K., Sampels, M., & Manfrin, M. (2003). Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In *In Proceeding of the Third European Workshop on Evolutionary Computation in Combinatorial Optimization (EvoCOP 2003)*, (pp. 334–345). Springer.

Soysal, O., Bahçeci, E., & Sahin, E. (2007). Aggregation in swarm robotic systems: Evolution and probabilistic control. *Turkish Journal of Electrical Engineering and Computer Science*, *15*(2), 199–225.

Stepney, S., Smith, R. E., Timmis, J., Tyrrell, A. M., Neal, M. J., & Hone, A. N. W. (2005). Conceptual frameworks for artificial immune systems. *International Journal of Unconventional Computing*, *1*(3), 315–338.

Stibor, T., & Timmis, J. (2007). An investigation into the compression quality of AINET. In *Proceedings of Congress On Evolutionary Computation*, vol. 3102, (pp. 491–498). IEEE.

Timmis, J., Andrews, P., & Hart, E. (2010a). On artificial immune systems and swarm intelligence. *Swarm Intelligence*, *4*(4), 247–273.

Timmis, J., Andrews, P., Owens, N., & Clark, E. (2008a). An interdisciplinary perspective on artificial immune system. *Evolutionary Intelligence*, *1*(1), 5–26.

Timmis, J., & Bently, P. (Eds.) (2002). *Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS 2002)*. University of Kent Printing Unit.

Timmis, J., Bently, P., & Hart, E. (Eds.) (2003). *Proceedings of the 2nd International Conference on Artificial Immune Systems (ICARIS 2003)*. Springer.

Timmis, J., & Edmonds, C. (2004). A comment on opt-AINet: an immune network algorithm for optimisation. In *Proceedings of the International Conference on Genetic and Evolutionary Computation*, vol. 3102 of *LNCS*, (pp. 308–317). Springer.

Timmis, J., Hart, E., Hone, A., Neal, M., Robins, A., Stepney, S., & Tyrrell, A. (2008b). Immuno-engineering. In *2nd IFIP International Conference on Biologically Inspired Collaborative Computing, 20th IFIPWorld Computer Congress*, (pp. 3–17). IEEE Press.

Timmis, J., Hone, A., Stibor, T., & Clark, E. (2008c). Theoretical advances in artificial immune systems. *Theoretical Computer Science*, *403*(1), 11–32.

Timmis, J., & Neal, M. (2001). A resource limited artificial immune system for data analysis. *Knowledge Based Systems*, *14*(3-4), 121–130.

Timmis, J., Tyrrell, A., Mokhtar, M., Ismail, A. R., Owens, N., & Bi, R. (2010b). An artificial immune system for robot organisms. In P. Levi, & S. Kernback (Eds.) *Symbiotic Multi-Robot Organisms: Reliability, Adaptability and Evolution*, vol. 7, (pp. 279–302). Springer.

Tsankova, D., Georgieva, V., Zezulka, F., & Bradac, Z. (2007). Immune network control for stigmergy based foraging behaviour of autonomous mobile robots. *International Journal of Adaptive Control and Signal Processing*, *21*(2-3), 265–286.

Vargha, A., & Delaney, H. D. (2000). Critique and improvement of cl common language effect size statistics of mcgraw and wong. *Journal of Educational and Behavioral Statistics*, *25 (2)*, 101–132.

Vaughan, R. (2008). Massively multi-robot simulation in stage. *Swarm Intelligence*, *2*(2), 189–208.

Vaughan, R., & Gerkey, B. (2007). Reusable robot software and the player/stage project. In D. Brugali (Ed.) *Software Engineering for Experimental Robotics*, vol. 30, (pp. 267–289). Springer.

Walker, J. H., & Garrett, S. M. (2003). Dynamic function optimisation: Comparing the performance of clonal selection and evolution strategies. In *Proceedings of the Second International Conference on Artificial Immune Systems (ICARIS 2003)*, vol. 2787 of *LNCS*, (pp. 273–284). Springer.

Whitbrook, A., Aickelin, U., & Garibaldi, J. (2008). Idiotypic immune networks in mobile robot control. In *in Proceedings of the 7th International Conference on Artificial Immune Systems (ICARIS 2008)*, vol. 5132 of *LNCS*, (pp. 266–278). Springer.

White, J. A., & Garrett, S. M. (2003). Improved pattern recognition with artificial clonal selection? In *Proceedings of the 2nd International Conference on Artificial Immune Systems (ICARIS 2003)*, vol. 2787 of *LNCS*, (pp. 181–193). Springer.

Williams, G. C. (1966). *Adaptation and Natural Selection*. Princeton University Press.

Winfield, A. F. (2009a). The potential and the challenges of swarm robotics in safety critical applications. In *Invited Keynote talk at Robots for Risky Interventions and Environmental Surveillance (RISE 09)*.

Winfield, A. F. (2009b). Towards an engineering science of robot foraging. In J. O. H. Asama, H. Kurokawa, & K. Sekiyama (Eds.) *Distributed Autonomous Robotic Systems*, vol. 8, (pp. 185–192.). Springer.

Winfield, A. F., Liu, W., Nembrini, J., & Martinoli, A. (2008). Modelling a wireless connected swarm of mobile robots. *Swarm Intelligence*, *2*(1), 241–266.

Winfield, A. F., & Nembrini, J. (2006). Safety in numbers: fault tolerance in robot swarms. *International Journal of Modelling, Identification and Control*, *1*(1), 3–37.

Yagmahan, B., & Yenisey, M. M. (2008). Ant colony optimization for multi-objective flow shop scheduling problem. *Computers & Industrial Engineering*, *54*(3), 411 – 420.