An Investigation into XSets of Primitive Behaviours for Emergent Behaviour in Stigmergic and Message Passing Ant-like Agents

A thesis submitted in fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

of

RHODES UNIVERSITY
Computer Science Department

by

COLIN CHIBAYA

March 2014

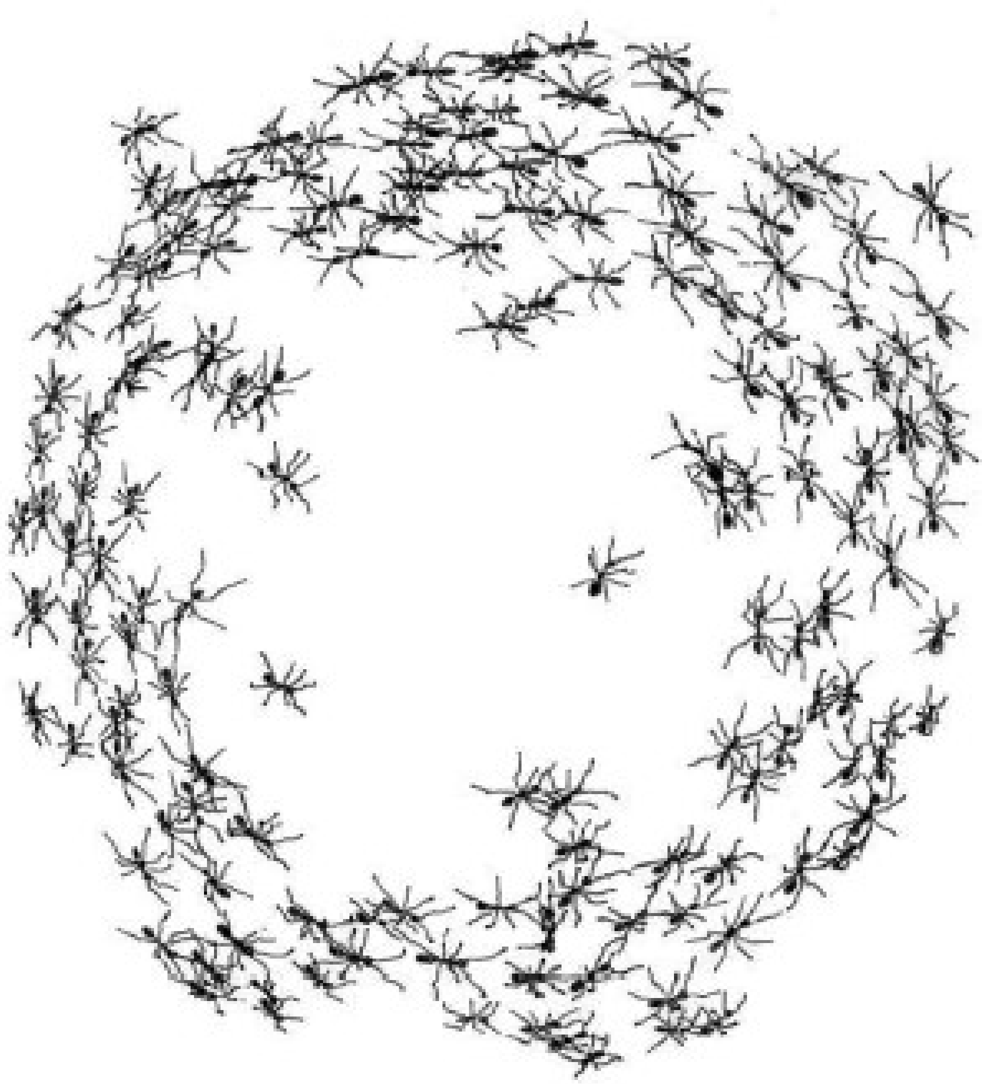
"….but ants, birds, nor fish have no leaders.

Their groups aren't organized that way.

Nobody directs the individual members of the groups.

Nor are the individual members genetically programmed for swarm intelligence.."

Michael Crichton, "Prey", 2002.



What knowledge does each individual ant in the group have which causes swarm level intelligence?

"Making small things count"

(See separate title page)

Colin Chibaya

10th April 2014

Abstract

Ants are fascinating creatures - not so much because they are intelligent on their own, but because as a group they display compelling emergent behaviour (the extent to which one observes features in a swarm which cannot be traced back to the actions of swarm members). What does each swarm member do which allows deliberate engineering of emergent behaviour?

We investigate the development of a language for programming swarms of ant agents towards desired emergent behaviour. Five aspects of stigmergic (pheromone sensitive computational devices in which a non-symbolic form of communication that is indirectly mediated via the environment arises) and message passing ant agents (computational devices which rely on implicit communication spaces in which direction vectors are shared one-on-one) are studied.

First, we investigate the primitive behaviours which characterize ant agents' discrete actions at individual levels. Ten such primitive behaviours are identified as candidate building blocks of the ant agent language sought. We then study mechanisms in which primitive behaviours are put together into XSets (collection of primitive behaviours, parameter values, and meta information which spells out how and when primitive behaviours are used). Various permutations of XSets are possible which define the search space for best performer XSets for particular tasks.

Genetic programming principles are proposed as a search strategy for best performer XSets that would allow particular emergent behaviour to occur. XSets in the search space are evolved over various genetic generations and tested for abilities to allow path finding (as proof of concept). XSets are ranked according to the indices of merit (fitness measures which indicate how well XSets allow particular emergent behaviour to occur) they achieve. Best performer XSets for the path finding task are identified and reported.

We validate the results yield when best performer XSets are used with regard to normality, correlation, similarities in variation, and similarities between mean performances over time. Commonly, the simulation results yield pass most statistical tests.

The last aspect we study is the application of best performer XSets to different problem tasks. Five experiments are administered in this regard. The first experiment assesses XSets' abilities to allow multiple targets location (ant agents' abilities to locate continuous regions of targets), and found out that best performer XSets are problem independent. However both categories of XSets are sensitive to changes in agent density. We test the influences of individual primitive behaviours and the effects of the sequences of primitive behaviours to the indices of merit of XSets and found out that most primitive behaviours are indispensable, especially when specific sequences are prescribed. The effects of pheromone dissipation to the indices of merit of stigmergic XSets are also scrutinized. Precisely, dissipation is not causal. Rather, it enhances convergence.

Overall, this work successfully identify the discrete primitive behaviours of stigmergic and message passing ant-like devices. It successfully put these primitive behaviours together into XSets which characterize a language for programming ant-like devices towards desired emergent behaviour. This XSets approach is a new ant language representation with which a wider domain of emergent tasks can be resolved.

Acknowledgements

This work would have taken forever without the intellectual guidance of my supervisor Professor Shaun Bangay, who tirelessly and unconditionally mentored me all the way from the start. I submit to his expertise and overall understanding of this field of research.

I want to thank the members of the Virtual Reality Special Interest Group (VRSIG) at Rhodes University (2005 - 2010) for providing a stimulating environment in which to work. Your critical advice and many hours of discussions made much of this work possible. Thanks to the Jindo team, particularly Shaun, Dennis, and Michael. Your highly valued contributions to the success of the Jindo project and this work are noted with honour.

My family rightfully deserves an applause too. Thank you Elamo, Onil, Victor, and Shaun for allowing me time out to work in isolation when I needed to.

I also want to thank the proof readers, the statistician, examiners, and volunteer readers whose feedback and comments added value to this work.

To Professor George Wells, thank you for accepting the responsibility of seeing me through the final stages of this work. To Dr Desmond Govender, thank you for your moral support and allowing me time-out to work on this project.

I acknowledge the generous sponsorship of equipment and subsidy by the Distributed Multimedia Centre of Excellence at Rhodes University, the National Research Foundation who funded the Jindo Project, Rhodes University Research office who paid the biggest part of my tuition through staff remission of fees, and the University of Kwazulu-Natal who chipped in with incentives and tuition to complete this project. Thank you all for your financial support.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Ant agent systems are successfully used in many areas of simulated construction (Werfel et al., 2006), search (Dorigo, 1992.; Dorigo et al., 1996), data mining and clustering (De Wolf, 2007), as well as for optimization purposes (Bonabeau et al., 1999). The collective effects of the individual activities of ant agents produce complex emergent behaviour at swarm levels with no central swarm organization at individual levels (Dorigo et al., 1999.; Dorigo et al., 2006.; Poon and Maher, 1996). What does each individual ant agent do in the swarm which allows deliberate engineering of emergent behaviour?

This thesis investigates the development and formalization of a language for programming swarms of ant agents towards deliberate engineering of desired emergent behaviour. In this context, *emergent behaviour* is the extent to which one can observe features in a swarm which cannot be traced back to the low level activities of the individual agents in a swarm (Fisher and Lipson, 1999.; Krink and Vollrath, 1998). In fact, the emergent behaviour that is observed in a swarm is more than the sum of the contributions of the individual agents in the swarm (Sato and Matsuoka, 2009.; Priesterjahn et al., 2005.; Stepney et al., 2007.; Negulescu and Barbat, 2004).

To achieve its goals, this thesis starts by investigating the low level activities of computational ant-like devices that are modelled on the behaviours of simulated ant agents. We refer to low level ant agent activities as *primitive behaviours* (a term derived from the works of Andersson et al. (2002) and Cao et al. (1997), where agent activities e.g. *wandering, following, aggregation, dispersion, and homing* - are referred to as communication primitives).

Successful identification of primitive behaviours, and appropriate combination of these primitive behaviours into sets of activities which characterize each ant agent's behaviour over time poses a number of challenges. Which primitive behaviours define ant agents' activities? How are these primitive behaviours put together in order to define an ant agent language? Even harder is to understand the *meta information* which presents the rules and conditions regarding how and when each primitive behaviour is used in order to guarantee deliberate engineering of predictable emergent behaviour at swarm levels. In this context, meta information relates to information about the use and functionality of primitive behaviours in a given context.

This work refers to a collection of primitive behaviours, parameter values of these primitive behaviours, and meta information, as an *XSet* (an acronym for "e**X**tended **Set**"). Thus, the primary goal of this thesis, and the key contribution thereof, is the identification of XSets which best describe a "*language*" for programming swarms of ant-like devices towards deliberate engineering of desired and predictable emergent behaviour.

Successful identification and formalization of XSets as a language for programming swarms of ant-like devices towards predictable emergent configuration has three benefits to the field of computer science. First, it extends knowledge in emergent systems, swarm algorithms, and agent coordination systems. It also extends the application of techniques from evolutionary programming and wireless sensor networks to ant systems. Most importantly, the results of this work have direct implications to future researches towards practical application of ant swarm architectures in commercial problem domains. The next section presents a succinct overview which mainly introduces the key concepts of the thesis.

1.1 Background

This section presents a brief background of the research problem we investigate in this thesis (which is the development and formalization of a language for programming swarms of ant-like devices towards deliberate engineering of desired emergent behaviour).

We build this work on the knowledge of researches that have been presented in the past whose aims have been to achieve predictable emergent behaviour from simulated swarms of similar agents (Burke and Kendall, 1999.; Geer et al., 2003.; Nagpal, 2001.; Nagpal et al., 2002.; Kondacs, 2003.; Mason, 2002.; Werfel, 2002.; Rothemund, 2006). This field remains an interesting research area in computer science for a number of reasons such as; the desire to simplify the creation of deterministic emergent behaviours; ambitions to eliminate the unpredictable nature of emergent behaviour in distributed systems comprising of many individual functional units, in favour of directed and purposeful design of global problem solving emergent behaviours; the hope to create and formalize agent "languages" for achieving desired emergent behaviour; the hope to reduce the costs of developing emergent systems; and the desire to achieve even more robust and fault tolerant agent based solutions with, potentially, practical applications.

This thesis particularly pays attention to the design of ant agents that can demonstrate most of the advantages that are mentioned in the previous paragraph. Most importantly, we emphasize on the design of ant agent languages that can allow deliberate engineering of desired emergent behaviour. However, we acknowledge that the concepts, methods, and theories that can help in identifying the key primitive behaviours that would sufficiently characterize ant agent XSets for these purposes can be found from a number of disciplines such as; entomology, artificial life, general biology, or other previously simulated ant systems. It is an ambitious task to try and investigate the viewpoints of all these possible disciplines regarding the primitive behaviours sought. As a case study and proof of concept, we focus our investigations on the views that are implied when ant-like computational devices or simulated ant agents are proposed.

We described an *ant agent* as a computational device that is modelled on the behaviour of simulated ant systems (see the preamble of this chapter, in the third paragraph for this description). A group of ant agents is described as a *swarm*. Each ant agent in a swarm is designed with abilities to follow a clear set of rules (collection of *primitive behaviours)* in order to collectively contribute towards the creation of desired emergent behaviour. In theory, a primitive behaviour characterizes an ant agent's discrete action in one movement step. Some examples of ant agent primitive behaviours that have been proposed or implied in the literature include (for illustration purposes); *dropping pheromone* (Fernandes et al., 2005.; Panait and Luke, 2004b), *agent orientation* (Chibaya and Bangay, 2007.; Panait and Luke, 2004b)*, agent movement* (Cerello eta al., 2010)*, flipping between different internal states* (Panait and Luke, 2004b. Chibaya and Bangay, 2007)*,* and *sharing vectors* (Nasipuri and Li, 2002.; Payton et al., 2001).

A combination of a collection of primitive behaviours with meta information defines an *XSet*. Our assumption is that an XSet encapsulates four pieces of information namely; the primitive behaviours that are required by ant agents; the parameter values of the primitive behaviours; conditions which stipulate how and when primitive behaviours are used; meta data (regarding the cardinality of XSets, agent memory, and number of internal states), and the sequences in which primitive behaviours are presented to ant agents.

Ant agent languages have been represented in many ways (see the categorization of agent interaction techniques in Chapter 2). This work regards the XSets approach as a foundation towards the development of a language for programming swarms of ant agents towards deliberate engineering of desired emergent behaviour. However we restrict our interest to the primitive behaviours of two categories of ant agents namely; stigmergic and message passing ant agents. Stigmergic ant agents support indirect and environment mediated interactions in which virtual pheromone chemicals are the key ingredients for ant agent *orientation* and *movement* (Dorigo et al., 1999). On the other hand, message passing ant agents support direct agent-to-agent interactions in which implicit communication spaces arise. A potential extension of this work arises when the primitive behaviours of other forms of ant agents (such as talking ants, leader following or queen managed ants, etc) are studied and incorporated into the search space we achieve in this thesis (thus extending the search space for best performer XSets, as well as increasing the chances of achieving generic combination of primitive behaviours).

Ant agents are designed with awareness of their low level task domains at all times (e.g. ant agents are always aware that they are exploring the environment or they are navigating, recruiting, or updating information). An ant agent's knowledge of its task domain defines its *internal state*, and such knowledge connotes ant agents as having some basic memory in which to hold internal state information.

An intriguing feature regarding stigmergic ant agents is that they *drop* specific levels of pheromones in every movement step (Panait and Luke, 2004a, 2004b). The levels of pheromones thereof mark trails with directional cues to other ant agents in the swarm. Thus, the levels of pheromones that are placed on the environment create shared memories for the swarm. On the other hand, adjacent ant agents in the message passing category can explicitly *exchange* directional information in the form of geometric vectors that are weighted by set confidence parameters (see Chapter 3 for details regarding these confidence parameters). This background and the assumptions we make in this section and in the preamble, motivate the need for a clear statement of the research question that is addressed in this thesis.

1.2 Problem statement

Literature informs us that simulated ant agents are guided by defined sets of rules (Werfel, 2002.; Cerello et al., 2010). Our work refers to these sets of rules as XSets. It is that explicit identification of the elements of XSets that is of interest to us because emergent behaviour is generally difficult to predict when we do not know its building blocks. Even harder would be an attempt to decompose emergent behaviour into component units, and hopefully characterize each unit in terms of its contribution to the emergent behaviour reported (Funes et al., 2003).

The particular research statement we address in this thesis is: *an investigation into XSets of primitive behaviours that can allow deliberate engineering of emergent behaviour in swarms of ant-like agents*. This is an ambitious task in an extensive emerging field, hence we constrain our investigations to five sub-problems, all of which contribute towards achieving our overall goal.

Figure 1.1. shows the breakdown of our research problem into these five sub-problems as follows: (a) the identification of ant agent primitive behaviours in the two categories we study, (b) devising strategies for putting primitive behaviours together in order to create XSets, (c) describing methods for evaluating and ranking XSets based on the outcomes of the swarms that use XSets for particular purposes, (d) validating the measures of emergence that

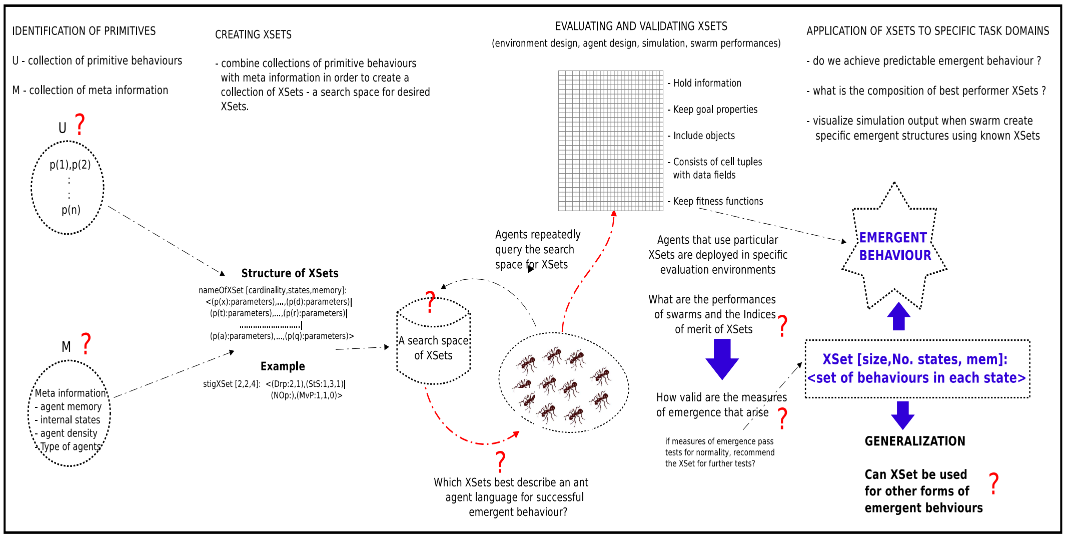


Figure 1.1: Generalization of the research problem and the concept of XSets

arise when particular XSets are used, and (e) application of XSets to different problem domains.

While these sub-problems may not address the very general ant agent programming problem, we believe that the XSets sought will provide a solid foundation for the development of, at least, a basic "language" for programming swarms of ant agents towards desired emergent behaviours. The next five subsections describe and motivate each sub-problem in details.

1.2.1 Identification of primitive behaviours

We can re-phrase this sub-problem as follows: *what are the low level activities of ant agents that can be used to describe the domain of primitive behaviours that allow emergent behaviour to occur?* Generally, primitive behaviours are viewed as agent instructions which cause emergent behaviour. However in computational terms, these are, in fact, parametrized routines which code ant agent controls, rules and conditions over time.

To identify these primitive behaviours and answer this sub-problem, we harness lessons learned from the cooperative behaviour of simulated foraging ant agents (Panait and Luke, 2004a, 2004b, 2004c). Chapter 3 addresses this sub-problem and reports the identification of ten candidate primitive behaviours namely: (a) pheromone based orientation, (b) vector based orientation, (c) dropping levels of pheromones, (d) pheromone evaporation, (e) pheromone diffusion, (f) normalizing vectors, (g) detecting target indicators and converting these to vectors, (h) agent movement, (i) switching between different internal states, and (j) the no action control.

Chapter 3 also describe meta information which define system level parameters such as (a) the type of ant agents that are supported in the work, (b) the number of internal states swarms of ant agents support, (c) the design and amount of memory an ant agent supports, (d) as well as agent density limits that would achieve meaningful results in the environments set.

The key response to this sub-problem and the main outcome of Chapter 3 are therefore twofold; (a) the list of primitive behaviours, (b) the meta information about the primitive behaviours.

1.2.2 Creating XSets

The second sub-problem can be re-phrased as follows: *how do we represent XSets which can summarize sufficient collections of ant agent actions that would allow deliberate engineering of emergent behaviour?* This sub-problem requires us to present a mechanism with which to put primitive behaviours together in order to form useful XSets. In this work, genetic programming principles are proposed as a search strategy for evolving and identifying best collections of primitive behaviours and parameter values that would allow deliberate engineering of desired emergent behaviour.

First, we propose a mechanism for defining the initial genetic population which serves as the search domain for desired components of useful XSets. To achieve this, a set of candidate primitive behaviours U is formed (see Figure 1.1). Precisely, U is the set of the ten primitive behaviours that are listed in section 1.2.1 in paragraph 2. A power set $P(U)$ of U consists of all the possible subsets of U. We propose the definition of an ordered power set $P(U)$, where subsets are partial permutations of the ten primitive behaviours. For example, if $U = \{a, b\}$. Then the ordered power set would be $P(U) = \{\phi_0; \{a\}_1; \{b\}_2; \{a; b\}_3;\ \{b; a\}_4\}$. Thus, each element of $P(U)$ is a unique subset without repetition and has a unique index in the power set. In this context, subsets $\{a; b\}_3$ and $\{b; a\}_4$ are regarded as different since the composite primitive behaviours are presented in unique sequences.

A combination of every ordered subset set with meta information creates a member of the initial genetic population of XSets from which new generations of XSets are evolved over time (the search space for desired XSets). In this case, meta information spells out system level parameters such as the category of XSets sought at the time (stigmergic, message passing, or hybrid XSets), the highest cardinality of the XSets, the number of internal states ant agents support, the number of memory blocks ant agents can hold at a time, agent density, genetic parameters, as well as other environment definition parameters.

This sub-problem is addressed in Chapter 4. The key outcome of Chapter 4 is therefore; the creation of the initial genetic population, evolution of XSets, and the identification of best performer XSets for resolving *a particular case study scenario* (the path finding problem) which serves as a task domain for proving the concept of XSets.

1.2.3 Evaluating XSets

The initial genetic population of XSets that arises from partial permutations of primitive behaviours in the set U is the basis for the creation of a search space for best XSets for desired emergent behaviour. In this sub-problem, we present a mechanism in which to quantify the extent to which emergent behaviour is manifest as a result of using particular XSets. The quantity sought is referred to as an *index of merit* of an XSet. In other words, an index of merit is a value that is associated with the performances of swarms of ant agents that use an XSet for a particular purpose. It is the relative fitness rank of the XSet within a given genetic population of XSets.

We can re-phrase this sub-problem as follows: *how do we quantify the extent to which emergent behaviour is manifest as a result of using a particular XSet for a known purpose?* We also tackle this sub-problem in Chapter 4. Five measures of emergence are proposed with which we determine the indices of merit of XSets. These measures of emergence are: *speed of emergence*, *quality of emergence*, *average delivery rate*, *average end-to-end delays*, as well as *Shannon's information value.*

A mechanism which spells out how these five measures of emergence are combined in order to determine the index of merit is presented in the same Chapter 4. At the end of the chapter, we administer an experiment which validates the processes and the measures of emergence that arise when best performer XSets for the path finding task are used.

1.2.4 Validation of measures of emergence

The fourth sub-problem investigates four aspects with regards to the validity of the measures of emergence that are reported as relatively best. First, we assess whether the outcomes we report are normally distributed (evidence of normal distribution is tested using Kolmogorov - Smirnov tests on the sets of measures of emergence). Then we investigate the degrees of association that exist between different sets of measures of emergence that are reported from replicated runs when best performer XSets are used. In particular, correlation analyses between sets of measures of emergence, comparisons between the mean performances, as well as analyses of variances are the key statistics we report.

This sub-problem is addressed in Chapter 5. Results which report successful Kolmogorov-Smirnov tests for normality, significant correlations between sets of measures of emergence, similar variations and mean performances are presented in this Chapter. These results indicate that, generally, the indices of merit we see are indicative of the extent to which the XSets thereof, represent repeatable dictionaries for an ant agent language that can allow deliberate engineering of desired emergent behaviour - hence the recommendations we make of testing the same XSets on different task domains in the next sub-problem.

1.2.5 Application of XSets in different task domains

The last sub-problem investigates the extent to which path finding XSets can form useful toolboxes for allowing other forms of emergent behaviour to occur. Precisely, we investigate elements of these XSets which support multiple targets location. In the context of this work, multiple targets location is about deploying swarms of ant agents in order to locate, as a case study and proof of concept, continuous regions of targets.

The choice of the setups of the continuous regions of targets we propose is motivated for in Chapter 2. In particular, we investigate the performances of best path finding XSets for abilities to allow the location of continuous regions of targets that are arranged in cross patterns, four-way cross patterns, polygonal layouts (such as triangles, rectangles, pentagon, hexagon, heptagon, and octagon), or in the form of circular shapes. These are sufficiently many examples of different goal setups to reveal the emergent properties we want to investigate in best performer XSets. We address this sub-problem in Chapter 6 and report results which indicate that, generally, path finding XSets are problem independent.

1.3 Strategy

We address the five sub-problems in search of best performer XSets that can allow emergent properties to emerge in swarms of ant-like devices. The following are the steps we follow to achieve these goals:

1. First, we explore the literature in search of the key concepts, methods, and theories that would help us in describing stigmergic and message passing ant agents' primitive behaviours. Other concepts and theories are derived from observing electronic versions of simulated ant systems. To augment the views we get from these sources, we develop a prototype which simulates foraging ant agents that are tasked to find a food source that is situated in an environment, and upon finding this food source, return back to the starting point and commence the search journeys all over again. The key outcome of this search is a list of ant agent activities that are commonly inferred as ant agent control rules or conditions. Let a set of all ant agent activities that are found be denoted as U.

2. Second, we algebraically manipulate the set U in order to determine its power set and the initial genetic population of XSets. A power set

contains all possible subsets of U. Let the power set of U be denoted as $P(U)$.

3. Meta information is then considered, which spells out particular simulation parameters depending on the task domain that is being tested at the time. A combination of meta information with every element of $P(U)$ creates the initial genetic population of XSets which defines a search domain for best performer XSets we require.

4. Experiments are administered thereafter, which genetically evolve better and better XSets using particular genetic operations. In each case, the XSets are ranked by the indices of merit they achieve when swarms of ant agents are deployed to solve a given task domain. In the end, best performer XSets are identified and reported.

5. The performances of the same best performer XSets are validated with regards to normality, significance of correlations that arise between pairs of measures of emergence, as well as the similarities between the means and variances thereof.

6. The same XSets are then evaluated for abilities to allow other forms of emergent properties in different task domains.

This strategy spells out the steps we follow before arriving at a conclusion regarding the possibilities of generalizing particular XSets as control dictionaries for achieving predictable emergent behaviour in swarms of ant-like devices.

1.4 Motivation

This section presents our general motivation for undertaking this research, as well as motivation for the different choices we make throughout this research.

It particularly motivates our choice of studying the behaviours of stigmergic and message passing ant agents, and the choice we make of using the XSets approach over traditional swarm control techniques. We also justify why we mainly test our XSets on the path finding task domain, and conclude the section with a motivation for choosing specific measures of emergence as emergent quantifiers.

1.4.1 General motivation for this work

A key drive for undertaking this research work arises from the literature. Ant agent systems have shown benefits over other agent control systems in terms of efficiency, the speed and quality of services, robustness, and fault tolerance (Abelson et al., 2000). This is because ant agents use local interaction rules when solving problems. We understand local interaction rules as agent abilities to perceive environments only within their observable neighbourhood (De Wolf, 2007).

Inspired by these numerous benefits, we develop a software engineering paradigm that constructs solutions from swarms of interacting ant-like devices. At present, the accepted definition of emergent behaviour as a non-reducible phenomenon prevents the use of accepted software decomposition strategies to characterize this software engineering paradigm.

A number of examples of plausible ant systems products that demonstrate these benefits can be listed. For example, successful nest construction has been reported (Downing and Jeanne, 1988.; Aleksiev et al., 2007.; Franks et al., 1992). Other ant systems simulated molds of anthill-like structures (Werfel et al., 2005.; Werfel et al., 2006), while others achieved pit construction (Burgess, 2009) and brood tending behaviour (Detrain and Deneubourg, 2006.; Merkle et al., 2006). We have also previously demonstrated shortest path formation and following behaviour (Chibaya and Bangay, 2007.; Porta et al., 2009.; Solé et al., 2000). However object segmentation properties (Tao et al., 2007), shape construction (Liu and Mohamed, 2008.; Soloveichik, 2008.; Meinhardt, 1982.; Werfel, 2002.; Yamins, 2007), and image evolution (Fernandes et al., 2005.; Rezaee, 2008) are even more inspiring examples. The possibilities of developing ant based XSets with properties for achieving similar swarm intelligence are many and far-ranging.

Another motivating factor emanates from software engineering principles that are applied in the field of amorphous computing (Abelson et al., 2000). Amorphous computing is a program designing principle whose aim is to develop programming paradigms for achieving coherent behaviour from the cooperation of unreliable swarms of devices (Grochow, 2002.; Nagpal and Coore, 1998). In this context, the terms "devices" and "agents" are synonymous. We note that the accepted definition of amorphous computing systems (Irons and Monk, 2006) draws closely to our definition of emergent behaviour (Poon and Maher, 1996.; Seevinck and Edmonds, 2008). As a result, the XSets we propose may inspire further developments of computational constructs for amorphous computing.

In addition, the demand to formalize swarm coordination principles is even higher in software engineering methodologies for nanotechnology (North, 2006). Nanotechnology is a new science aimed at building structures atom by atom using tiny mobile robotic devices known as nanites (Cavalcanti and Freitas, 2005.; North, 2006.; Schneider et al., 2006). The hope of nanotechnology is that swarms of nanites, under the control of reliable controller XSets, would successfully self-organize into robust and fault tolerant objects at nano-meter scales (Treder, 2004). Our work therefore comes at the right time when contributions towards describing such controller XSets are sought.

The development of Micro-Electro-Mechanical Systems (MEMS) devices (Gage, 1993) has also inspired the development of this thesis. Although it is now possible to manufacture millions of MEMS devices and configure them, for example, into paintable materials (Butera, 2002.; Bullock and Cliff, 2004), formal control principles with which to guarantee predictable outcomes are still pending. Our hope is that the XSets we propose can be adapted for coordinating MEMS devices towards commercial configurations (Couzin and Franks, 2002).

Furthermore, calls for contributions towards building paradigms with which to coordinate smart ant colonies have been made in the past (Saenz, 2011). The key highlights of these calls are the demands for control routines that can tell ant agents to find paths, giving preferences to finding the resources, as opposed to wandering at random. The work of Winfield et al., (2013) also confirmed lack of practical examples, as yet, where collective foraging systems have been successfully employed in real-world applications. Our work directly addresses this problem. As such, the academic benefits of the results of this work, particularly to novice programmers in the field, make this research work worthwhile.

Given this wide range of motivating factors, governments around the world have began to show interest in funding related projects. In South Africa alone, the National Research Foundation (NRF) announced a provision to fund projects categorized as the Blue Skies Researches, aimed at developing novel, cutting edge, and speculative research ideas with potentials to shift disciplinary paradigms (Van Jaarsveld and Bozzoli, 2009). Our work falls into this category. As a result, the Jindo project at Rhodes University, funded by the NRF and Rhodes University Centre of Excellence, was established. These numerous funding opportunities are a motivating factor on their own.

1.4.2 Motivation for stigmergic/message passing swarms

Various agent control metaphors have been proposed in the literature. Dominant are agent control metaphors which support language based interactions (Nagpal et al., 2002.; Nagpal et al., 2003.; Sussman, 1999.; Belani et al., 2002.; Stefano and Santoro, 2001.; Kraus and Lehmann, 1995.; Nagpal and Coore, 1998.; Cao et al., 1997.; Butera, 2002.; Beal 2005a, 2005b.; Abelson et al., 2000.; Nagpal et al., 2006.; Cranefield et al., 2000). These metaphors attempt to develop agent languages with a vocabulary, full syntax, and semantics that are understood by the agents. However such language based models are generally impractical and unrealistic for application in the development of ant agent systems because there is lack of sufficient vocabulary for this purpose.

Other agent control metaphors heavily rely on the laws of mathematics, geometry, or physics as origins of agent control axioms (Ngo et al., 2005.; Harris, 2007.; Trofimova et al., 1998.; Spears et al., 2004a, 2004b, 2005.; Azzag et al., 2007.; Beckers et al., 1989.; Balch and Arkin, 1999.; Cao et al., 1997). However these metaphors are often economically not feasible for the development of simple and naive ant agents since they serve best when sophisticated robots are required.

A few agent systems consider the message passing metaphors (Trianni and Dorigo, 2005.; Rajbhupinder et al., 2010.; Hara and Ichimura, 2005.; Montes De Oca et al., 2005.; Lien et al., 2005.; Rodriguez et al., 2007.; Bayazit et al., 2002.; Bayazit et al., 2005.; Caicedo et al., 2001). Most of the works that have been presented in this regard make use of agents with substantial amount of memory capacities to hold blocks of messages and historic information that is shared one-on-one. However, vector arithmetic is a general algebra that is applicable for computation in many domains (from machine learning, network analysis, and spatial representation) so has potential to allow ants to solve problems relevant to these domains. In this case, it also allows location reasoning which is important for spatial problems. To the best of our knowledge, the message passing metaphor has not been sufficiently explored for application in the development of naive agents such as ant-like devices - hence the choice we make to investigate the primitive behaviours of ant agents in this category. Successful identification of primitive behaviours that characterize the activities of ant agents in this category would constitute novel findings in the field.

Most ant agent metaphors consider swarms in which interactions are environment mediated (Dorigo, 1992.; Dorigo et al., 1999.; Dorigo et al., 1996.; Di Caro et al., 2004.; Negulescu et al., 2006.; Chibaya and Bangay, 2007.; Montgomery et al., 2007.; Panait and Luke, 2004a, 2004b, 2004c.; Cavalcanti and Freitas, 2005.; Cavalcanti et al., 2005b). We take advantage of the availability of profound literature in which the key concepts, methods, and theories for describing and developing ant agent primitive behaviours are likely comprehensive - hence the choice we make to study stigmergic swarms.

It is an ambitious task to try and identify the primitive behaviours of all possible agent metaphors, especially in this preliminary study of the concept of XSets. As a case study and proof of concept, we restrict ourselves to the study of stigmergic (environment mediated) and message passing ant metaphors. The notion is that similar studies on other agent metaphors would only extend the set of candidate primitive behaviours, and increase the cardinality of the power set $P(U)$, as well as the diversity of the initial genetic population. That on its own would diversify the search space for even better XSets with a wider application domain.

1.4.3 Motivation for proposing the XSets method

The XSet method is a novel approach (invented by the author) for representing the syntax and semantics of a language for programming swarms of ant agents towards deliberate engineering of desired emergent behaviour. It creates a new data structure which encapsulates meta information, agent activities, and the key simulation parameters with the view of purporting some degree of generality that would potentially allow the same XSets to find solutions in a wide task domain. That theory we envision regarding the possibilities of generalizing the ant agent language thereof, to a wide task domain, is what inspires the choice of the XSets approach.

1.4.4 Motivation for using the path finding task domain

We are informed by the literature that environment exploration, ant interaction and recruitment, as well as path formation are pre-requisites for resolving most tasks in the ants inspired robotics domain (Nouyan and Dorigo, 2007.; Panait and Luke, 2004a, 2004b, 2004c.; Dorigo et al., 1999). This piece of information is, on its own, a motivating factor for testing our proposed XSets on the path finding problem since it is a pre-requisite for resolving other related tasks. The notion is that, if our XSets successfully demonstrate emergent properties in this respect, then they can serve as building blocks for adapted XSets with which we can create related swarm configurations with, potentially, practical benefits.

On the other hand, we also learn from related works that most ACO algorithms are initially tested (alpha, beta, and functional tests) on the path formation or foraging problem (Winfield et al., 2013.; Dorigo, 1992.; Cordon et al., 2002.; Panait and Luke, 2004b). Tests on new approaches in ant systems are often mainly inspired by Dorigo (1992)'s first experiment in the field (the two-way bridge setup) which assesses path finding abilities in swarms. Our XSets approach is a new innovation in the field which requires testing from the first principles as well - hence the choice of the path finding task domain as proof of concept.

In addition, we indicated earlier on that the path formation problem is a well researched task domain where concepts, methods, and theories are likely profoundly documented. Thus, we have a wider search space for useful primitive behaviours and controls for developing the XSets we want.

However, it is important to note that the aim of this work is not to demonstrate path formation behaviour and compare the outcomes with known traditional ant system models. Rather, our concern is to be able to explicitly pinpoint the low level activities of ant agents in swarms which characterize a language for building desired emergent behaviour (such as path finding). Thus, the path finding domain (or any other task domain) is used only to demonstrate successful or unsuccessful identification of such XSets.

1.4.5 Motivation for specific measures of emergence

The ability to quantify emergent behaviour is another young research dimension in swarm intelligence systems. Only a few measures of emergence have been reported and formalized for this purpose so far. Our work looks at the merits and demerits of most of these measures of emergence and recommend those that are suitable for assessing emergency in ant systems.

Commonly, measures of emergence have been reported which study the relationships among events in simulation (Gore and Reynolds, 2008). A prevalent view in these measures of emergence is the determination of the *amount of change in a system (comparing the proportion of outputs to the amount of inputs)* (Schaefer et al., 2002.; Hinchey et al., 2005.; Rouff et al., 2004.;

Hamann et al., 2011.; Chan, 2011). We can re-phrase this measure of emergence as a measurement of the throughput of a system. Our choice to assess the *average delivery rates* in swarms translates to an evaluation of system throughput as well. Thus, our choice of this measure of emergence is directly inspired by similar works in the literature.

There are emergent systems in which the *frequency of agent interaction* are assessed (Noble and Letsky, 2002.; Wang and Zhu, 2007). Other works evaluate the *time a system takes to converge* (Wang and Zhu, 2007.; Hovda, 2008.; Minati, 2002.; Chan, 2011). This work interprets time in simulation in two contexts (*speed of emergence* and *average end-to-end delays* of ant agents in the swarm). Thus, the choices we make of these two measures of emergence are also informed by related works in the literature.

In other cases, the *quality of the products* yield determines the extent to which emergent behaviour was achieved (Noble and Letsky, 2002). This work adopts this metric and re-phrases it to the *quality of emergence*. The last measure of emergence we consider takes from Shannon's information theories (Bavaud et al., 2005.; Martin, 2006.; Schneider, 2007). This is, in a way, similar to quantifying emergent behaviour in terms of the amount of change in a system (Schaefer et al., 2002). However this work treats the measure of information differently (see Chapter 4 for this aspect of the thesis).

A few more measures of emergence exist in the literature. However most of these involve rigorous mathematics which would defeat our goal of designing and using simple rules and naive ant-like devices. Some examples of these mathematically biased measures of emergence include *correlation analysis* (Wang and Zhu, 2007.; Valverde et al., 2006), *axiomatic geometry* (Sumpter et al., 2001.; Parunak and VanderBok, 1997.; Bonabeau et al., 1999.; Haglich et al., 2010.; Shalizi, 2001.; Grossman et al., 2009), *formal reasoning* (Sumpter et al., 2001.; Fulbright and Stephens, 2003.; Wang and Zhu, 2007.; Tofts, 1991), and *cognitive analysis* (Crutchfield, 1994).

1.5 Contributions

While that singular most important substantial and original contribution of this thesis is the identification of XSets which characterize a software engineering paradigm for constructing solutions from swarms of interacting ant-like devices, the value of this thesis is further emphasized by a number of incremental contributions, both from an academic, practical, and general point of views.

On the academic side, we demonstrate the following contributions:

1. Successful identification of the primitive behaviours which characterize ant agents' behaviours at individual levels, which give rise to particular forms of emergent behaviour at swarm levels, is a big milestone in the study of ant systems. If we get to know what each ant agent does as an individual, then we can control swarms of similar ant-like devices to produce desired and predictable emergent behaviour at global levels. Success in this regard creates new and relevant knowledge in the field, particularly to the benefit of future researches and studies in the area.

2. We present a creative mechanism in which primitive behaviours are combined with meta information in order to form XSets that can characterize necessary and sufficient rules for controlling swarms of ant agents. This mechanism may inspire the development of useful emergent based object assemblers with practical and commercial impacts.

3. Representation of primitive behaviours in computational terms (in the form of algorithms) is innovative (see Chapter 3 for these algorithms). The abstract implementation we show in each case creates relevant content with which the ant agent programming problem is further addressed.

4. Available literature lacks clarity with regards to how we can detect and quantify ant based emergent behaviour. We propose five innovative measures of emergence with which we can determine the extent to which emergent behaviour is manifest as a result of using a particular XSets (see Chapter 4 for details regarding these measures of emergence). In addition, the properties of most of these measures of emergence make them suitable for verifying other different forms of emergent behaviours that are achieved in similar swarm configurations. Thus, we open up new research avenues in the field.

5. The processes we follow when we validate the results we achieve when a a particular XSet is used (determining correlations, comparing means and variances) are innovative. Similar validation techniques may inspire further developments in other swarm intelligence models.

On the practical side, we present the following contributions:

1. The ability to explicitly specify ant agents' primitive behaviours at individual levels, and combine these into XSets that are useful at swarm levels, has direct relevancy to many fields in science. Swarms of simulated ant-like devices such as nanites, amorphous devices, or MEMS devices, can be deployed in similar simulation environments using similarly designed XSets and create commercially attractive emergent structures. Thus, the results of this work may potentially inspire industrial and commercial developments.

2. Our emphasis on specificity, both in terms of the XSets sought, as well as in terms of the emergent behaviour thereof, changes the way we see and think of the consequences of upcoming sciences such as nanotechnology (Joy, 2000). Generally, nanotechnology is feared that, one day, nanites may aggregate into unpredictable emergent formations that are

disastrous to nature and life (Joy, 2000). The XSets we propose may guarantee predictable nanites outcomes. Success in this regard may even inspire new developments in nano-medicine, nano-construction, and other related fields.

From a general point of view, we make the following contributions:

1. Although the thesis does not solve the very general agent programming problem, it provides a working baseline upon which further investigations in the field may arise. This work provides a solid foundation for investigations aimed at identifying more primitive behaviours with which the ant agent programming solutions can be generalized.

2. Many ant systems that exist in the literature do not explicitly present the white-box side of the routines ant agents follow in computational terms. As a result, the domain of the solutions that are presented in the literature is currently limited, especially for commercial recommendations. Our emphasis on specificity, and explicit description of the routines that characterize each ant agent behaviour may inspire the development of a wider range of ant based solutions.

1.6 Notation

This section introduces the notation and syntax we use to represent primitive behaviours and XSets. We explain the semantics of each of the ten primitive behaviours we identify and, at the end of this section, provide an arbitrary representation of an XSet.

primitive-behaviours: although, in general, primitive behaviours are ant agent activities at individual levels, we view them as computational routines with parameter values and code which spells out what ant agents should do at a particular point in time. Below are the ten primitive behaviours we study, emphasizing on the mnemonics we use, as well as the parameter values they take:

- ($\mathbf{MvH}$: $p_1, p_2, ..., p_n, w_1, w_2, ...w_n$) - this is a stigmergic primitive behaviour for orientation. It supports multiple levels of pheromones p_i, each weighted by a specific attractiveness value w_i. The primitive behaviour is used to determine a direction an ant agent would follow next based on the levels of pheromones around the ant agent. Different levels of pheromones p_i are assigned integer aliases, while related weights w_i are assigned float aliases. Target indicators, in this case, are regarded as different levels of pheromones, thus bearing integer aliases as well. For example, the primitive behaviour - ($\mathbf{MvH}$: $1, 2, 3, 0.5, 0.5, -0.8$) tells a stigmergic ant agent to favour movements towards higher concentrations of target indicator 1 with a 50% chance of success, while at the same time biasing its movements towards locations that contain relatively higher levels of pheromone 2 whose attractiveness value is set to 50%. The same ant agent should however penalize movements towards locations which contain relatively higher levels of pheromone 3 whose weight, at the moment, is set to a negative value.
- ($\mathbf{MsP}$: v_c, v_j, l_j) - this is an orientation primitive behaviour in the message passing category. This primitive behaviour presents the vector components that can be shared at the time, as well as the levels of confidence thereof. Vectors and confidence levels are stored in separate memory blocks in the ant agent's internal state. For example, the primitive behaviour - ($\mathbf{MsP}$: $1, 1, 2$) tells a message passing ant agent to favour movements towards higher concentrations of target indicator 1, sharing geometric vectors that

are stored in memory block 1, whose related confidence levels are stored in memory block 2.

- (**MvP** : x_i, y_i, z_i) - this is a general primitive behaviour that is used by both categories of ant agents to relocate an ant agent to a specified location. The x and y values are unit offsets from the ant agent's current position. The z value is often always set to 0 because, in this case, we operate in $2D$ environments. However its inclusion in the routine is necessary in view of the future possible generalizations of the work to $3D$ environments. For example, the primitive behaviour - (**MvP** : $1, -1, 0$) tells an ant agent to relocate to one cell along the positive direction of the $x-axis$ and one cell along the negative direction of the $y - axis$ ($\searrow$).

- (**Drp** : p_i, q) - this is a stigmergic primitive behaviour which tells an ant agent in this category to drop specific levels of pheromones p_i in specific quantities q. For example, the primitive behaviour - (**Drp** : $1, 1$) tells a stigmergic ant agent to drop the levels of pheromone 1 in unit quantities.

- (**Evp** : α) - this is a pheromone dissipation control which when triggered evaporates, at a specified evaporation rate α, all types of the levels of pheromones that are on the environment at the time.

- (**Dfs** : α) - this control diffuses pheromones to neighbouring locations at a specified dissipation rate.

- (**Nrm** : v_x, v_y, v_z) - this is a message passing primitive behaviour with which the resultant vector and any other vectors are normalized. This is an important control which guarantees that related ant agents do not make unrealistic movement steps in simulation. It guarantees that ant agents move at a constant speed of one grid cell per step.

- ($\mathbf{PtV}$: p_i, x)- this primitive behaviour assesses the quantities of specific target indicator p_i around a message passing ant agent, and trigger appropriate actions. For example, ($\mathbf{PtV}$: $3, 0.5$) tells a message passing ant agent to detect the levels of pheromone 3 around, evaluating if these levels are above a threshold value 0.5. If so, the ant agent must drop its directional knowledge and follow the direction of the location which contains high levels of pheromone 3 (because hopefully the ant agent has arrived on target). The choice to follow this new direction sets the ant agent's confidence factor to the highest level possible .
- ($\mathbf{StS}$: m, n, x) - this is a general primitive behaviour which tells an ant agent to switch between different internal states when set conditions are met. The first parameter in this primitive behaviour indicates the ID of the new internal state an ant agent would switch to. The second and third parameters are components of the condition for switching from one internal state to another. For example, the primitive behaviour - ($\mathbf{StS}$: $2, 3, 1$) tells an ant agent to flip to internal state 2 if the levels of pheromone 3 are above 1 at the agent's current location.
- ($\mathbf{NOp}$:)- this instruction tells an ant agent to do nothing. In computer terms, it is a filler instruction which we use to complete the XSet template when fewer instructions are required than those proposed in the meta information.

U- Figure 1.1. uses U to symbolize the collection of primitive behaviours that are identified as building blocks of the desired XSets. In this thesis, $U = \{(NOp :); (StS : m, n, x); (MvH : p_1, p_2, ..., p_n, w_1, w_2, ...w_n); (MsP : v_c, v_j, l_j); (MvP : x_i, y_i, z_i); (Nrm : v_x, v_y, v_z); (Drp : p_i, q); (PtV : p_i, x); (Evp : \alpha); (Dfs : \alpha)\}$.

M - Figure 1.1 also uses M to indicate a collection of meta information that are required as initial simulation information, which includes agent memories, number of internal states, agent density, and agent types.

XSet - An XSet is mathematically defined as a combination of meta information and sets of primitive behaviours.

- In combining these two pieces of information, we first provide meta information regarding the category of ant agents that are preferred at the moment. There are three possibilities (stigmergic, message passing, or hybrid ant agents (which combine primitive behaviours from the other two categories).
- Three parameters follow which respectively indicate the cardinality of the XSet in each ant agent internal state, the number of internal states supported, and the number of memory blocks an ant agent can hold at a time. Each memory block can only hold one unit of data at a time.
- The rest of the entries that follow are elements of the sets of primitive behaviours in the the power set. In our representation, primitive behaviour and their parameters are enclosed in round brackets. Different primitive behaviours that are supported in the same XSet are separated by commas. A vertical bar separates the lists of primitive behaviours that are required in different internal states.
- The following is an example of the representation of an XSet (for illustration purposes). In this case, we illustrate the composition of an XSet that is marked as an $msgXSet$. The cardinality of the same XSet in each internal state is set to 6 primitive behaviours. Each ant agent can support 3 internal states. Ant agents can hold up to 8 memory blocks at a time.

msgXSet[6,3,8]: $< (MsP : 0,0,1), (PtV : 0,0), (Nrm : 0,0,0), (MvP : 0,0,0), (NOp :), (StS : 1,0,0) \mid (Drp : 4,1), (NOp :), (NOp :), (NOp :), (NOp), (StS : 2,0,0) \mid (MsP : 2,2,3), (PtV : 1,1), (Nrm : 2,2,2), (MvP : 2,2,2), (NOp :), (StS : 3,1,0) >$

Figure 1.2: An arbitrary representation of an XSet

1.7 Overview of the thesis

The chapters of the thesis, and the manner in which these chapters are related to each other, are arranged as shown in Figure 1.3, and are further outlined as follows:

- Chapter 2 reviews literature relating to the five sub-problems of this thesis. First, we take a survey of the various agent control mechanisms that have been proposed in the past, and position the ant agent control problem in the literature. This review mainly identifies the key concepts and methods that are useful in characterizing ant agents. In these reviews, we split ant agent systems into two categories namely: interactive and non-interactive control systems. We further divide each category into particular sub-groups depending on the mechanisms in which agent interactions are achieved (see chapter 2 for details). Then, the chapter discusses the key parameters of emergence in various agent control systems. Related work follows which describes how specific emergent behaviour has been successfully simulated in the past. Lastly, literature relating to the measures of emergence that have commonly been proposed in different scenarios is reviewed, closing the chapter with conclusions which highlight the contributions of the chapter to the thesis.

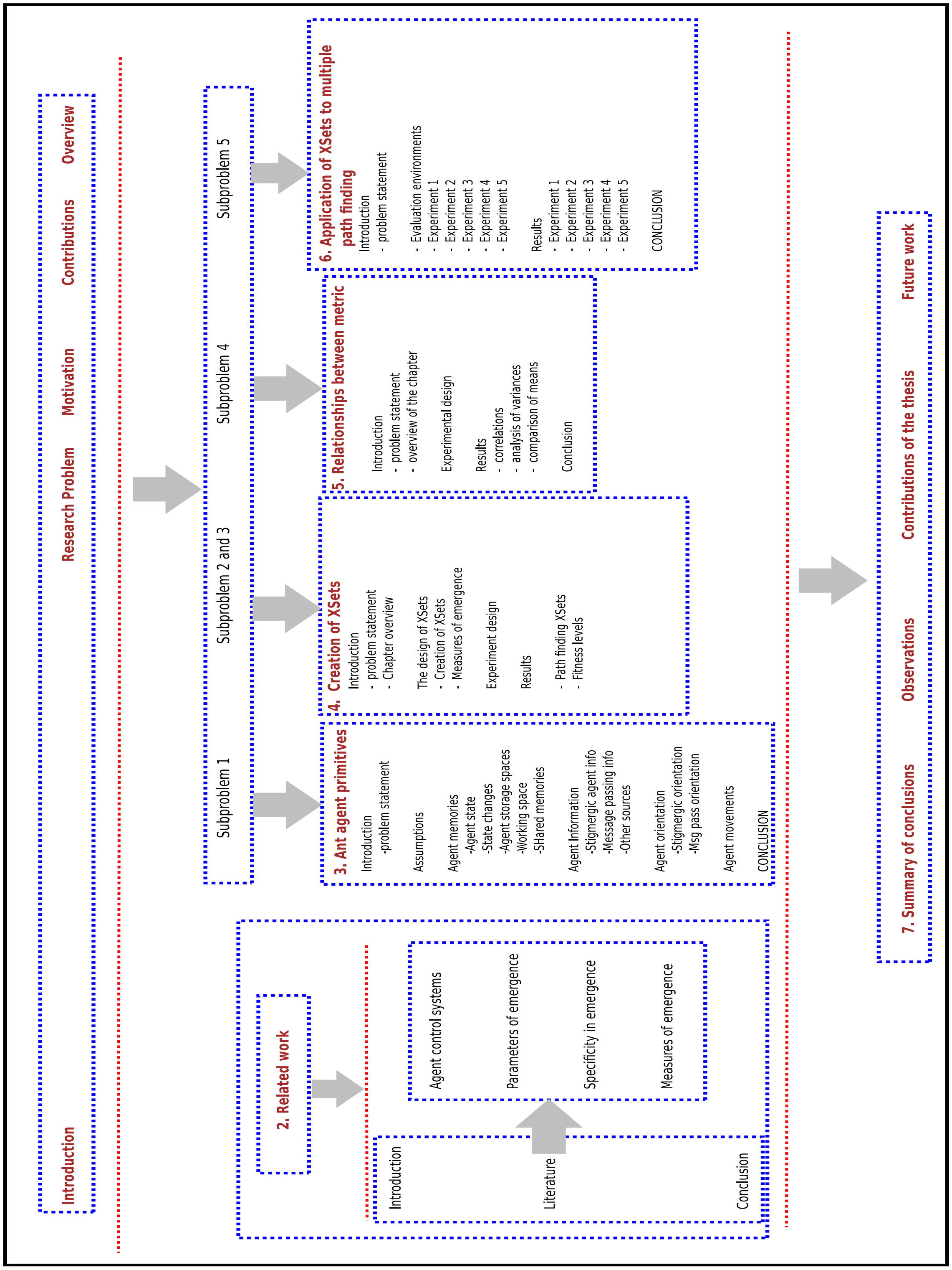


Figure 1.3: Chapters of the thesis and their relationship

- The first sub-problem of the thesis is addressed in chapter 3. Key in this chapter is the identification and detailed description of ant agent activities at individual levels. The semantics of the rules and conditions that are encoded in each routine further illustrate the functionality of the primitive behaviours that are identified.

- Chapter 4 develops a strategy for creating XSets and mechanisms for quantifying the extent to which emergent behaviour is manifest as a result of using particular XSets. The key outcome of this chapter are XSets which best characterize path finding behaviour.

- Chapter 5 validates the measures of emergence that are reported when best performer XSets are used, as a case study, to resolve the path finding problem. It precisely establishes any correlations between different measures of emergence, and compare the means and variances that arise thereof. Understanding these relationships is critical when we explain the phenomena that arise, as well as when we provide insights into which primitive behaviours are causal.

- Chapter 6 responds to the fifth sub-problem of the thesis, investigating the application of best performer XSets (in the path finding context) to different problem domains (particularly multiple targets location).

- Chapter 7 summarizes the thesis, presenting the overall observations we make and the contributions of the work, along with the proposed future directions of researches in this field.

Chapter 2

Related Work

2.1 Introduction

The literature review we present in this chapter explores, in an incremental manner, the concepts, agent control mechanisms, theories, and methods that help us in responding to the research problem of this thesis. First, we take a detailed survey of different agent control systems with the aim of finding common control rules and popular agent communication and interaction strategies. These reviews give useful insights into which strategies best characterize ant agent activities at individual levels - relating to the first sub-problem of the thesis.

The second sub-problem emphasizes on discovering the representation, in computational terms, of constructs that give rise to emergent behaviour. We particularly scrutinize literature which explores the common parameters of emergence that have been reported in different agent control systems. In this context, a parameter of emergence is a view, construct, data structure, or a system component which directly or indirectly influences the performances of simulated swarms, whether positively or negatively. These reviews give foresight towards understanding the representation of "collections of agent activities" that can allow deliberate engineering of desired emergent solutions - thus inspiring our proposed representation of XSets - relating to the second sub-problem of this thesis.

The third sub-problem pays attention to the mechanisms in which emergent behaviour is detected and quantified. We take a succinct survey of the literature which discusses strategies and approaches that have been proposed as emergence quantifiers in the past. That way, the choices we make of using specific measures of emergence are motivated.

The aim of the fourth sub-problem is to validate the measures of emergence that are recorded when particular XSets are used. It seeks to gather evidence for justifying the use of particular XSets as dictionaries for achieving a wide range of emergent behaviours. In the fifth sub-problem, we go on and assess possibilities of applying valid XSets to different task domains, testing the same XSets for properties and controls that can allow the creation of specific emergent behaviours. An abstract survey is taken which reports the common forms of specific emergent behaviours that have been simulated in the past - motivating the solutions sought in the fifth sub-problem.

2.1.1 Overview of the chapter

Figure 2.1 summarizes the sections of this chapter, as well as the topics that are discussed in each section. The details of the content covered in each section are further outlined as follow:

- Reviews which provide a detailed survey of different agent control systems with the goal of identifying key concepts, rules, theories, as well as the common agent interaction and communication strategies are

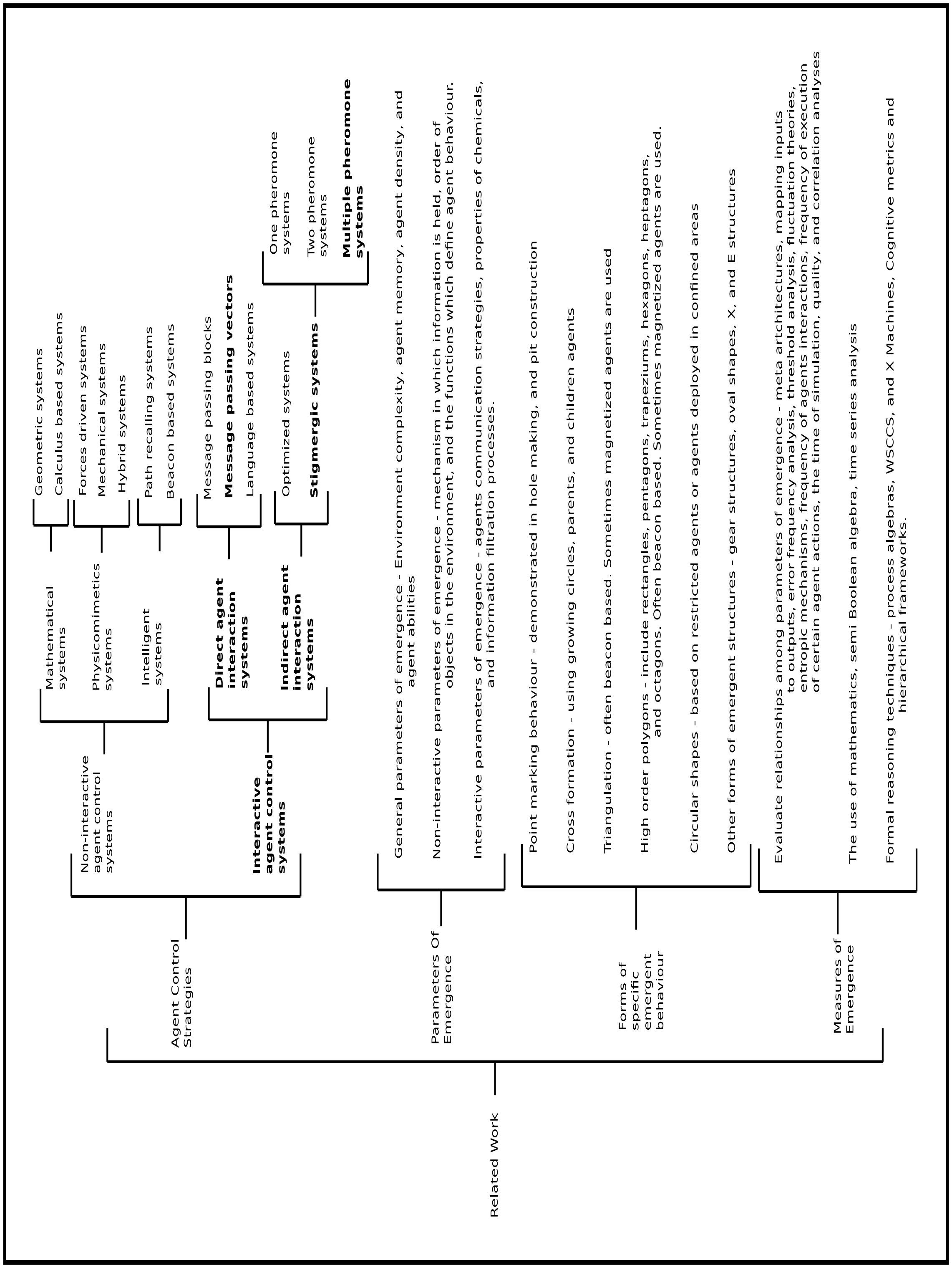


Figure 2.1: Overview of chapter 2

presented in section 2.2. In these reviews, we distinguish between two categories of agent control systems namely: non-interactive and interactive agent control systems.

- Section 2.3 takes a survey of the common parameters of emergence that have been reported in the literature, emphasizing on the factors which commonly influence emergent behaviour and how these factors are represented in computational terms. Two categories of parameters of emergence are identified, namely; system level and agent level parameters of emergence. In our context, system level parameters of emergence translate to meta information, while agent level parameters relate to the primitive behaviours. These two categories of parameters of emergence motivate the definition we proposed of XSets (meta information plus sets of primitive behaviours).

- In Section 2.4, we review literature which informs the choices we make regarding the measures of emergence which we propose when we address the third sub-problem of this thesis.

- Section 2.5 is developed with the assumption that generalized XSets exist which can allow different forms of specific emergent behaviour to occur. We then emphasize on scrutinizing related works in which specificity has been defined in different scenarios, the representation of specific tasks, and discuss how swarms of ant agents have been encoded with abilities to recognize these specific target goals.

- Section 2.6 concludes the chapter summarizing our findings, as well as highlighting the observations we make and the contributions the chapter makes to the thesis and board of knowledge.

2.2 Common agent control systems

The first sub-problem of this thesis investigates the key concepts, theories, and methods with which we can describe computational constructs which best characterize ant agent behaviours at individual levels. To review literature related to this sub-problem, we identify two categories of agent control systems and look at these categories separately. These categories are; non-interactive agent control systems and; interactive agent control systems.

2.2.1 Non-interactive agent control systems

Agent control systems in this category are commonly modelled on mathematical and physics laws of motion in which movement trajectories are defined using equations, matrices, or vectors. Agents in this category are often characterized by large memory capacities (Ngo et al., 2005) in which they keep information related to positions of objects on the environments (Mullen et al., 2009.; Sudd, 1960.; Montes De Oca et al., 2005). These agents may also keep record of vector information which indicates the preferred directions of motion (Wu et al., 2005). In other cases, agents are able to recall the properties of landmarks and beacons in the environment (Wehmer et al., 2006).

Three classes of non-interactive agent control systems are observed, namely: mathematically driven agent control systems, physics based agent control systems, and intelligent agent control systems. Each of these classes of agent control systems possesses unique control rules, unique agent communication policies, different agent orientation techniques, and different agent movement controls. We discuss each of these three classes of non-interactive agent control systems separately.

2.2.1.1 Mathematically driven agent control systems

This class of agent control systems heavily relies on rigorous mathematics for swarm coordination. The mathematics rules of motion thereof define agent orientation and movement policies (Ngo et al., 2005). However a question often arises regarding which field of mathematics is used for these purposes. We categorize mathematically driven agent control systems into two groups, one in which geometric controls are dominant, and another one where agents resolve navigation problems using calculus.

Geometry based agent control systems - neither require direct nor indirect agent interactions. Rather, each agent's positional preferences are based on Cartesian geometry (Trofimova et al., 1998). Agents in this category are designed with computational abilities to self-localize relative to the positions of specific objects in the environment. These agents can perform independent calculations out of which they can orientate, measure distances to the targets, and estimate the angles to turn relative to specific objects in the environment (Ngo et al., 2005). Agent motion is handled using velocity profiles and collision avoidance schemes that are often in-built in the system. Thus, the key parameter of emergence in this group are *agent memories*, agents' *individual abilities*, and the *mathematics laws of motion* that are define in the geometries thereof (Ngo et al., 2005).

A number of disadvantages are noted in this group of agent control systems which discourage us from recommending similar control mechanisms for ant agents. First, geometry based agents must possess extra computational abilities to generate local coordinate systems in which to self-localize. In addition, related agents require large memory capacities to record movement trajectories and environment features that would steer future orientation. Worse still, the same agents must be able to calculate velocities, distances, orientation angles, and define movement and collision avoidance profiles. These are handy characteristics for the simple and naive ant agents we propose.

Calculus based agent control systems - neither require direct nor indirect agent interactions either (Sarfati, 2001). Instead, each agent in this category can calculate its movement trajectories based on the relative positions of globally perceived objects of the environment. Here, the main activity of every agent is to self-localize. Jacobian matrices have been successfully proposed for this purpose (Harris, 2007). However these agents must possess abilities to solve equations and simplify mathematical functions into directional information. These requirements are too complex for the simple and naive ant-like devices we propose.

2.2.1.2 Systems driven by physics laws

Physics based control systems are characterized by the laws of motion they support. Three classes of agent control systems are observed in this field, namely; forces driven agent control systems, mechanical agent control systems, and hybrid agent control systems.

Forces driven agent control systems - support agents with abilities to respond to in-built virtual forces for sensing the proximity of one another (Balch and Arkin, 1999), as in the case of flocking boids (Reynolds, 1987). Agents in this category can attract and repel each other depending on their distances apart (Azzag et al., 2007.; Beckers et al., 1989). That way, movement speed and orientation is regulated depending on the push and pull effects of the virtual forces that are exerted between neighbours at the time (Bayazit et al., 2002). These virtual forces define agents' positional and directional preferences relative to other agents in the swarm (Lua et al., 2005.; Parrish et al., 2002).

Typical examples of forces driven systems have been presented in the works of Spears et al. (2004a, 2005) and Spears et al. (2004b). In these works, agents can successfully self-organize into mobile hexagonal lattices, moving towards specific light-bulb targets. The key drivers, and main parameters of emergence in these works are agents' sensory skills. Precisely, the potential field of energy which builds around each agent is the key ingredient for subsequent agent actions. However these agents lack autonomy since their behaviour heavily relies on the density of attractive and repulsive agents around (Cao et al., 1997). Worse still, these agents must have mechanical sensor devices that are physically mount.

Mechanical agent control systems - propel agent movements using electric motors that are physically mounted on each agent. Agent orientation and movement trajectories are pre-defined in the motion routines of the system with neither direct nor indirect interactions. The electric motors that are mounted on each agent are often built with enough energy to run for the duration of the simulation (Paulson, 2008). However mechanical systems prevalently achieve pre-programmed outcomes rather than emergent behaviour. In addition, agents in this category must be deployed in specific agent densities, where each agent has a well defined schedule of tasks to accomplish (Paulson, 2008.; Regan et at., 2005). These properties are not realistic for the ant agents we study.

Hybrid agent control systems - combine the features of forces driven and those of mechanically motivated agent control systems. A typical hybrid agent control system is proposed in the work of Pelechano et al. (2007), in which both virtual forces and geometry based control rules are put together to trigger agent navigation. In these cases, displacement equations are prescribed when agents' sensory abilities detect obstacles ahead. However the integration of different agent skills, as described in these models, does not take away the complexities that are associated with using each model separately. Rather, this would add more special cases to agent motion requirements (Pelechano et al., 2007).

2.2.1.3 Intelligent agent control systems

The last group of non-interactive agent control systems is referred to as intelligent agent control systems. These are characterized by agents with sufficient memories to recall previous events in simulation, and use that information to infer appropriate actions. Two groups of intelligent agent control systems are dominant in the literature. These two groups are distinguished from one another by the kind of information the agents keep in memory. One group of agents has abilities to recall the entire paths or maps they followed from the start of journey to the destination (Cordon et al., 2002.; Mullen et al., 2009). The other group relies on landmarks and beacons that are held on the environment as holders of information with which agent orientation is achieved (Wehner et al., 2006).

Path recalling agents - have selective abilities to choose the control mechanism to employ at a given time in simulation. At one point, agents may recall landmarks and beacons around, and use these to steer orientation (Sudd, 1960). In other cases, agents remember what to do next from the behaviour of neighbouring agents. However, when isolated, the same agents may even recall and follow the direction and angle of the sun (Koichi and Mari, 1996). A similar model has been demonstrated in the work of Erbas et al. (2013) where agents successfully use imitation of observed behaviours of neighbours to navigate environments without any internal state access or sharing of experiences.

A key parameter of emergence in path recalling agents is the mechanism is which path records are kept in the system. Commonly, sequence generation techniques (Cordon et al., 2002), and tabu search strategies (Ghaiebi and Solimanpur, 2007) are employed. There are models in this category whose agents can recall other agents' identities (Sheeham and Tibbetts, 2008), thereby steering one-on-one cooperation or aggression. As such, agent activities at individual levels, and the degree of success at swarm levels are dependent on the quality of information that is held in agent memories (Viana et al., 2007.; Xu et al., 2008). However the demand for large memory capacities in the agents thereof, discourages us from recommending these controls for the ant agents we propose.

Landmarks and beacons based agent control systems - make use of agents that have unlimited memory capacities in which to keep important information relating to the properties of landmarks and beacons on the environment (Wehner et al., 2006). The recalled landmarks provide direction vectors and orientation information which steers agents towards desired directions. Usually, these landmarks are used to estimate the Euclidean distances between the agent's current location and the target sought (Wu et al., 2005). Desert ant agents in particular, have been shown to exhibit the characteristics of agents in this category (Roumeliotis et al., 2000), thus achieving emergent behaviour without neither direct nor indirect agent interactions. Stigmergic interactions are completely impossible in these swarms because all the levels of pheromones would dissipate before they are useful to the swarms due to the harsh conditions in desert environments.

In other cases, landmarks based agent control systems possess selective abilities to decide on how agents can orientate. Isolated agents may re-align with the rest of the swarm by using sensory hints (Cavalcanti et al. 2006a.; Cavalcanti et al., 2007). However once they are back in formation, they can follow specific vectors based on the information held in the landmarks and other agents around (Jackson et al., 2004). Generally, the knowledge held in an intelligent agent's memory is filtered in each step until the agent acquires deterministic directional information (Dhariwal et al., 2004).

2.2.2 Interactive agent control systems

These control systems are predominantly nature inspired and most of them are modelled on the behaviours of living organisms such as cells (Xi et al., 2005), birds (Reynolds, 1999), DNA sequences (Reif, 2002), bees (Reynolds, 1987), or ants (Chibaya and Bangay, 2007). Agents in this category depend on one another in order to complete individual level agenda. All interactions, whether direct or indirect, are local.

Two classes of interactive agent control systems arise in the literature, namely; those in which agents interactions are directly one-on-one, and those in which interactions are indirectly mediated. We discuss these two classes of interactive systems separately.

2.2.2.1 One-on-one agent interaction systems

This class is commonly modelled on the behaviours of agents with abilities to exchange information one-on-one. The information that is often shared is in the form of memory blocks which hold specific data relating to directions (Nasipuri and Li, 2002), paths histories (Rajbhupinder et al., 2010), or positions of specific objects (Montes De Oca et al., 2005). In some cases, this information relates to explicit calls that are made in a specific agent communication language (Nagpal et al., 2002.; Nagpal et al., 2003.; Abelson et al., 2000). Some important considerations in all direct agent interaction systems pertain to the requirement to know what information is transmitted between agents, how this information is transmitted, and when it is appropriate for agents to explicitly share information (Haasdijk et al., 2013.; Couzin et al., 2002). Consequently three types of direct agent interaction systems arise.

Interaction systems in which path histories are explicitly shared - are prevalent. In these, message blocks which hold path histories in the form of stacks are shared (Trianni and Dorigo, 2005). The stacks thereof record the coordinates of the paths an agent followed in the past (Rajbhupinder et al., 2010), or information relating to the best tours an agent made to that far (Hara et al., 2005). Other historic records reports the entire environment maps an agent followed, including pointers to promising locations in the environment (Montes De Oca et al., 2005).

Usually, a learning framework arises (Lien et al., 2005), in which agents learn the experiences of their neighbours by explicitly referencing neighbours' path histories when necessary. These agents would create their own roadmaps based on the experiences of their neighbours (Rodriguez et al., 2007). Neighbouring agents' path histories can be accessed both directly and indirectly (Bayazit et al., 2002.; Bayazit et al., 2005).

Although the notion of information sharing is nature inspired (Nouyan and Dorigo, 2007), there are three obvious disadvantages that arise. First, agents in this category must possess large memory capacities to hold the message blocks. In addition, agents' memory structures must be compatible with the message blocks that are shared. Thus, all agents are similar (Caicedo et al., 2001). Worse still, important information that is held in the memories of less successful agents may be lost when path histories of relatively more successful agents are inherited.

Interactive systems in which agents can share geometric vectors - are more inspiring. Agents in this category do not require relatively excessive memory capacities since they would only keep record of specific vector components for orientation and navigation purposes (Nasipuri and Li, 2002). The vectors thereof usually interpret the levels of pheromone on the environment (Payton et al., 2001). In other cases, they are geometric pointers to specific objects or directions in the environment (Nasipuri and Li, 2002), with x, y, and z components.

Systems in which a communication language - is used have been reported as well. Agents in this class often have a common communication language with which to share information one-on-one. Most agent communication languages are developed with full syntax, vocabulary, and semantics that are only understood by these agents (Nagpal et al., 2002).

Popular in this category are agent communication languages that are based on the growing point and origami shape theories (Nagpal et al., 2003). In particular, the work of Butera (2002) is more inspiring, in which a growing point language has been used to implicitly enhance pheromone dissipation in swarms of ant-like devices.

In other agent communication languages, high level description of functions and relationships among agents are required upfront (Sussman, 1999). Such agent communication languages often incorporate processes and properties to coordinate the behaviour of individual agents all the way (Belani et al., 2002). In most cases, sets of pre-programmed coordination laws and primitive behaviours are incorporated in the system upfront (Stefano and Santoro,

2001) together with the vocabulary for the agents (Kraus and Lehmann, 1995).

Other agent languages support call protocols that are explicitly developed into agent verbs such as "*move*", "*respond*", "*avoid*", "*recruit*", or "*hello*" (Nagpal and Coore, 1998.; Cao et al., 1997). However these calls are often broadcast to the entire swarm, a feature which compromises agent privacy and system security in general.

An agent communication language of geometric primitives and homeostasis maintenance has been successfully used as an amorphous medium language in the work of Beal (2005a, 2005b). In this work, Beal (2005a, 2005b) used this language to describe agent behaviour in terms of the spatial regions of the amorphous media (Abelson et al., 2000), where neighbouring agents are only allowed to communicate by means of a shared memory region (Nagpal et al., 2006).

Investigations aimed at identifying the primitive behaviours of ant agents with abilities to communicate using specific agent communication languages are outside the scope of this thesis for a number of reasons. First, agent languages in the literature are, at the moment, very limited in vocabulary (Nagpal et al., 2002). As such, only a limited domain of emergent behaviours have been tested using related swarms. Secondly, although researches that use sentence messages are in progress (Cranefield et al., 2000), the results presented so far lack in that the roles of receiver agents are made consequences of the desires of sender agents (Dastani et al., 2003). In other words, the independence of the receiver agents is grossly compromised.

2.2.2.2 Indirect agent interaction systems

Models in which agents interactions are indirectly mediated are predominantly chemically inspired. Virtual chemicals are usually placed on the environment, thus creating shared memories for the swarms. These virtual chemicals are placed on the environments in two ways. First, there are cases where the objects of the environment are the origins of these chemicals (Naeem et al., 2007). In other cases, the agents place these chemicals on the environment (Dorigo, 1992.; Dorigo et al., 1999). This thesis refers to interaction systems in the former group as optimized, and those in the latter group as stigmergic.

Optimized interaction systems - support chemical markers that are placed on the environment by the objects in the environment other than the agents (Naeem et al., 2007). For example, chemical plume gradients have been created at specific sources in the environment which guided agents to those sources (Dhariwal et al., 2004.; Naeem et al., 2007). What stands out in optimized interaction systems is the requirement for agents to self-localize relative to the chemical sources (Nagpal, 1999.; Nagpal et al., 2003). In other words, local coordinate systems arise in which agents can determine the direction to follow relative to the quality of the chemicals around the agent (Merkle et al., 2006).

In most cases, chemicals in optimized systems define uni-directional paths (Jackson et al., 2004). As a result, elitist mechanisms are often required which provide extra selective and adaptive characteristics to agents when bi-directional paths are required (Koichi and Mari, 1996.; Negulescu and Barbat, 2004). A common form of elitism involves agents that can conveniently switch between different interaction strategies when it becomes necessary (Montes De Oca et al., 2005). At one point, agents may use sensory cues together with chemical gradients (Wehner et al., 2006). In other cases, the same agents may use some form of limited vision to augment chemical tracing (Colin, 2006). However the bulk of optimized systems supplement chemical tracing with extra agent memories to facilitate agent recall (Ravary et al., 2007.; Healey and Pratt, 2008).

Further studies related to investigations of the primitive behaviours that characterize optimized ant agents are outside the scope of this thesis because of that requirement to consider elitism in individual ant agents (Di Caro et al., 2004.; Yang and Zhuang, 2010). Precisely, elitism takes away agent autonomy as seen in the works of Gottlieb et al. (2003), Solnon and Fenet (2006), Svenson and Sidenbladh (2003), and Balch and Arkin (1999).

Stigmergic interaction systems - make use of agents with abilities to excrete specific levels of pheromones (Dorigo, 1992.; Dorigo et al., 1999.; Dorigo et al., 1996). These agents make use of a non symbolic form of communication which is mediated via the environment (Di Caro et al., 2004.; Negulescu et al., 2006). The term stigmergy was coined in 1959 by Grassé (Theraulaz et al., 1998.; Bonabeau et al., 1999.; Socha, 2008). It is formed from the Greek words *"stigma"*, which means "signs", and *"ergon"* which means "actions". The term therefore captures the notion that agents' individual activities would leave "signs" on the environment, signs which would determine agents' subsequent "actions" (Parunak, 2005).

Literature further distinguishes between two forms of stigmergy (White, 1997.; Shell, 2003). The first form is called sematectonic stigmergy (Parunak, 2005), which involves changing the physical characteristics of the environment. Examples of sematectonic stigmergy are demonstrated in the hole making problem (Ghaiebi and Solimanpur, 2007), pit construction problem (Burgess, 2009), and nest building problem (Downing and Jeanne, 1988.; Andrew, 1978.; Aleksiev et al., 2007.; Franks et al., 1992.; Jeanne, 1996).

We are more interested in sign-based stigmergy in which pheromone signs are marked on the environment. Although these pheromone signs may not have direct relevancy to the tasks being undertaken by the agents at the time, they indirectly influence subsequent agent actions and behaviours, those behaviour which may be task related.

In sign-based stigmergic interactions, agent mobility is probabilistic (Chibaya and Bangay, 2007). Agents' path selection decisions are biased by the levels of pheromone that are held on the agent's local environment (Montgomery et al., 2007). Sign-based stigmergy is further classified into three categories namely: single pheromone systems, two pheromone systems, and multiple pheromone systems.

Single pheromone systems - support agents with abilities to excrete and perceive one and only one form of pheromone. All agents in this category are sensitive to this single level of pheromone regardless of the task they have at hand. The sources of these single level of pheromone are the agents of the swarm themselves (Dorigo et al., 1999). However, there are cases where search targets in the environment have been designed with abilities to excrete this single level of pheromone as well (Cavalcanti et al., 2006a). Nevertheless that would characterize optimized agent systems (Schneider et al., 2006). In this review, we focus on stigmergic systems in which the sources of pheromone are the agents of the swarm.

Among the most popular examples of single pheromone systems is the double bridge scenario (Dorigo, 1992) that is illustrated in Figure 2.2, and adopted from the work of Cordon et al. (2002). In this experiment, ant agents are able to excrete and update the same level of pheromone regardless of the direction in which they are travelling across the bridge. Food sources and the nest are situated on different ends of the two way bridge. The task of each ant agent is to travel across the bridge in search of food sources, and upon finding them, pick up the resources, and return back to the starting point (Dorigo, 1992.; Cordon et al., 2002.; Schoonderwoerd et al., 1996). These trips are repeated for the entire duration of the simulation.

In most cases, the single pheromone trails that arise are uni-directional. Gradients often arise in which agents can only move from low to high chemical

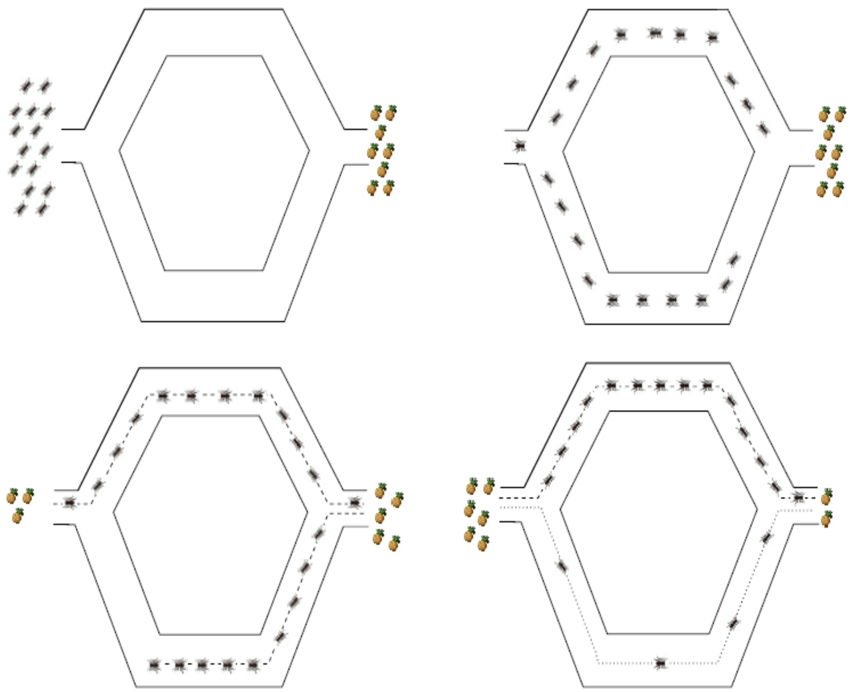


Figure 2.2: The double bridge experiment setup

concentrations. As a result elitist strategies are required when bi-directional agent movements are sought (Koichi and Mari, 1996). That requirement for elitism make single pheromone systems not suitable for the ant agents we study.

Two pheromone agent control systems - are primarily designed to minimize the need for elitism in single pheromone agent control systems. Agents in this category are sensitive to two different levels of pheromone that can co-exist on the same local environments without interfering with one another (Panait and Luke, 2004b).

Commonly, both levels of pheromone are placed on the environment by the agents of the swarms, where one level is excreted when agents travel from the starting point in search of the resources, and another level is deposited when agents travel in return trips (Chibaya and Bangay, 2007.; Solimanpur et al., 2005). However there are cases where one or both levels of pheromones originate from other objects of the environment (Alcala et al., 2001.; Nakamichi and Arita, 2004). This thesis only considers stigmergic systems in which the levels of pheromones originate form the ant agents of the swarms since our hope is to, one day, apply the same constructs to different task domains that are coordinated in differently configured environments.

Multiple pheromone interaction systems - remedy most flaws that are noted in optimized, single pheromone, and two pheromone interaction systems. These models are usually built on the notion that every extra level of pheromone that is added in the system reduces some form of elitism. As a result they are more robust and fault tolerant, flexible, and adaptable to supporting autonomous agents (Cavalcanti et al., 2006a).

A number of examples of multiple pheromone systems are available in the literature. Most of these examples simulate medical scenarios. The work of Cavalcanti et al. (2006a) is a typical example in which mobile cancer cells are simulated as operating in human blood vessel-like environments. In their model, cancer cells can excrete the first level of pheromone called cancer pheromone, which is attractive to cancer attacking agents. Cancer free cells excrete the second level of pheromone called obstacle pheromone, which is repulsive to cancer attacking agents. This mechanism ensures that the cancer attacking agents do not waste time examining cancer free cells for the symptoms of cancer.

Upon finding a cancer cell, an agent excretes the third level of pheromone called alarm pheromone, which is also attractive to cancer attacking agents that are still in the seek mode. As a result, helper agents can be recruited around the cancer cell and help to destroy the tumor. This model has inspired the development of many other routines for nanite-like agent coordination in similarly inside-the-body environments (Cavalcanti and Freitas, 2005.; Cavalcanti et al., 2005b).

Closely related to the cancer treatment model (Cavalcanti et al., 2006a), is the wound detection system (Schneider et al., 2006). In this model, platelet-like agents are simulated as moving inside vessel-like environments. These platelet-like agents are tasked to identify and cure wounds that are inside the blood vessels. The wounds themselves excrete the first level of pheromone called wound pheromone which is attractive to the platelet-like agents. The platelet-like agents would only stick around a location if the concentration of wound pheromone on that location is high enough to indicate the presence of a wound. Agents around a wound can excrete the second level of pheromone called alarm pheromone in order to attract other platelet-like agents towards the wound. Clean surfaces in the vessels are similarly obstacle-like and can excrete a third level of pheromone which is repulsive to platelet-like agents. This way, platelet-like agents' medical examination time is not wasted on clean surfaces.

A major flaw in most multiple pheromone systems is that most levels of pheromone originate from other objects of the environment other than the agents, which is elitist. In addition, related agents are required to possess special skills to be able to selectively perceive (Engle and Whalen, 2003), and distinguish between different levels of pheromone (Nakamichi and Arita, 2004, 2005.; Nowé et al., 2004).

The stigmergic model we propose in this thesis combines the advantages of two pheromone interaction systems with those of multiple pheromone systems. It investigates ant agent activities in scenarios where one level of pheromone is excreted when ant agents travel in search of the targets, and another level is excreted when agents forage back to the nest-like starting point. The environments we propose support multiple levels of pheromones, including chemical markers which indicate the positions of key objects and targets. This model has a number of advantages over other agent control systems in the literature. First, our ant agents are completely free from the need to have large memory capacities. The important information they require throughout the simulation is held on the environment in the form of shared memories. As a result, errors at agent levels do not affect completion of tasks at swarm levels, resulting in relatively robust and fault tolerant solutions. In addition, elitism in agent activities at individual levels is not required. Thus, defining autonomous ant agent activities.

The next section reviews literature which reports the common parameters of emergence, concentrating on how such parameters are represented in computational term (thus motivating the design we follow when we represent the XSets we propose - relating to the second sub-problem of the thesis).

2.3 Parameters of emergence

This section reviews literature which takes a survey of the common parameters of emergence that have been discussed in different agent control systems. We emphasize on the description and representation of these parameters of emergence, thus inspiring the design we follow when we describe and represent the XSets we propose - relating to the second sub-problem of this thesis.

Parameters of emergence were defined in section 2.1 as factors which influence swarm performance towards emergent behaviour. We learn of a number of parameters of emergence that are used in different agent control systems. These parameters of emergence can be classified into two categories namely;

system level parameters and agent level parameters of emergence. We discuss each category separately.

2.3.1 System level parameters of emergence

These usually influence the performance of the swarm at large. Our work refers to system level parameters of emergence as meta information. These parameters spell out and initialize the key components of the simulation system. Often, they are provided by the user at run-time. The following are common system level parameters of emergence that are inferred in the literature:

1. **Environment** :- simulation environments are the most popular system level parameters of emergence. Environments, and the way in which environments are described, influence emergence since they are designed to hold key system information (Negulescu et al., 2006.; Haasdijk et al., 2013). They often record the coordinates of landmarks and beacons in models where these components are supported (Valckenaers et al., 2001). In addition, environments provide platforms on which agents reside (Haasdijk et al., 2013). In stigmergic models, the levels of pheromones are stored on the environments, thus creating environment mediated shared memories for the swarms (Panait and Luke, 2004a.; Babaoglu et al., 2006). More importantly, environments usually handle the high level descriptions of desired emergent behaviour (Seevinck and Edmonds, 2008.; Mason, 2002), including the fitness functions with which agents adapt the problem domain over time (Bredeche et al., 2012). Practically, environments are tuples with many fields of information for the swarm. Given these numerous importance of environments, this thesis considers our simulation environments as key independent variables in the experiments we conduct.

2. **Agent density**:- Beckers et al. (1989) and Van den Bergh and Engelbrecht (2001) confirm the relationships that are observed between agent density and the quality of emergent behaviour thereof, as well as the relationships that are observed between agent density and the speed with which emergent behaviour occurs. However, although large agent densities are desirable to direct agent interaction systems (Weyns et al., 2004) where one-on-one interactions are more effective when agents are congested, these may be depletive to indirect agent interaction systems where dissipative pheromone levels may saturate the environments early in simulation time (Chibaya and Bangay, 2007). In this thesis, we try to establish appropriate agent densities in different scenarios, and express these agent densities as functions of the sizes of the environments, as well as functions of the distances between the targets and starting point.

3. **Agent memories** :- although most of the simulated system resources are held on the environments, agents usually require some basic memory to record specific meta information at individual levels (Dorigo and Blum, 2005). Non-interactive agents in particular, require memories to record information relating to landmarks and beacons (Wu et al., 2005). Other agents in this category require memories in which to hold navigation equations and velocity control functions (Ngo et al., 2005), or to store the roadmaps the agents would use (Rodriguez et al., 2007). Interactive agents on the other hand, require limited memories in which to hold internal state information.

4. **Mechanism in which information is stored** :- agent control systems support different ways of handling system information. There are agent control systems in which information is recorded in separate routing tables (Brown et al., 2005) or on virtual noticeboards (Schoonderwoerd et al., 1996). However that would, in a way, jeop-

ardize agent autonomy (Nagpal and Coore, 1998.; Kondacs, 2003) and system security (King et al., 2005).

5. **The order of objects** :- although the environment is generally the main holder of the key objects of the system, the order in which these objects are arranged on the environment influences the quality of the output of the swarms (Gulyas et al., 2006), as well as the speed with which emergent behaviour occurs. Models have been reported in which specific placement scores have been allocated to important objects of the environment in order to improve agent awareness (Don and Amos, 2007). Our work sets the starting point at fixed positions in the environment, while resources are randomly distributed.

The reviews we presented in this section hint us on the key variables of the simulation system we require. The representation and composition of the environments stand out as the key parameter of emergence to consider. In addition, agent memories, agent density, as well as the order in which objects are arranged on the environment are also factors of interest in our work. The next section discusses agent level parameters of emergence.

2.3.2 Agent level parameters of emergence

Agent level parameters of emergence are more inspiring since they describe what agents do at individual levels which influences emergent behaviour. These give an insight into potential primitive behaviours for ant agent controls. The following are common parameters of emergence in this category:

1. **Agent abilities** :- this refers to agent activities that are common to all members of the swarm such as moving, picking up a resource, dropping resources, orientation, interaction, flipping between states, or sharing

information. Similar unit level agent actions have been explicitly mentioned in the work of Haasdijk et al. (2013) and Winfield et al. (2013), in which the agent's internal cues are the key parameters of emergence. Our desires to explicitly state and describe ant agent actions at unit levels is what drives research in this thesis.

2. **Laws of motion** :- while non-interactive agent control systems rely mainly on sophisticated mathematics and physics laws of motion for defining agent trajectories (Johnson and Rossi, 2006), interactive agent control systems are relatively flexible in this respect. Interactive agents rely on local information, often held in neighbouring agents or on the environments. Non-interactive agent control systems often combine various views and axioms in order to solve navigation tasks (Wei-min et al., 2004). For example power-laws (Adamic et al., 2001) have been used to empower particular groups of agents that would assume leadership roles. The ant agents we propose are all interactive where stigmergic ant agents follow pheromone gradients and message passing ant agents follow geometric vectors as laws of motion.

3. **Information update rules** :- apart from being held in different ways, system information is updated regularly. Different agent control models update meta information differently depending on the tasks at hand. Algorithm 2.1 (derived from the works of Ke et al. (2008), Montes De Oca et al. (2004), and Viana et al. (2007)) shows the common components of foraging swarms. In this algorithm, we see the requirement to initialize information (Panait and Luke, 2004c). Orientation is based on the information available which must be updated in each agent step (Dorigo et al., 1999). In stigmergic systems, global update rules may be added which handle pheromone dissipation processes (Chirico, 2004.; Schoonderwoerd et al., 1996). We design our XSets with this sequential representation of agent activities in mind (see algorithm 2.1).

Algorithm 2.1 Components of foraging agent systems

```
Initialization //environment components
For-each agent in each step
    orientate //check direction to follow
    move update_info //drop pheromone or update vectors
    review_state //is the internal state still valid?
End-for
```

The reviews we presented in these last two sections augment the concepts, methods, and theories which we can derive from the previous survey on agent control systems. We highlight some of the key agent activities that are inferred as influential to emergent behaviour. Generally, agents possess abilities to orientate, move, interact, and update information. The next section discusses the mechanisms that are commonly used for quantifying the emergent behaviours that arise from using agents with similar abilities.

2.4 Quantification of emergence

The third sub-problem investigates the mechanisms for detecting and quantifying the extent to which emergent behaviour is manifest as a result of using particular XSets. This section discusses the common strategies that have been proposed for this purpose. In our view, the ability to quantify emergent behaviour is critical for understanding the validity of XSets and the extent to which XSets work.

Popular quantifiers study the relationships that exist among events in simulation (Gore and Reynolds, 2008). These relationships are described in various ways depending on the scenarios in which they are measured. The following are examples of works in which relationships among events have been investigated:

- **Amount of change** :- The work of Schaefer et al. (2002) measure degrees of emergence by mapping inputs to outputs. The gap between inputs and outputs is interpreted as the amount of emergence thereof. This however has the disadvantage of requiring that we understand the full processes through which inputs are processes until outputs are reported. Hinchey et al. (2005) quantify amount of change in terms of error frequencies. Thus, the frequency with which, and sequence in which, certain time-based errors occur is tracked. A variant of error frequency analysis is threshold analysis in which parameters which violate certain predefined system conditions over time are detected (Rouff et al., 2004). Similar fluctuation theorems compare the probabilities of observing certain time based entropy over time (Hamann et al., 2011.; Chan, 2011). Our work interprets amount of change in terms of system throughput, and re-phrases this measure of emergence to *average delivery rate*.

- **Frequencies of agents interaction** :- There are models in which emergence has been quantified using the frequency of agent interactions (Noble and Letsky, 2002). In these models, the number of agent interactions that are completed in a given time frame is recorded, and the change in interaction frequencies is observed. The gap between the frequency of interactions over time follow a pattern which indicates the degree of emergence thereof. Similarly, agent actions can be associated with certain priority probabilities which evaluate the chances of priority activities being enabled (Wang and Zhu, 2007). Thus, specific agent activities are marked as priority actions, and the frequency with which these activities are completed is tracked. We re-phrase this measure of emergence as *average end-to-end delays*.

- **Convergence time** :- The time it takes a system to converge is an important measure of emergence (Wang and Zhu, 2007). Time in simulation is often measured in iterations. The work of Hovda (2008) refers to time in simulation as the "amount of simulation". Minati (2002) calls it ergodicity. In particular, ergodicity associates the average behaviour of a set of agents at a particular time to the average behaviour of an individual agent. Often, ergodicity changes when emergent behaviour arises, indicating deviation of the system from its original state (Chan, 2011). In this work, convergence time is re-phrased to *speed of emergence*.

- **Quality of products** :- The quality of the products, their timeliness, as well as the efficiency with which these products are produced is an important measure of emergence as well (Noble and Letsky, 2002). In particular, quality assesses agents' adherence to schedules, as well as their levels of engagement with the task at hand. That degree of engagement indicates the amount of emergence in the system. Our work translates this measure of emergence to *quality of emergence*.

- **Correlation analysis** : - Most inspiring are the correlation analyses that are performed on different system events and metrics (Wang and Zhu, 2007) as quantifiers of emergent behaviour. These correlation measures reveal the relationships among events, and how influential each event is to the system. Correlation analysis gives meanings to individual parameters of the system. As a result, centrality measurements can be deduced from the data sets (Valverde et al., 2006).

- **Mathematical analysis** : - Rigorous mathematical axioms have also been proposed as means for quantifying emergence (Sumpter et al., 2001). Such axioms often rely on statistical theories to identify deterministic dependencies among system elements (Parunak and VanderBok, 1997). The same dependencies can be derived using discrete event simulations (Bonabeau et al., 1999), semi-Boolean algebra (Haglich et al., 2010), or time series analysis techniques (Shalizi, 2001). Mathematical techniques are good pattern discovery techniques, and changing point detection algorithms have been developed for this purpose (Grossman et al., 2009).

- **Formal reasoning approaches :-** In these, process algebras are popular (Sumpter et al., 2001). They measure the ability of a system to support emergent behaviour, as well as the extent to which the system can support agent interactions (Fulbright and Stephens, 2003). They allow formal reasoning about which components of a system contribute to the overall emergent behaviour (Dixon et al., 2011), and prove that there exist relationships between these components. An effective process algebra must be able to predict the emergent behaviour of the swarm based on its components (Wang and Zhu, 2007). This has been successfully demonstrated in the work of Tofts (1991).

- **Cognitive approaches** :- These quantification strategies evaluate the effectiveness of collaboration in swarms by quantifying the extent to which agent collaboration improves team effectiveness, and provide reasons for the improvements (Noble and Letsky, 2002). In other words, cognitive metrics measure the extent to which a swarm understands what it needs to do. In this context, cognitive metrics are related to hierarchical frameworks that are based on computational mechanics in emergence (Crutchfield, 1994). These metrics take into account deterministic and stochastic complexity factors of the system in order to determine the effectiveness of the system. Most successful measures of emergence in this category integrate different methods together. For example, integrating the memory and transition function aspects of X-Machines with the priority and probability aspects of process algebras produces a specification method that allows all the necessary aspects for specifying emergent behaviour possible (Wang and Zhu, 2007).

- **Entropy measures** - system fluctuation theories have been considered as measures of emergence (Hamann et al., 2011), mainly investigating entropy measures over time. These entropic mechanisms are often based on Shannon's measurement theories (Bavaud et al., 2005.; Martin, 2006.; Schneider, 2007). A review of Shannon's information theory as a measure of emergence is proposed and detailed in the work of Fernández et al. (2013). Precisely, Shannon proposed a function to measure the value of information a process produces by considering the choices that are involved (Fernández et al., 2003). An analogy to Shannon's definition of information value can be derived for ant agent swarms. For example, suppose a stigmergic ant agent has the following possible destinations in a movement task : $L_1, L_2, ..., L_n$. This collection of possible destinations form a set of the possible choices where each destination has an associated probability of being chosen $(p_1, p_2, ..., p_n)$. Shannon determines how uncertain an agent in this situation is of the outcome using the function $(\sum p_i log_2 p_i)$. Our work takes advantage of this analogy and determines information values around ant agents in simulation. In this case, the average information value we observe indicates the extent to which emergent behaviour occurred. In determining this metric, we bear in mind the flaws that arise when not all agents are part of the emergent behaviour being evaluated at the time.

This work considers correlation analysis in Chapters 5. However mathematical analysis, formal reasoning, and cognitive approaches are outside the scope of this work because they involve rigorous mathematical axioms which rely on statistical theories, semi-Boolean algebra, or time series analysis. Although these techniques are good pattern discovery techniques, they would compromise our intention to develop simple and naive ant agent XSets.

2.5 Multiple targets location

Our work gathers evidence for justifying the use of particular XSets as dictionaries for creating different forms of emergent behaviours by assessing particular measures of emergence. Once justified, which forms of specific emergent behaviours can we create? How do we describe specificity? How do agents perceive specificity? Generally, there is no conclusive stand as to which specific emergent behaviours are better testbeds than others. We take an abstract survey of the various experiments that have been administered for assessing agent abilities to generate specific emergent behaviours. These reviews will inspire the choices we make of simulating the creation of specific emergent formations.

Nature inspired solutions are dominant in the literature. They are also more successful in this respect owing to relatively better robustness, adaptability, and fault tolerance (Eyiyurekli et al., 2013.; Polack et al., 2005). For example, a number of specific emergent behaviours have been achieved using cell propagation theories (Nagpal, 2006.; Bai et al., 2008), cellular automata (Geer et al., 2003.; Green, 1994.; Sanders and Smith, 2009), cell growth and morphogenesis theories in developmental biology (Nagpal et al., 2002), as well as using scaffolding DNA origami theories (Rothemund, 2006). This work extends the list and investigates the use of ant agents for generating specific emergent behaviour.

Simulation systems in which agents create emergent geometric structures are common. This is often because success or failure is easily measured in these setups (D'Hondt, 2000.; Kaewkamnerdpong et al., 2007). We particularly identify how specificity has been described in most of the geometric structures that have been simulated.

The smallest unit of drawings is a point. Systems have been reported in which point plotting is implicitly achieved. For example, the hole-making problem is practically a point marking solution (Ghaiebi and Solimanpur, 2007), as well as the pit construction problem (Burgess, 2009), and target location in our work (Chibaya and Bangay, 2007).

Works have also been presented in which swarms of agents successfully create crosses (Nagpal et al., 2002). These cross structures have been generated using the growing circles theory in which the agents are able to reproduce children agents in the process. However the major drawback of using growing circles theory is the requirement to use power laws which set parent agents as more powerful than children agents. Parent agents have authority to hold leadership roles, and power to act as cardinal references to children agents. In this case, specificity is defined in the parent agents' architectures (Nagpal et al., 2002).

On the other hand, the triangulation problem has been tackled in similar ways (Rothemund, 2006.; Werfel, 2002.; Kaewkamnerdpong et al., 2007). Triangulation solutions are important for solving graphics and surface subdivision problems (Hardy, 2005). Popular agent coordination techniques in which the triangulation problem has been addressed include DNA origami languages (Rothemund, 2006) and beacon based approaches (Werfel, 2002). Often, beacon based approaches make use of agents with awareness of the solutions sought. Such awareness may include knowledge of the coordinates of the vertices of the target (Werfel, 2002).

Mechanical processes have also been proposed for generating triangular structures (Kaewkamnerdpong et al., 2007). In these, agents are magnetized in order to attract each other towards desired formations. The same agents are structurally built into unique concave or convex shapes which would tessellate into desired structures, including triangles.

Research aimed at generating swarm based polygons such as rectangles, pentagons, hexagons, and even trapeziums have been reported as well (Seevinck and Edmonds, 2008.; Mason, 2002.; Kaewkamnerdpong et al., 2007). Nonetheless, the use of beacons to mark the vertices of the structures sought is dominant (Mason, 2002). Magnetized agents in the work of Kaewkamnerdpong et al. (2007), where the agents are tailor designed to tessellate into desired structures have also been popular for this purpose. Other models proposed the use of "no fit polygon" theories for the same purposes (Burke and Kendall, 1999).

The generation of swarm based circular structures is however rare. Nevertheless, natural ants have been shown to efficiently converge into circular formations when they are restricted to confined areas (Parrish et al., 2002). This observation has inspired the development of related circular structures in the works of Couzin and Franks (2002) and Butera (2002).

More complex structures such as the gear shape, oval shape, and diamond structures have also been simulated, where Chemotaxis methods are used (Eyiyurekli et al., 2013.; Bai et al., 2008). Even complex are the **E** and **X** shaped structures that have been generated in the works of Kondacs (2003) and Nagpal (2006) respectively. However the agents that are often used to create such complex structures predominantly possess multiple abilities. For example, they may have abilities to use both the growing point language (for filling spaces in the structures) and origami shape language (for shape scaling properties) (Nagpal, 2006).

Inspired by these works, we assess ant agents' abilities to achieve multiple targets location and construct geometric shapes. The ant agents we propose differ in that they neither support the use of growing circles and children agents (Nagpal et al., 2002), nor use origami languages (Rothemund, 2006). The use of beacons and global information to mark shape vertices is not supported either (Werfel, 2002). Similarly, tailor designed agents that can tessellate into desired formations (Kaewkamnerdpong et al., 2007) are elitist, hence not supported. Our ant agents are simple naive and autonomous, mainly relying on local interactions. Precisely, we assess our ant agents' abilities to form cross structures, four-way structures, polygonal structures (triangles, rectangles, pentagons, hexagons, heptagons, and octagons), as well as circular structures.

2.6 Conclusion of the chapter

The chapter mainly distinguished between interactive and non-interactive agent control systems with the goal of placing ant agent systems in the literature. In discussing each category, the chapter emphasized on identifying key concepts, methods, and theories with which to describe ant agent activities at individual levels.

Table 2.1 summarizes the key properties of most of the agent control models we reviewed. It shows the categories of agent systems (interactive or non-interactive). Each category is further split based on the agent interaction techniques they support (direct, indirect, mathematical, physics based, or intelligent systems). In addition, the table shows the different classes of agents that are common in the literature (path recalling, geometric, language based, optimized, stigmergic, calculus based, forces driven, mechanical, hybrid, or beacon and landmarks based agents), and highlight the communication media they commonly use (direct message passing, environment mediated, sensor based, vision, or hybrid mechanisms).

More so, the table shows the type of information that is used by each class of agents (stacks, vectors, chemicals, forces, landmarks, or beacons). Agent orientation strategies are also summarized in the same table (vector based, language based, probabilistic, calculated directions, forces based, or steered by landmarks).

category	**interaction**	**type of agents**	**communication**	**information**	**orientation**	**key activities**	**key factor**
Interactive	Direct	recall	message pass	stacks	stacks	read stacks	agent memory
		geometric	message pass	vectors	vectors	calculations	agent memory
		language	message pass	verbs	verbs	decoding messages	vocabulary
	Indirect	optimized	environment	chemicals	elitism	detecting chemical	elitism
		stigmergic	environment	chemicals	probability	detecting chemical	environment
Non-interactive	Mathematical	geometric			calculations	self localizing	agent abilities
		calculus			calculations	calculations	agent abilities
	Physics laws	forces	sensors	forces	forces	sensory actions	physics laws
		mechanical			equations	motion planning	mechanics
		hybrid	combined	forces	equations	motion planning	physics laws
	Intelligent	recall	vision	recall	landmarks	recalling landmarks	agent memory
		recall	vision	recall	beacons	locating beacons	agent memory

Table 2.1: Summary of the categories of agent control systems

Furthermore, we summarize the key agent activities (reading stacks, interpreting language verbs, detecting chemicals, self localizing, motion planning, or calculating directions), and indicate the key parameters of emergence that characterize each class of agents (agent memory, verbs, elitism, agent abilities, environment, laws of motion, or communication mechanisms). We make the following observations in this respect:

1. Generally, all agent interaction systems emphasize on agent o*rientation* and *movement* as the key ingredients for swarm intelligence. Orientation is guided by some form of meta information such as agents' sensory skills or *agent memories*. On the other hand, movement is commonly based on specific displacement factors such as attraction or repulsion effects. This observation inspires our selection of primitive behaviours with which ant agents achieve orientation and movements.

2. Successful agent orientation relies on the availability of locally perceived information around the agents (mathematical equations, geometry, forces, sensory factors, chemicals, or other agents). This information is updated regularly in order to appropriately inform the swarms. Our choice of pheromone update rules (dropping levels of pheromones, pheromone evaporation and diffusion), as well as vector modulation policies (message passing, detecting targets, and normalizing vectors) are inspired by this observation.

3. We learn about the requirement to design ant agents that possess some basic memory in which to keep important information regarding the tasks at hand. This observation inspires the design we follow when we represent ant agent memories and internal states.

4. Although agents remain autonomous, interactive systems often create a learning framework (Haasdijk et al., 2013) - both at individual and social levels - in which agents collectively engineer solutions from locally

shared information. Related interactive ant agents are fascinating devices, not because they are intelligent as individuals, but because they collectively achieve compelling emergent behaviours as swarms. Our choices to investigate the primitive behaviours of stigmergic and message passing ant agents is partly inspired by the cooperative nature of these classes of interactive agents. We are also inspired by the learning framework that arises, the dominance of interactive systems in achieving stable solutions, as well as the simplicity of related agents regarding memory requirements.

Table 2.2 summarizes the characteristics of the common measures of emergence that are observed in most agent control systems. Then, table 2.3 shows the common shapes that have been simulated as specific emergent behaviours in the past. These tables also show the types of agents that have successfully created desired shapes. We make the following observations in this respect:

- The description of amount of change translates to our understanding of system throughput. Our choice of average delivery rate as a measure of emergence is inspired by the characteristics, pros, and cons of this measure of emergence.
- Frequencies of interactions determine the average end-to-end delays per agent. On the other hand, convergence time indicates the speed of emergence in a system. On the contrary, the quality of products that are yield relates to quality of emergence. The characteristics, pros, and cons of these measures of emergence motivate our choice to assess similar measures of emergence on ant agent metaphors.

The value of this chapter is further emphasized by the following four contributions that it makes to the thesis:

Measure of emergence	typical measure	Pros	Cons
Amount of change	-mapping I/O	-I/O proportion	-understand processes
	-error frequency	-time based changes	-individual errors trivial
	-threshold analysis	-entropy fluctuation	
Frequencies	-agent interactions	- no. of interactions	-not all interactions are useful
	-priority probabilities	-chances of actions	-subjective
Time	-amount of simulation	-time in simulation	-ignore outliers
	-ergodicity		
	-time to converge		
Quality	-timeliness	-adherence to schedules	- hard to find limits
	efficiency	- levels of engagement	
Correlation analysis	on metrics	-supports centrality tests	
	-meanings to parameters	-reveals relationships	
	-influences of events		

Table 2.2: Pros and cons of particular measures of emergence

Shape	characteristics of agents
point	mostly hole making ant agents
line	path finding ant agents,
cross	growing circle theory, reproductive agents, use power laws, parents are leaders
triangle	DNA origami language, agents often have awareness, some agents are mechanical
polygons	mechanical agents dominate, but beacon based agents are also many
circular	ant agents
shapes e.g. X and E	chemotaxis, growing point theory, origami language

Table 2.3: Common geometric structures

1. Categorization of agent control systems, as summarized in table 2.1, positions ant interaction systems in the literature. We believe that the XSets we propose and evaluate in future chapters are motivated and inspired by some of the concepts and theories in this classification.

2. The parameters of emergence that are discussed in section 2.3 inspire the choices we make of the quantifiers we propose. The same quantifiers

show potentials to successfully detect emergency in other agent systems in the future.

3. Quantification of the degree of emergence in a system is critical. The survey we provided in this regard highlights the pros and cons of using specific measures of emergence in different situations, thereby motivating our choices of which measures of emergence to use in the context of this work. The pros and cons we refer to are summarized in table 2.2.

4. Works in which specific emergent behaviours have been investigated indicate a common bias towards simulating geometric emergent structures. In these, specificity is defined in many ways. We get inspirations from these reviews regarding particular emergent behaviours to simulate. Table 2.3 summarizes the top ten geometric structures that have been achieved in this respect, indicating at least the type of agents that has been used for these purposes.

The next chapter explores the computational representation, and detailed description, of the ant agent activities we identify.

Chapter 3

Ant Agent Primitive Behaviours

3.1 Introduction

Figure 1.1 in Section 1.2, as well as the system architecture shown in Figure 3.2stipulate our obligation to address two issues of this thesis in this Chapter. First, we are required to characterize meta information which we defined in Section 1.2.1 (and in Figure 1.1) as those parameters of the simulation system which spell out when (indicated in ant agent internal states) and how (stored in ant agent memories) primitive behaviours are used, and by which ant agents (indicated by the type of ant agent and the agent density supported).

Thereafter, we are required to investigate the primitive behaviours which characterize ant agent activities in the two categories we study. A motivation for the choice to investigate the primitive behaviours of these two classes of ant agent metaphors was presented earlier on in Section 1.4.2. However in summary, we cannot consider language based metaphors (Nagpal et al., 2002.; Nagpal et al., 2003.; Sussman, 1999.; Belani et al., 2002.; Stefano and Santoro, 2001.; Kraus and Lehmann, 1995.; Nagpal and Coore, 1998.; Cao et al., 1997.; Butera, 2002.; Beal 2005a, 2005b.; Abelson et al., 2000.; Nagpal et al., 2006.; Cranefield et al., 2000) since literature lacks sufficient vocabulary (Nagpal et al., 2002) to describe agent verbs and the semantics of the language. In addition, language based models grossly compromise the independence of receiver agents in the communication circles (Dastani et al., 2003) - a feature which is not attractive for the ant agent devices we propose.

In addition, this work neither recommends mathematical (Ngo et al., 2005.; Harris, 2007.; Trofimova et al., 1998) nor physics based metaphors (Spears et al., 2004a, 2004b, 2005.; Azzag et al., 2007.; Beckers et al., 1989.; Balch and Arkin, 1999.; Cao et al., 1997) because their design mainly consists of complex equations and knowledge bases which basically characterize sophisticated robotic actions. The ant agents we propose in this work are naive devices that can follow very simple rules at individual levels (Chibaya and Bangay, 2007).

Most message passing metaphors that have been tried in the literature prescribe agents which support large memory capacities (Trianni and Dorigo, 2005.; Rajbhupinder et al., 2010.; Hara and Ichimura, 2005.; Montes De Oca et al., 2005.; Lien et al., 2005.; Rodriguez et al., 2007.; Bayazit et al., 2002.; Bayazit et al., 2005.; Caicedo et al., 2001) - a design feature which is also unattractive for the simple and naive ant-like devices we propose. However, a novel class of ant agent system can be derived from related theories in this category. Precisely, ant-like agents can be designed to use local interaction rules in which they explicitly share direction vector components which indicate the preferred directions of motion (Wu et al., 2005.; Nasipuri and Li, 2002). Vector arithmetic is a general algebra that is applicable for computation in many domains (from machine learning, network analysis, and spatial representation) so has potential to allow ants to solve problems relevant to these domains. In this case, it also allows location reasoning which is important for spatial problems.

Message passing ant agents can use the shared vector components to perform independent calculations out of which they can orientate (Ngo et al., 2005). Thus, swarm level successes are based on the quality of the resultant vectors that are calculated in each ant agent step using the shared direction vectors. The message blocks that are shared can be in the form of stacks (Trianni and Dorigo, 2005) of $(x; y; z)$ vector components. Implicit communication spaces arise in which these message blocks are passed between neighbouring ant agents one-on-one (Viana et al., 2007.; Xu et al., 2008). The message blocks are always similar in structure (Caicedo et al., 2001). To the best of our knowledge, detailed investigations and application of such a framework to the ant agent problem is novel. The message passing model we propose in this work is thus new, motivated by these promising features. As a result, most of the primitive behaviours we propose in this respect are deduced from logic and innovational in the field.

The bulk of interactive agent control systems are environment mediated (Dorigo, 1992.; Dorigo et al., 1999.; Dorigo et al., 1996.; Di Caro et al., 2004.; Negulescu et al., 2006.; Chibaya and Bangay, 2007.; Montgomery et al., 2007.; Panait and Luke, 2004a, 2004b, 2004c.; Cavalcanti and Freitas, 2005.; Cavalcanti et al., 2005b). The primary advantage of environment mediated ant systems is that the ant-like devices thereof would require minimal communication using very little processing power (Panait and Luke, 2004a). The same ant agents can tolerate a degree of agent error (Mason 2002) without jeopardizing the completion of the task at hand.

In our search for the key concepts, methods, and theories around ant agent activities, we take advantage of profound documentation of ant systems in this category. We remind the reader at this point that our aim in this thesis is not to report the forms of emergent behaviours that arise. Rather, we are interested in the actions of ant agents in swarms, actions which give rise to emergent behaviour. Identification of these low level actions requires us to take a detailed and reliable literature survey. Such detailed and reliable literature is prevalent in the stigmergic ant agent group - hence the choice we make to study this class of ant agents over relatively new and poorly explored ant system models.

Although many other ant system metaphors could be studied and potentially give similar or more insightful results, our premise is that any further studies on related ant control metaphors would merely extend the set of primitive behaviours - U that we present in this work, thus raising the cardinality of the power set $P(U)$ and widening the search space for even better and more generic XSets with potentially a wider application domain.

3.1.1 Problem statement

The two particular issues we address in this chapter can be re-phrased as follows:

1. *How do describe meta information M, which sets forth the parameters for spelling out when and how ant agents use particular primitive behaviours that are included in an XSet?* This question requires us to present our assumptions upfront, the technical setup of our simulation system, the design of the components and parameters of the system, as well as the design of the ant agent we propose regarding memory and internal states. In doing so, we emphasize on the computational representation of these meta items, and motivate why each is relevant to the ant system we propose.

2. *Which ant agent activities describe the domain of primitive behaviours that can allow emergent behaviour to occur?* - Once the system design

issues are in place, the second question requires us to take a detailed survey of the commonly inferred agent activities and relate these to the context of stigmergic and message passing ant agents - thus identifying common primitive activities and the ingredients of these primitive behaviour in computational terms.

Consequently, the key outcome of this Chapter is an explicit set of primitive behaviours U. This set of primitive behaviours is the basis for the answer to the formulated research problem, as well as the basis for a system that will function as a proof of concept in Chapters 4, 5, and 6.

3.1.2 Overview of the chapter

The rest of the sections of this chapter are arranged as follows:

- Section 3.2 presents a detailed architecture (how the system is laid out) of the simulation system we propose. In this section, we firstly present our assumptions regarding the purpose of the swarms, the design of the ant agents we use, as well as assumptions regarding the environments in which ant agents reside. Thereafter, we discuss the technical setup of the system in Section 3.2.2, followed by a description of the components and parameters of the same system (Section 3.2.3).
- The first question of this chapter (see Section 4.1.2 for this aspect of the thesis) is addressed in Section 3.2.4, where we describe the design of the ant agents we propose, particularly characterizing their memories and internal states.
- Figure 3.1 summarizes the sub sections of Section 3.2.5 which presents the bulk of the work of this chapter. Precisely, this section identifies and justifies the various lemmas which we accept in order to find out

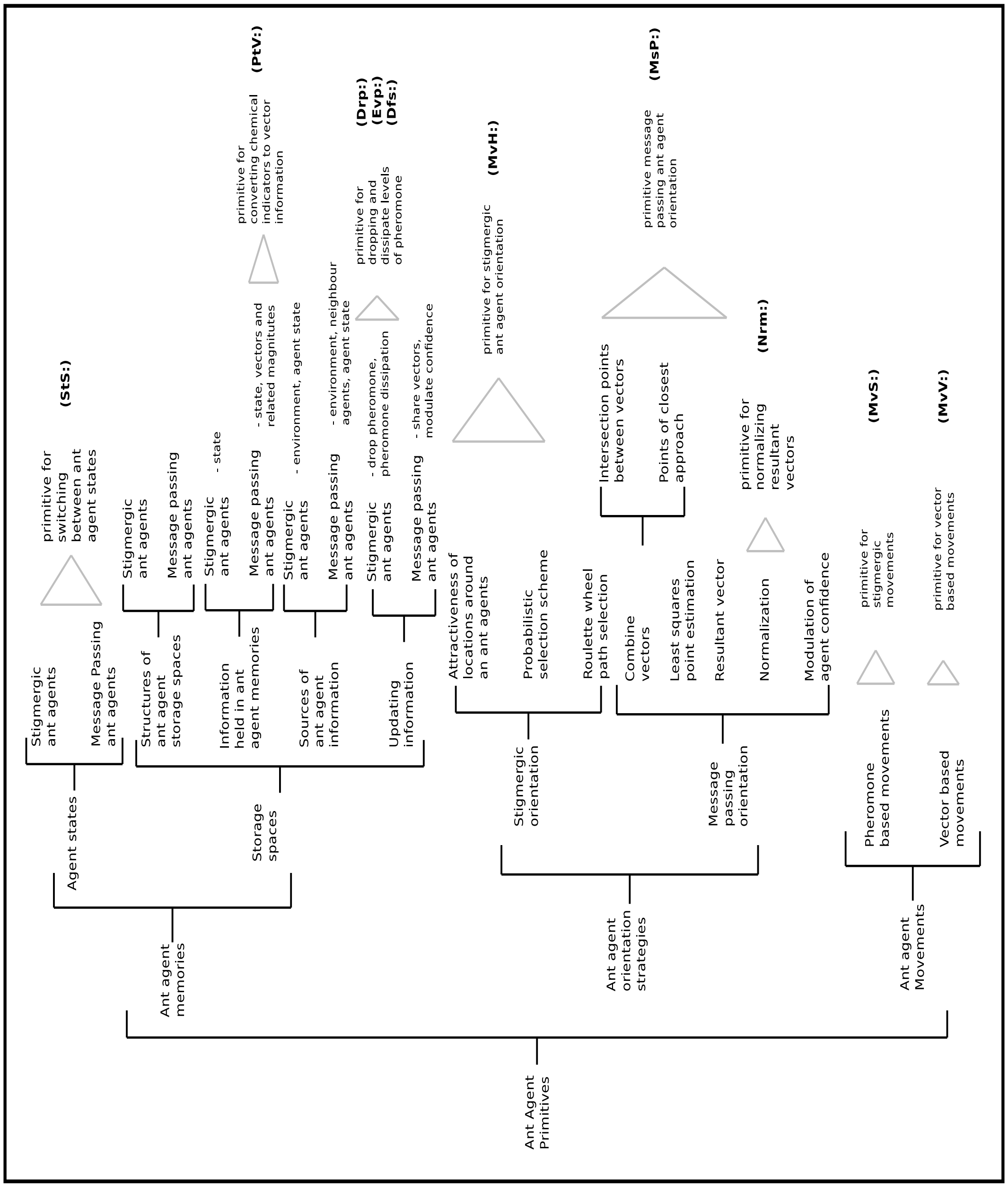


Figure 3.1: Overview of chapter 3

whether the XSet approach works. Discrete mathematics define a lemma as a minor result whose purpose is to help in proving a theorem (a stepping stone on the path to proving a theorem). The primitive behaviours we propose, as well as the design choices thereof, are proposed as building blocks for proving the concept of XSets. We describe the ingredients which characterize each of these primitive behaviours in computational terms.

- We conclude the chapter in section 3.3, summarizing the work presented, as well as highlighting the contributions of the chapter to the thesis and the board of knowledge.

3.2 System Architecture

3.2.1 Assumptions

As an anticipated setting for our simulation system, consider a swarm of ant-like devices that are deployed in an unknown continuously wrapping environment. Every ant agent's goal is to locate food-like resources that are situated somewhere in this environment, and upon finding these resources, travel back to a nest-like starting point (Chibaya and Bangay, 2007.; Panait and Luke, 2004a, 2004b.; Solimanpur et al., 2005). Although both the food-like resources and the nest-like starting point can be placed at randomly picked location of the environment, we assume fixed location in order to achieving fair experiment results. This is a common ant problem setup in the literature from which we will likely gather sufficient theories, methods and ideas regarding ant agent activities at individual levels.

We make an assumption that all ant-like devices in this system are very simple and naive. They do not have neither *a prior* knowledge of the environment in which they reside, nor knowledge of the positions of the targets sought. All ant agents are assumed to remain in motion throughout the simulation, travelling at a constant speed of one grid step per cycle. Note that variation of ant agent speed is not a subject of study in this work. The movement speed of our ant agents is implicitly regulated to remain constant (Bayazit et al., 2002.; Lua et al., 2005.; Parrish et al., 2002).

Although ant agents do not have knowledge of the environment, a learning framework often arises (Lien et al., 2005) with which global-level perceptions are built, either on the environment (stigmergic ant agents), or in ant agent memories (message passing ant agents). To learn from others, ant agents reference (directly or indirectly) other ant agents' historic experiences (Rodriguez et al., 2007.; Bayazit et al., 2002.; Bayazit et al., 2005.; Nasipuri and Li., 2002). Often, the referenced information expands the ant agent's awareness of the environment in which it resides.

Please note that ant agents do not require any physical contact with one another in order to learn from each other. Instead, it is sufficient that ant agents read and interpret each other's perceptions (directly or indirectly). In the message passing ant agent context, implicit communication spaces arise in which vectors are sent across spaces between adjacent neighbours. In our case, detection of proximity is encoded in ant agents' abilities to create local coordinate systems (see section 3.2.5.2 for details regarding local coordinate systems). These abilities also rely on the internal states of the ant agents (Balch and Arkin, 1999). We presume that ant agents can co-exist, implying that we do not assume one ant agent max per location.

As a case study, we propose obstacle free environments since environment complexity is not a subject of study in this work. In addition, inclusion or exclusion of obstacles does not connote any special ant agent design requirements, particularly at individual levels. We assume environments in 2D or 3D. The sizes of these environments can vary. Each cell of the environment is, in fact, a tuple which keeps record of a collection of data regarding ant agent activities on that cell at each time (Negulescu et al., 2006.; Haasdijk et al., 2013). Other examples of data that is held in the cells of the environment include positions of target objects (Valckenaers et al., 2001), positions of starting points (Mullen et al., 2009.; Sudd, 1960.; Montes De Oca et al., 2005), or chemical markers (Panait and Luke, 2004a.; Babaoglu et al., 2006). An environment also provides a platform on which the agents reside (Haasdijk et al., 2013). Most importantly, they also handle the high level description of the emergent outcome sought (Seevinck and Edmonds, 2008.; Mason, 2002).

Our system is not suggested as a replacement or an improvement to any existing ant system in the literature. It is rather, an alternative approach for describing ant agent languages that may allow deliberate engineering of emergent behaviour over time. Although comparisons of the outcomes of this work with the results yield in similar traditional studies are of importance, this work emphasizes on justifying the validity of the XSets approach as an ant agent design paradigm. We are content with the demonstration of the functionality of XSets as toolboxes for desired emergent behaviour.

We make another assumption that all ant agents are identical. Agent density is variable depending on the size of the environment in use at the time. Ant agents in a swarm operate synchronously, executing the same sets of primitive behaviours all the time. The measures of emergence that are required are based on information with which ant agents interact (levels of pheromone or direction vectors). In this work, ant agent neighbourhood is restricted to one grid cell around the ant agent ((see section 3.2.5.2 for details regarding local coordinate systems).

Figure 3.2 shows the important design aspects of our system. Precisely, it emphasizes on three key areas of interest: (a) identification of agent information - indicating the data that is necessary for the simulation to start, (b) some search heuristic for identifying useful combination of agent information, and (c) evaluation of best combination of agent information for application in different problem domains. This chapter addresses the first of these three aspects. The rest of the issues are addressed in future chapters of the thesis.

3.2.2 Technical setup

As a by-product of this research, we propose a simulation system which searches for novel control processes (XSets) that would drive ant-like devices towards emergent behaviour. We propose a Single-Instruction-Multiple-Data (SIMD) agent simulator which runs on a general-purpose graphics-processing-unit (GP-GPU) architecture. This system makes use of a standard technology adapted to handle large numbers of identical ant-like devices, all limited in memory, computing, and communication capabilities.

Although the requirement for high processing power and computer memory can be met in principle by most machines today, our simulator is designed to run on a Supercomputer (hydra) which minimizes limitations related to CPU speed. However for illustration purposes, most of the test results we present in this work are recorded from simulations that were coordinated on an Intel(R) Core (TM) i5 CPU, M450 @ 2.4GHz with 3GB RAM.

3.2.3 System components and parameters

Our ant agent control algorithms are all implemented in C on a CUDA platform which gives us direct access to the virtual memory and increases computing performance by harnessing the power of the GPU. However the visualization modules are written in C++ using Qt widgets in Qt Designer. The system consists of mainly three subsystems: the central resource, the XSets generator, and the visualizer.

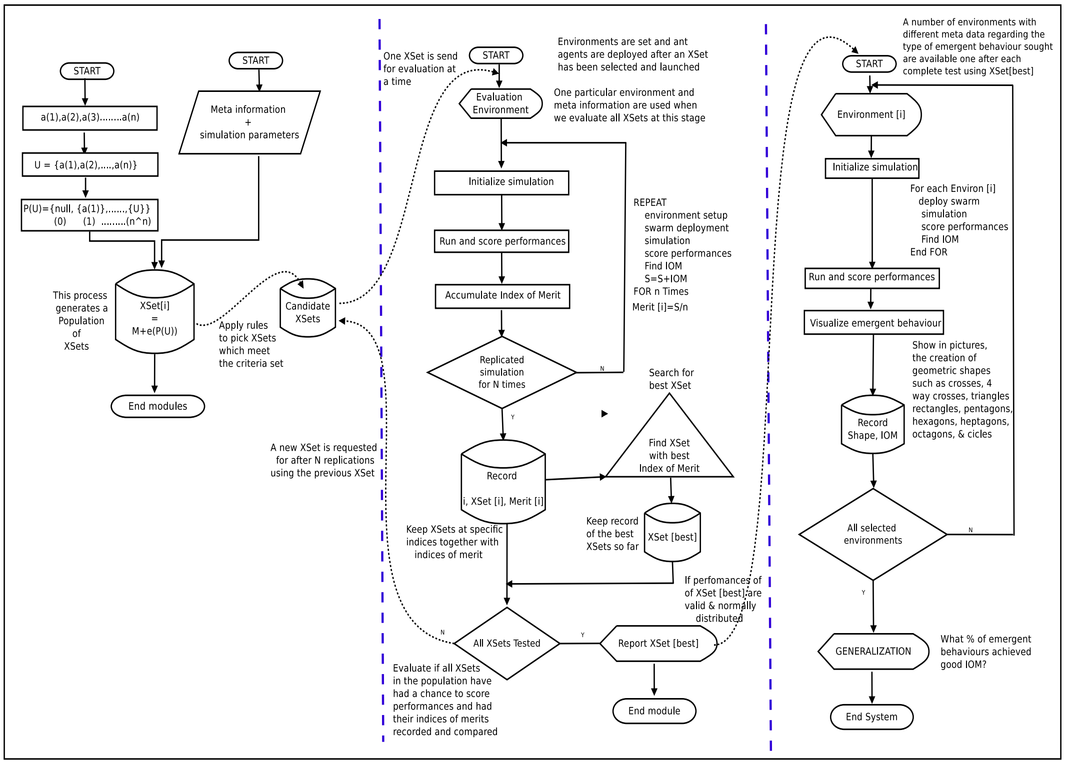


Figure 3.2: A complete system architecture

The central resource - is loaded first. This subsystem mainly manages all interactions between the graphical objects of the system. It defines the canvas (a bitmap drawing surface) and acts as a repository for the objects that are shared in the system. Most importantly, this subsystem verifies that the architectures available are compatible with the hardware assumptions, and double checks that the selected parameters and meta information correspond to the tasks.

XSets generator - addresses three key tasks of the system. First, it prompts for specific user level data entries. Then, this subsystem uses the meta data provided and the parameters that are input by the user to create a population of XSets. Remember that XSets are created by adding meta information to every possible ordered element of the power set (i.e $XSet = M+ \in_i P(U)$, such that $i \in [0; n^n]$, where n is the number of primitive behaviours under consideration). The remainder of the parameters are used to create an inference engine which selectively isolates those XSets that satisfy some criteria set. Figure 3.2 shows that these three tasks are all addressed in the left column of our system architecture. Among the key user choices are the following:

- *Ant type:*- there are three possible choices, namely; stigmergic, message passing, or hybrid. Selecting a stigmergic ant type sets a rule for the inference engine to isolate XSets that are formed using stigmergic primitive behaviours only. The same rule would apply when a message passing ant type is selected. Hybrid ant type considers XSets that are made from any combination of primitive behaviours regardless of their origins.

- *Environment size:*- the user is allowed to indicate the number of cells of the environment in each direction, assuming a square environment in 2D or a cubical environment in 3D. This feature allows flexibility when we test swarm performances. The default assumption is that all evaluation environments are 100×100 grids in 2D.

- *Environment dimension:*- our simulator allows visualization of environments in 2D or 3D. For illustration purposes, and as proof of concept, we report results that are extracted from simulations in 2D environments. This is because environment complexity is not a subject of study in this work.

- *Evaluation environment:*- this parameter selects an environment to be used at a time. Environments include in their properties the fitness functions which stipulate the type of emergent behaviour sought. In this work, we developed ten evaluation environments (path finding environment, cross formation, four-way cross, triangle, rectangle, pentagon, hexagon, heptagon, octagon, and circle formation environments). The default choice is the path finding environment.

- *Scoring time:-* this is the time frame in which the system is allowed to score swarm performances (speed of emergence, quality of emergence, average delivery rate, average end to end delays, and Shannon's information value) before an index of merit is calculated. In this work, time is measured in iterations.

- *Agent density*:- this parameter indicates the number of ant agents that are deployed into the environment at a time. We set the default agent density to 5000 ant agents.

- *Ant memory:-* this parameter indicates the number of message blocks an ant agent can hold in memory at a time. In C, this is a vector data structure. Smaller values are desirable since that would comply with our dictum to design simple and naive ant agents. However one has to allocate sufficient memory blocks to, at least, allow every ant agent to hold state information. Our default setting is 4.

- *Cardinality:-* this parameter indicates the maximum number of primitive behaviours that an ant agent would execute in each internal state. The choice of this parameter is mainly based on one's perception of the task. However users often cannot predict this correctly. Choosing a bigger value than required will not jeopardize the XSet since a filler primitive behaviour (NOp :) can be used in any extra slots.

- *Internal states:-* this parameter indicates the number of internal states each ant agent can support. The decision regarding how many internal states one can choose often depends on one's high level interpretation of the task as well. Our default setting is 4.

- *Pheromones:*- this parameter indicates the maximum number of the levels of pheromone each ant agent in the swarm can interpret or perceive. Thus, our system can be adapted to support single pheromone metaphors, two pheromone metaphors, or multiple pheromone metaphors, as suggested in various related works (see chapter 2 for this categorization).

Our premise is that giving users the choice to provide specific parameter values enhances system flexibility and adaptability in different contexts. However we simplify the system by proposing default values in each case.

The visualizer - basically addresses the requirements of the middle and right columns of Figure 3.2. It sequentially accesses the XSets which meet the criteria set in the inference engine (defined by user defined parameters and meta information), and allow swarms of ant agents to work towards achieving the emergent behaviour whose fitness functions are characterized in the selected evaluation environment. It scores the performances of the XSet thereof, and report their indices of merit. Our visualizer allows users to see the configuration of the XSets in use, as well as to see the visual performances of the swarms thereof (showing mobile ant agents in action). Once instantiated, the parameters of the visualizer are stored in the system and are effectively immutable. Thus, further manipulation of the parameters or the XSet itself will have no effect on the visualization process. As a result, multiple visualization is supported without interfering with one another. This feature allows users to visually compare the outcomes of two or more XSets at a time.

Although most of the system parameters are held on the evaluation environment, management of conversions into serialized formats and subsequent transfer to the GPU to facilitate accelerated simulations, as well as transportation of parameters over the network to the visualization subsystem, is an independent functionality.

3.2.4 Ant Agent Design

This section describes our view of the design of an ant agent. Key is the design of our ant agent memory which we discuss in Section 3.2.4.1. We emphasize on the type of data that is stored in these memories including ant agents' internal states (discussed in Section 3.2.4.2), as well as message passing ant agent vectors (discussed in Section 3.2.4.3) and vector weights (discussed in Section 3.2.4.4).

3.2.4.1 Ant agent memories

Characterization of agents by their memory capacities is very common (Ngo et al., 2005). Generally, agents require some basic memory in which to record specific meta information (Dorigo and Blum, 2005), internal states (Wu et al., 2005), landmarks and beacons (Wu et al., 2005), roadmaps (Rodriguez et al., 2007), or navigation equations and velocity control functions (Ngo et al., 2005). Conforming to this norm, and in line with our dictum to prescribe simple and naive ant agents, our ant agents are designed with basic memory in which they keep navigation information.

Figure 3.3 shows the composition of stigmergic and message passing ant agent memories. We discuss the memory contents of each type of ant agent in the following three subsections.

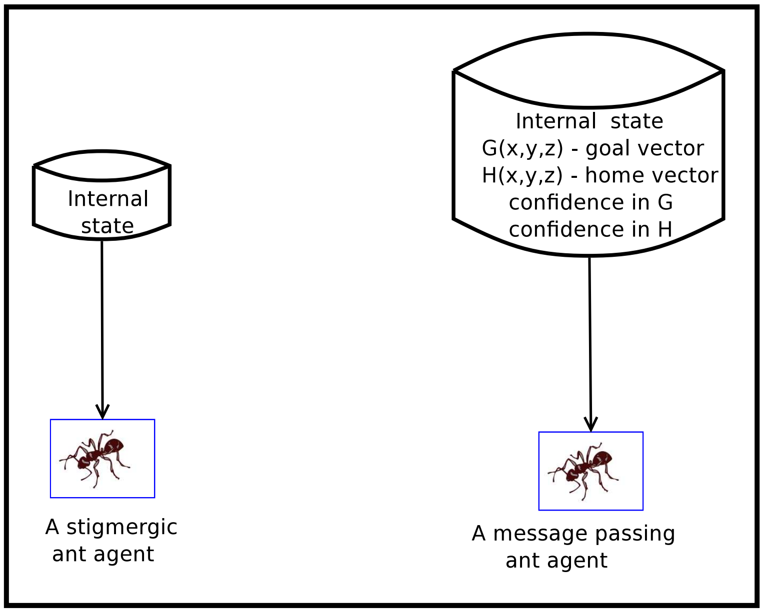


Figure 3.3: Ant agent memories

3.2.4.2 Ant agent internal states

Internal state is the most important and common parameter of emergence that is required by most agent control systems (Chan, 2011.; Dorigo and Blum, 2005). Ant systems in particular, commonly place internal state information in the agent's memory (Merkle et al., 2006). This has the advantage of speeding up information access when ant agents make path decisions.

We view internal state as a logical field in which one, and only one option, is possible at a time. It is a self-contained computational object whose contents are invisible to other agents, but would influence other agents' internal states (Parunak, 2005). Please note that this work does not view ant agent internal states as state variables as in the case of cellular automata, but rather as merely Boolean flags. As a result, an ant agent switches between

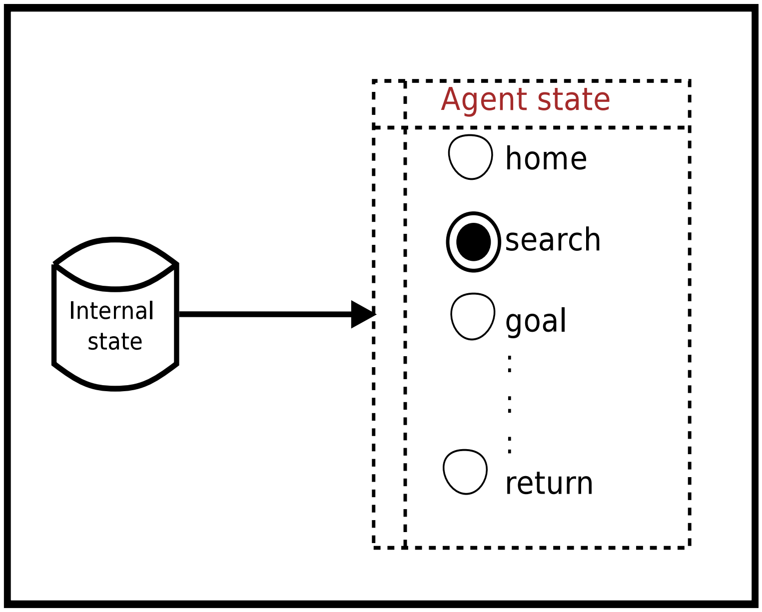


Figure 3.4: Ant agent internal state

these mutually exclusive logical options when particular system conditions are satisfied.

In the literature, agents flip between different internal states based on their location on the environment (Parunak, 2005). Sometimes they flip between internal states as a reward for their actions (Panait and Luke, 2004a). It is also common to trigger state changes based on the behaviour of neighbouring agents (Soloveichik, 2008.; Shalizi, 2001). However in this work we adopt the views of (Panait and Luke, 2004a) where an ant agent flips from one internal state to another as a reward for successfully searching for a target. Figure 3.4 visualizes our ant agent's internal state setup.

By default, ant agents are deployed in the seek mode. Such basic knowledge (indicating whether the ant agent is searching for food sources or returning to the starting point) spells out the ant agent's purpose at the time (Panait and

Luke, 2004a). It influences the way other ant agents in the swarm interact with that agent (Soloveichik, 2008.; Shalizi, 2001). In stigmergic ant agent context, internal state determines which levels of pheromones the ant agent will drop in each step, as well as which levels of pheromones are attractive or repulsive at the time. This is useful information when stigmergic ant agents orientate (see section 3.2.5.1 for this aspect of the thesis).

On the other hand, message passing ant agents use state information to determine which vectors to read from neighbours, as well as those vectors to penalize. They would also appropriately associate vector weights to corresponding vector components. Likewise, this information is important to message passing ant agents when they orientate and make movement decisions (see section 3.2.5.2 for this aspect of the thesis).

Lemma 1: *the ability of an ant agent to review it's internal state in each step, and switch to appropriate internal states when it becomes necessary, is a discrete autonomous action that is undertaken at individual levels* - hence a primitive behaviour in ant systems.

- We defined a primitive behaviour in Section 1.1 as an ant agent's discrete activity at individual levels. This is a view we make which will be validated in chapter 4 when we evaluate XSets for causal properties. Algorithm 3.1 shows the computational interpretation of how an ant agent evaluates and switches from one internal state to another. This is an innovative algorithm of our own making which resulted from various pre-tests and evaluations.

- Let $(StS : m, n, x)$ represent the mnemonic of this primitive behaviour. This is, in our context, a mnemonic for (*Set State to* : m if n is true in x). In this mnemonic, m is the ID of the new internal state the ant agent would conditionally switch to. Internal state IDs are integer parameters ranging from 1 to the number of internal states that are

Algorithm 3.1 Switch internal state

```
/* DEFINITION OF PARAMETERS

    s_t ← internal state at time t
    n ← input to a condition
    x ← domain of the problem
    m ← new internal state
    s_{t+1} ← internal state at time t+1
    L ←current location

*/
(StS : m, n, x)
{
    foreach agent i at L
    {
        if (n is true in x)
            s_{t+1} ← m
    }
}
```

supported (NB: the number of internal states supported is a user entry - see section 3.2.3). On the other hand, n is a condition which indicates what aspects of the simulation would trigger an ant agent's desires to switch from the current internal state to another. Then, x sets the domain in which n must be satisfied before internal state changes are effected.

- For example, $(StS : 1, 0, 0.5)$ is read as: *Set internal state to* 1 *if the levels of pheromone whose ID is* 0 *are above* 0.5.

3.2.4.3 Message passing ant agent vectors

Figure 3.3 also shows that a message passing ant agent requires relatively more memory space than stigmergic ant agents. This is because message passing ant agents are designed to also keep a record of the direction vectors that point in their perceived directions of the targets and the starting point. Inspired vector geometry and by the work of Trianni and Dorigo (2005), where $(x; y; z)$ vector components are shared as message blocks and used to calculate new directions, our message passing ant agents similarly share message blocks of the format: $(x_i, y_i, \vec{s_v}, s_w, \vec{r_v}, r_w)$ and use this information to determine resultant vectors. In this tuple, $(x_i; y_i)$ indicates the relative offset of the ant agent from the receiver ant agent. Only those ant agents whose offsets are one cells away; $((0; 0)$, $(0; 1)$, $(1; 0)$, $(0; -1)$, $(-1; 0)$, $(-1; 1)$, $(1; -1)$, $(-1; -1)$ or $(1; 1))$ can share message blocks. This assumption insinuates that message passing ant agents can self-localize relative to their local neighbours.

The components; $\vec{s_v}$ (search vector) and $\vec{r_v}$ (return vector) are respectively geometric vectors that point in the perceived directions of the target and starting point. On the other hand, s_w and r_w are the respective weights of these geometric vectors. These geometric vectors and their weights are updated and hopefully improved in every ant agent step in order to incorporate and acknowledge the views of neighbouring ant agents. In this work, we view the ability of an ant agent to determine the next direction to follow, and that ability to update its knowledge (ant agent orientation) as an autonomous low level activity - hence a primitive behaviour. These abilities are discussed in details in Section 3.2.5.2.

3.2.4.4 Message passing vector weights

We mentioned earlier on that the vectors that are carried in message passing ant agents' memories are associated with some weight parameters which indicate how well an ant agent has performed in the past. This is not the first time confidence weights have been proposed for attaching a trust factor on paths (Zarnani and Rahgozar, 2006). In our view, vector weights reflect the quality of the information that is shared between ant agents.

We propose vector weights that are float indicators between 0 and 1 inclusively. A vector whose weight is 1 is regarded as distinctly pointing to the location of the target. This would indicate that the ant agent that holds the corresponding vector "knows" where the target is and must be trusted more when neighbouring ant agents determine their own paths.

Equation (3.1) shows how the weights of direction vectors are adjusted in each ant agent step. To the best of our knowledge, this is a new update rule that has not been used before. In this equation, let $w_i(t)$ represent the weight of vector i at time t. Suppose there exist k neighbouring ant agents around the ant agent that is holding vector i. If we denote each of the neighbouring ant agents' vector as j, then the expression $\frac{\sum_{\forall_k} w_j(t)}{k}$ indicates the average weight of the weights of neighbouring ant agents' vectors at time t. This average weight indicates the average trust of neighbouring ant agents in the vectors they are following.

However, we acknowledge that, although average values desirably determine the central tendency, they often neglect outlier cases. As a result, we establish a dispersion factor which indicates how similar or how spread the views of neighbouring ant agents are. This dispersion factor is, in fact, the standard deviation of the weights of the vectors that are carried in neighbouring ant agents' memories.

Let the dispersion factor sought denoted as c. Mathematically, this value is smaller when the weights in neighbouring ant agents' memories are relatively similar,and bigger when the weights are too different. However these values are remain in the range 0 to 1 because all the weights are within this range as well. To penalize bigger values in favour of smaller standard deviations, our system uses $(1 - c)$ as the dispersion factor. Thus, a bigger dispersion factor implies consensus among neighbouring ant agents and trust in the information received.

An average measure between the ant agent's own weight of its direction vector$w_i(t)$ and the weighted average weight of the neighbours, give the updated weight of the direction vector which the ant agent is following next - $w_i(t+1)$. Some minimal degree of randomness λ is added to the resulting weight in order to enhance the ant agent's independence in subsequent actions. That randomness is especially useful when the ant agent is isolated.

$$w_i(t+1) = \frac{1}{2}\left(w_i(t) + \frac{\sum_{j\in k} w_j(t)}{k} * (1-c)\right) + \lambda \tag{3.1}$$

Variable	Meaning of variable in equation 3.1
$w_i(t+1)$	vector weight of i^{th} agent at time $t+1$
$w_i(t)$	vector weight of i^{th} agent at time t
$\frac{\sum_{j\in k} w_j(t)}{k}$	average vector weight of k neighbours
c	stdev of vector weights of k neighbours
λ	some degree of randomness

In this work, a message passing ant agent's ability to update the weight of the direction vector it is following is incorporated in the ant agent's orientation processes because they concurrently occur. We discuss the movement, orientation, target detection, and information update rules in the next section.

3.2.5 Primitive behaviours

Generally, agents in the literature are designed with abilities to ***move*** around the environment in search of specific targets to ***detect*** e.g. searching for food sources (Dorigo, 1992.; Cordon et al., 2002.; Panait and Luke, 2004b), tracing chemical sources (Naeem et al., 2007), moving towards a light source in formation (Spears et al., 2004a), detecting wound-like targets (Schneider et al., 2006), or searching for cancer infected cells (Cavalcanti et al., 2006a). To make useful movements, agents must ***orientate*** appropriately before taking a step (Dorigo et al., 1999.; Chibaya and Bangay, 2007.; Panait and Luke, 2004c). Orientation is based on local information around an agent (Panait and Luke, 2004b). This information is ***updated*** regularly (Chibaya and Bangay, 2007).

This section interprets the ant agent framework we propose in technical terms. It proposes and characterizes unit level primitive behaviours for achieving ant agent orientation, movement, information update, and detecting targets. This is not the first time unit level agent actions have been explicitly mentioned as drivers of ant agents (Haasdijk et al. ,2013.; Winfield et al., 2013). However, explicit interpretation of these primitive behaviours in computational terms is an innovative approach in this work.

3.2.5.1 Stigmergic ant agent orientation

We characterized a stigmergic model as an indirect and environment mediated ant agent interaction model in which virtual pheromone chemicals are the key ingredient for ant agent orientation (Dorigo et al., 1999.; Montgomery et al., 2007.; Nakamichi and Arita, 2004, 2005). We also characterized our stigmergic model as a multiple pheromone interaction system (see section 2.2.2.2 for details regarding this categorization). An ant agent can only

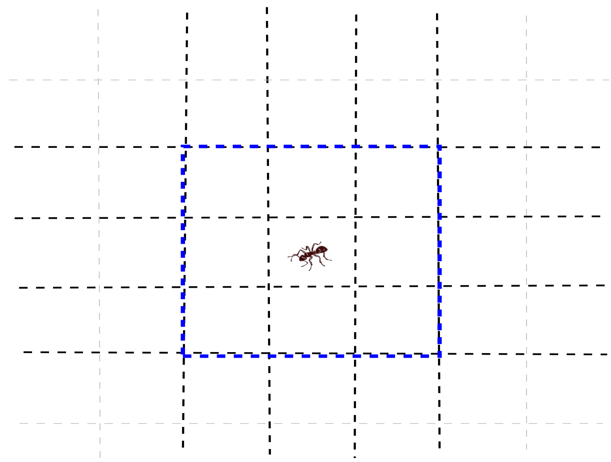


Figure 3.5: An ant agent's local environment

perceive and use the levels of pheromones that are held on its local environment (Panait and Luke, 2004a.; Chibaya and Bangay, 2007).

Figure 3.5 visualizes an ant agent's local environment which defines the agent's radius of vision. It shows eight possible directions an ant agent can follow. To orientate, a stigmergic ant agent's first task is to determine the relative attractiveness of the eight possible destination locations around (Panait and Luke, 2004c). A location's attractiveness is based on the concentration of the levels of attractive and repulsive pheromones it holds (Chibaya and Bangay, 2007). The attractiveness of each level of pheromone is based on the ant agent's internal state. In the path finding context, attractive levels of pheromones are those that have been placed on a location by ant agents that visited that location when they were in an opposite internal state to the current ant agent's internal state. Repulsive levels are those that have been placed on a location by ant agents that were in the same internal state when they visited the location. The notion is that an ant agent would rather follow the trails that are formed by ant agents in the opposite state because they have, at some point in simulation, found their seek targets.

The stigmergic orientation model we propose is inspired by the works of (Chibaya and Bangay, 2007) and (Panait and Luke, 2004a,2004b,2004c). Let a neighbouring location around an ant agent be denoted as L. For each L, an ant agent retrieves the levels of attractive and those of repulsive pheromones. For illustration purposes, let the levels of attractive pheromones at location L be denoted as τ_L and the levels of repulsive pheromones that can co-exist at the same location L be denoted as η_L. The the sum of the levels of attractive pheromones around the ant agent can be expressed as $\sum_{k\in N} \tau_k$, and that of the levels of repulsive pheromones can be expressed as $\sum_{k\in N} \eta_k$, where $N = 8$, indicating the number of possible destination locations around the ant agent.

We can find the relative weights of each level of pheromone on every L by dividing the concentration of the same levels of pheromones by the sum around the ant agent (thus giving $\frac{\tau_L}{\sum_{k\in N} \tau_k}$ and $\frac{\eta_L}{\sum_{k\in N} \eta_k}$ for each L). Our simulator initializes all levels of pheromones to a negligibly very small quantity in order to avoid overflow errors - division by zero - when we calculate these weights.

The final attractiveness value is obtained by subtracting the weight of repulsive levels of pheromones from the weight of attractive levels of pheromones at the same location. Let the attractiveness value of a location L be denoted as A_L. Equation (3.2) defines this relationship in mathematical terms (taken from (Chibaya and Bangay, 2007)). It is possible to get negative attractiveness values when the weights of the levels of repulsive pheromones are higher than the weights of the levels of attractive pheromone. We geometrically translate such attractiveness values relative to the smallest measure found. This is done in order to avoid having locations with negative chances of being selected. Let the adjusted attractiveness value of a location L be denoted as A_{L_T}. Equation (3.3) illustrates this adjustment.

$$A_L \leftarrow \left(\frac{\tau_L}{\sum_{k\in N}\tau_k}\right) - \left(\frac{\eta_L}{\sum_{k\epsilon N}\eta_k}\right) \tag{3.2}$$

$$A_{L_T} \leftarrow (A_L - A_{L_T,min}) \tag{3.3}$$

Variable	Meanings of variable in equation 3.2 and 3.3
A_L	Attractiveness value of location L
τ_L	Quantity of attractive levels of pheromone at location L
η_L	Quantity of repulsive levels of pheromone at location L
$\frac{\tau_L}{\sum_{k\in N}\tau_k}$	Weight of attractive levels of pheromones at location L
$\frac{\eta_L}{\sum_{k\epsilon N}\eta_k}$	Weight of repulsive levels of pheromones at location L
A_{L_T}	Adjusted (translated) attractiveness value of location L

The attractiveness values we get for each of the locations around an ant agent are further scaled so that they add up to 1 (merely to comply with probabilistic selection schemes). This is achieved by dividing each attractiveness value A_{L_T} by the sum of all the attractiveness values found. Let the scaled attractiveness value be denoted as A_{L_s}. Therefore, the scaled attractiveness value of location L is expressed as: $A_{L_s} \leftarrow \frac{A_{L_T}}{\Sigma_{i\in N} A_{L_T}}$.

Associating each location with an adjusted and scaled attractiveness value creates a probabilistic scheme in which a roulette wheel selection scheme arises (Jadaan et al., 2008). This is a stochastic selection scheme in which competing outcomes are allocated roulette intervals based on their chances of being selected (Jadaan et al., 2008). In this case, we would have eight intervals, each corresponding to each of the eight possible destinations. The width of each interval would correspond to the relative size of the scaled attractiveness value of the possible destination location.

Figure 3.6 shows an example of a roulette wheel setup in which each arc length represents the scaled attractiveness value of the corresponding location.

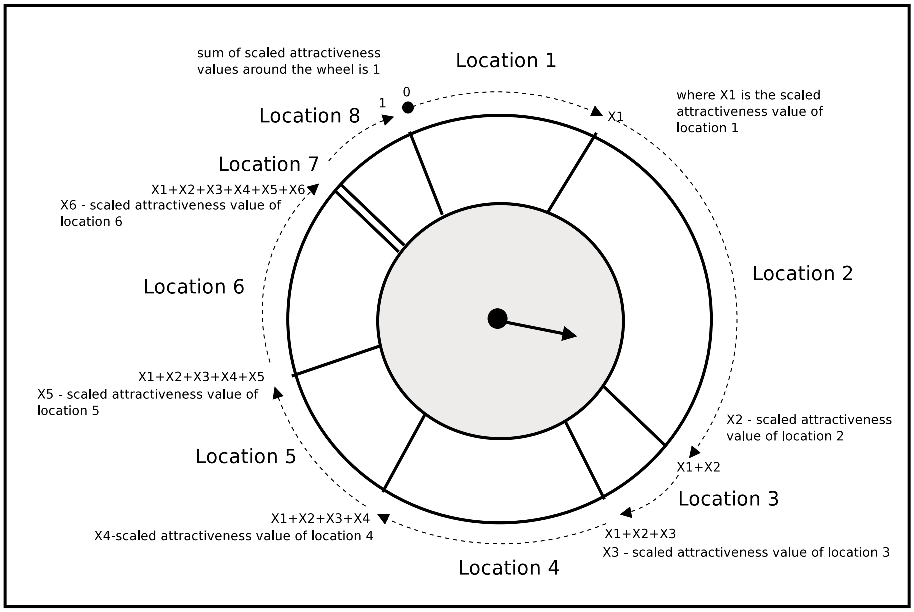


Figure 3.6: An example of a roulette wheel selection scheme

In this setup, location 2 is the most attractive destination because its arc length is longest.

Orientation is completed when a stigmergic ant agent "spins" the roulette wheel pointer in order to randomly pick an interval to follow. Spinning the roulette pointer is a stochastic process where the arrow would always stop pointing towards a randomly picked interval. That location whose corresponding roulette interval is pointed to by the roulette pointer, is the ant agent's direction of choice. In computational terms, "spinning" the roulette wheel refers to generating a float random number ρ whose value is in the range $(0, 1]$. Although this mechanism is fairly random, highly attractive locations are likely selected more often because their roulette intervals are wider. These intervals would get even wider with time in simulation because every time corresponding locations are visited, more and more levels of pheromones are added.

Algorithm 3.2 Stigmergic orientation

```
/* DEFINITION OF PARAMETERS

    ω_τ ← weight of attractive levels
    ω_η ← weight of repulsive levels
    A_L ← attractiveness value of location L
    A_Ls ← scaled attractiveness value
    ρ ← random number between 0 and 1
    d_i ← direction to follow
    L ←ant's current location

*/
```

$(MvH : \tau_i, \tau_i, \eta_i, w_\tau, w_\tau, w_\eta)$

{

foreach location L around ant agent i

{

$\omega_\tau \leftarrow \frac{\tau_L}{\sum_{j \in N} \tau_j}$

$\omega_\eta \leftarrow \frac{\eta_L}{\sum_{j \in N} \eta_j}$

$A_L \leftarrow ((\omega_\tau - \omega_\eta) - min(A_i))$

$A_{L_s} \leftarrow \frac{A_L}{\sum_{j \in N} A_j}$

}

Generate roulette wheel intervals

ρ ←pick a random number in $(0; 1]$

$\vec{d_i} \leftarrow L$ whose interval contains ρ

}

Lemma 2- *stigmergic ant agents' ability to orientate based on the concentration of the levels around is a discrete autonomous action that is undertaken at individual levels* - hence a primitive behaviour in ant systems.

- Algorithm 3.2 interpret these orientation processes in computational terms. Let $(MvH : p_1, p_2, p_3, w_1, w_2, w_3)$ represent the mnemonic of this primitive behaviour. This is an acronym for *Move to Highly attractive location*. The algorithm is an innovation we make from pre-tests and observation of the simulation over time. In this case, p_i are IDs of different levels of pheromone an ant agent can perceive and use during orientation. On the other hand,w_i are the weights that are associated with using each level of pheromone.

- For example, $(MvH : 1, 1, 2, 0, 1, -1)$ is read as: *Consider movements towards locations with higher levels of pheromone* 1 *while at the same time penalizing locations with higher levels of pheromone 2 given that the levels of pheromone* 1 *are weighted by* 1 *while the levels of pheromone 2 are weighted by* -1.

3.2.5.2 Message passing ant agent orientation

We indicated in the introduction of this chapter that the processes through which our message passing ant agent orientate are derived from related theories, and that most of the formula we propose are innovative. However the idea of sharing vector components with which to perform independent calculations for orientation is not new (Ngo et al., 2005). As such, the main activity of every message passing ant agent is referencing neighbour ant agents' historic experiences (Rodriguez et al., 2007.; Bayazit et al., 2002.; Bayazit et al., 2005.; Nasipuri and Li., 2002) and use the gathered information to calculate vectors for orientation purposes. That way, they expand their awareness of the environment in which they reside.

These ant agents are deployed in the default seek mode for the same purpose as stigmergic ant agents (Panait and Luke, 2004b). Upon deployment, we make an assumption that message passing ant agents pick direction vectors to follow at random. However they would assign the least confidence weights possible to these vectors. The hope is that these ant agents would work

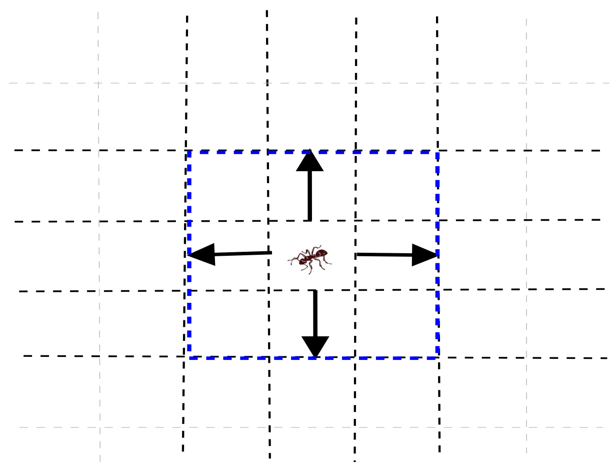


Figure 3.7: An ant agent's local coordinate system

towards improving their confidence weights based on the reward they achieve with time in simulation.

We indicated earlier on that message passing ant agents can self-localize relative to their neighbours. Each ant agent creates a local coordinate system in which the ant agent is placed at the origin. This coordinate system spans over three grid cells of the environment in each axis. Figure 3.7 illustrates this view. As a result, message blocks are only shared between the ant agent at the origin and those that are within accepted offsets within the defined local coordinate system.

To cut down on computing time, an ant agent in this category only references sets of attractive vectors at the time, as well as the related vector weights. The notion is that it is not necessary to explore directional history towards the opposite direction of the journey. However the main challenge the ant agent faces at this point is to decide on which of those attractive vectors would be best to follow. Mechanisms are required with which a new vector is selected which would fairly represent the ant agent's own perception of the

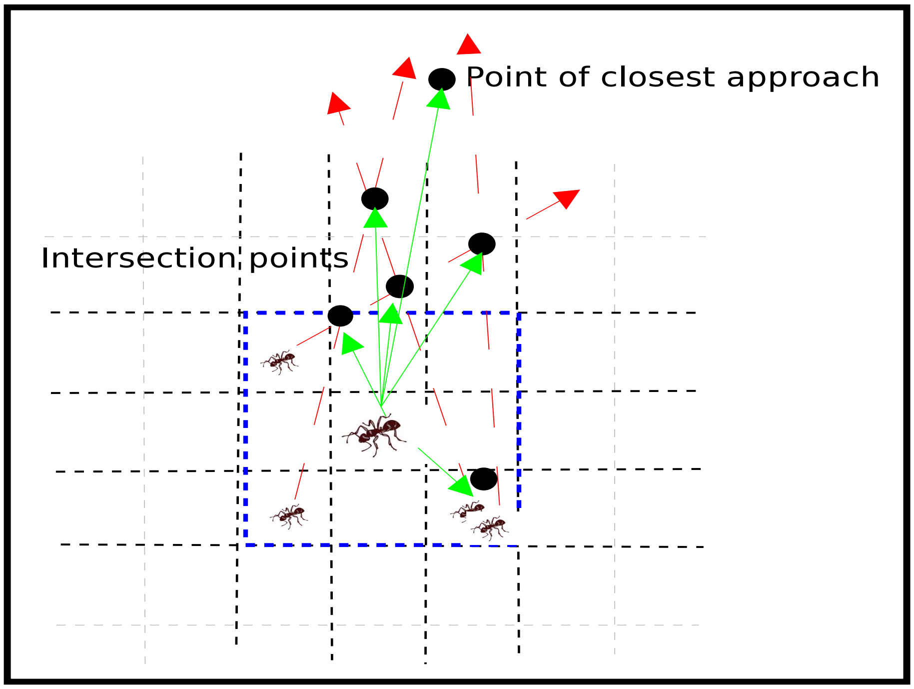


Figure 3.8: Intersection points and points of closest approach

direction to follow, as well as represent the views of its neighbours. That way a learning framework would arises (Lien et al., 2005).

To determine that fair direction vector to follow next, our message passing ant agent performs a number of computations. First, it calculates and creates a set of intersection points and points of closest approaches between all possible pairs of the vectors that are taken from its neighbours. The notion is that, two vectors which represent the knowledge of two independent ant agents in the neighbourhood would possibly intersect at a point which is most likely at the target. That vector which originates from the origin of the local coordinate system to that intersection point between a pair of vectors is a strong candidate direction vector for the orientating ant agent to follow.

In the event of a pair of vectors not intersecting within the defined environment span, we determine the point of closest approach between the two vectors, and record the vector that points to that point of closest approach as the candidate direction the orientating ant agent can follow. Figure 3.8 illustrates this framework using an example of a scenario in which a message passing ant agent has four neighbours. Two of these are situated at the same location. However each of the four neighbouring ant agents is following a unique direction vector (indicated by dashed arrows in red). The intersection points and points of closest approach between pairs of vectors in this scenario are marked using circular discs in black. The set we require records the x and y coordinates of each of the points marked by these circular discs. However, how do we find the actual values of x and y where these intersection points are, or where the points of closest approach are located? These are purely geometric tasks which we address in the next two section.

3.2.5.3 Intersection points between vectors

The problem of finding the coordinates of a point where two vectors intersect in a 2D plane is geometric (Xu et al., 2008). In our particular scenario, let the vector that is being followed by the i^{th} neighbouring ant agent around an orientating ant agent be denoted as $\vec{d_i}$. Therefore the vector that is being followed by the the j^{th} neighbouring ant agent is denoted as $\vec{d_j}$. The task we resolve in this section is to find the coordinates of a point at which the vector $\vec{d_i}$ intersects with the vector $\vec{d_j}$, if ever they do.

Suppose the offsets of the i^{th} and j^{th} neighbouring ant agents relative to the origin where the orientating ant agent is situated are $(x_i; y_i)$ and $(x_j; y_j)$ respectively. It is possible to determine another point along each line segment that is represented by $\vec{d_i}$ and $\vec{d_j}$. We can find these points geometrically using the expressions: $\vec{d_i} + s \times \hat{\vec{d_i}}$ and $\vec{d_j} + t \times \hat{\vec{d_j}}$ respectively. In these expressions, $\hat{\vec{d_i}}$ and $\hat{\vec{d_j}}$ are unit vectors of $\vec{d_i}$ and $\vec{d_j}$ respectively. The parameters s and

t are the magnitudes or relative weights of these two vectors. Suppose the coordinates of the points defined by the two expressions: $\vec{d_i} + s \times \hat{\vec{d_i}}$ and $\vec{d_j} + t \times \hat{\vec{d_j}}$ are $(x_{i+1}; y_{i+1})$ and $(x_{j+1}; y_{j+1})$ respectively. With two points

$$x = \frac{\begin{vmatrix} \begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix} & \begin{vmatrix} x_i & 1 \\ x_{i+1} & 1 \end{vmatrix} \\ \begin{vmatrix} x_j & y_j \\ x_{j+1} & y_{j+1} \end{vmatrix} & \begin{vmatrix} x_j & 1 \\ x_{j+1} & 1 \end{vmatrix} \end{vmatrix}}{\begin{vmatrix} \begin{vmatrix} x_i & 1 \\ x_{i+1} & 1 \end{vmatrix} & \begin{vmatrix} y_i & 1 \\ y_{i+1} & 1 \end{vmatrix} \\ \begin{vmatrix} x_j & 1 \\ x_{j+1} & 1 \end{vmatrix} & \begin{vmatrix} y_j & 1 \\ y_{j+1} & 1 \end{vmatrix} \end{vmatrix}} \tag{3.4}$$

$$y = \frac{\begin{vmatrix} \begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix} & \begin{vmatrix} y_i & 1 \\ y_{i+1} & 1 \end{vmatrix} \\ \begin{vmatrix} x_j & y_j \\ x_{j+1} & y_{j+1} \end{vmatrix} & \begin{vmatrix} y_j & 1 \\ y_{j+1} & 1 \end{vmatrix} \end{vmatrix}}{\begin{vmatrix} \begin{vmatrix} x_i & 1 \\ x_{i+1} & 1 \end{vmatrix} & \begin{vmatrix} y_i & 1 \\ y_{i+1} & 1 \end{vmatrix} \\ \begin{vmatrix} x_j & 1 \\ x_{j+1} & 1 \end{vmatrix} & \begin{vmatrix} y_j & 1 \\ y_{j+1} & 1 \end{vmatrix} \end{vmatrix}} \tag{3.5}$$

along each neighbouring ant agent's direction vector, we can determine that point at which the two vectors $\vec{d_i}$ and $\vec{d_j}$ intersect (Line Intersection, 2014),

Algorithm 3.3 The closest point of approach method

let $w = L_i - L_j$

$= (d_i + s \times \hat{\vec{d_i}}) - (d_j + t \times \hat{\vec{d_j}})$

Therefore:

$w \times \hat{\vec{d_i}} = (d_i + s \times \hat{\vec{d_i}} - d_j - t \times \hat{\vec{d_j}}) \times \hat{\vec{d_i}} = 0$ (1)

$w \times \hat{\vec{d_j}} = (d_i + s \times \hat{\vec{d_i}} - d_j - t \times \hat{\vec{d_j}}) \times \hat{\vec{d_j}} = 0$ (2)

From equation (1), after expanding the brackets:

$d_i \times \hat{\vec{d_i}} + s \times \hat{\vec{d_i}} \times \hat{\vec{d_i}} - d_j \times \hat{\vec{d_i}} - t \times \hat{\vec{d_j}} \times \hat{\vec{d_i}} = 0$

$= s \times \hat{\vec{d_i}} \times \hat{\vec{d_i}} - t \times \hat{\vec{d_j}} \times \hat{\vec{d_i}} = \hat{\vec{d_i}} \times (d_j - d_i)$ (3)

From equation (2), after expanding the brackets:

$d_i \times \hat{\vec{d_j}} + s \times \hat{\vec{d_i}} \times \hat{\vec{d_j}} - d_j \times \hat{\vec{d_j}} - t \times \hat{\vec{d_j}} \times \hat{\vec{d_i}} = 0$

$= s \times \hat{\vec{d_i}} \times \hat{\vec{d_j}} - t \times \hat{\vec{d_j}} \times \hat{\vec{d_j}} = \hat{\vec{d_j}} \times (d_j - d_i)$ (4)

Let: $\hat{\vec{d_i}} \times \hat{\vec{d_i}} = a$

$\hat{\vec{d_i}} \times \hat{\vec{d_j}} = b$,

$\hat{\vec{d_j}} \times \hat{\vec{d_j}} = c$,

$\hat{\vec{d_i}} \times (d_j - d_i) = d$,

and $\hat{\vec{d_j}} \times (d_j - d_i) = e$

Substituting in equations (3) and (4), t and s are:

$t = \frac{ae-bd}{ac-b^2}$

$s = \frac{be-cd}{ac-b^2}$

if ever they do. Equations (3.4) and (3.5) show how the x and y coordinates of the required intersection point are calculated. These equations calculate the matrix determinants using the coordinates of the points found.

3.2.5.4 Points of closest approach between vectors

In cases where two vectors $\vec{d_i}$ and $\vec{d_j}$ do not intersect within the defined environment, a point of closest approach between these two vectors is determined and recorded as the orientating ant agent's candidate direction to follow. Let the geometric equations of the line segments along the vectors $\vec{d_i}$ and $\vec{d_j}$ be defined as $L_i = \vec{d_i} + s \times \hat{\vec{d_i}}$ and $L_j = \vec{d_j} + t \times \hat{\vec{d_j}}$ respectively. The point at which these two vectors have a minimum offset $w = L_i - L_j$ is the point of closest approach we require. This point occurs when w is perpendicular to both L_i and L_j. This is the same point where $w \times \hat{\vec{d_i}} = 0$ and $w \times \hat{\vec{d_j}} = 0$. We show in algorithm 3.3 how the two equations ($w \times \hat{\vec{d_i}} = 0$ and $w \times \hat{\vec{d_j}} = 0$) are solved for s and t, and how we arrive at the points P_s and P_t along the line segments L_i and L_j where w is smallest. This algorithm is derived from vector geometry (Sunday, 2007).

Note that the denominator $ac - b^2$ in algorithm 3.3 is always non-negative because $ac - b^2 = |\hat{\vec{d_i}}|^2|\hat{\vec{d_j}}|^2 - (|\hat{\vec{d_i}}||\hat{\vec{d_j}}|cos\theta)^2 = (|\hat{\vec{d_i}}||\hat{\vec{d_j}}|sin\theta)^2 \geqslant 0$ (Sunday, 2007). When $ac - b^2 = 0$, the two vectors $\vec{d_i}$ and $\vec{d_j}$ are parallel to each other. We handle cases where two vectors are parallel by fixing the value of one parameter, and use either equations to solve for the other. In the end, that parallelism is completely absorbed. For the purpose of the message passing ant agents we propose, the direction we consider when $\vec{d_i}$ is parallel to $\vec{d_j}$ is that of the midpoint along the line segment w.

3.2.5.5 The choice of a direction vector

Once a set of $(x; y)$ coordinates of the intersection points and points of closest approaches for all possible pairs of the direction vectors that are taken from neighbours is in place, the ant agent's next challenge is to make a choice of which way to follow based on the information gathered so far. The vector sought must represent the general consensus of all neighbouring ant agents. We determine this vector using least squares point estimation (Francis, 1990). This approach is applied on the set of coordinates of the intersection points and points of closest approaches.

Two least squares regression lines are derived from the set of intersection points and points of closest approaches, one for y on x, and another for x on y. The regression line for y on x is represented as $y = a_1x + b_1$, and that for x on y is represented as $x = a_2y + b_2$ (Francis, 1990). Thus, equations (3.6) and (3.7) calculate the values of a_1 and b_1 in the regression equation for y on x, while equations (3.8) and (3.9) determine the values of a_2 and b_2 in the regression equation for x on y. In both cases, m indicates the density of ant agents around the orientating ant agent, from which the set of intersection points and points of closest approach are obtained. Thus, $\sum y$ sums up the y coordinates, and $\sum x$ adds the x coordinates of the intersection points and points of closest approach in the set. Therefore $\sum xy$ sums up the products of corresponding x and y coordinates of every intersection point or point of closest approach in the set.

$$\sum y = a_1 \sum x + mb_1 \tag{3.6}$$
$$\sum xy = a_1 \sum x^2 + b_1 \sum x \tag{3.7}$$

$$\sum x = a_2 \sum y + mb_2 \tag{3.8}$$
$$\sum xy = a_2 \sum y^2 + b_2 \sum y \tag{3.9}$$

$$a_1 = \frac{m \sum xy - \sum x \sum y}{m \sum x^2 - \left(\sum x\right)^2} \tag{3.10}$$

$$b_1 = \frac{\sum y \sum x^2 - \sum x \sum xy}{m \sum x^2 - (\sum x)^2} \tag{3.11}$$

Algorithm 3.4 Message passing orientation

```
/* DEFINITION OF PARAMETERS

    L_i; L_j; L_{i2}; L_{j2} ← points on vectors
    d⃗_i; d⃗_j ← direction vectors
    s_i; s_j ← weights/magnitudes
    x; y ← intersection points
    P_{s_j}; P_{s_i} ←points of closest approach
    < Set >← vector of directions
    L_k ←new direction to follow   */

(MsP : v_c, v_j, v_j)
{foreach agent k at origin

    {foreach agent j around k
        {  L_j ← (x_j; y_j; z_j)  :  d⃗_j  :  s_j
           L_{j2} ← d⃗_j + s_j × d̂⃗_j
           forevery other agent i around k
           {  L_i ← (x_i; y_i; z_i)  :  d⃗_i  :  s_i
              L_{i2} ← d⃗_i + s_i × d̂⃗_i
              if  (d⃗_j ∩ d⃗_i)
                  {x←from equation (3.4)
                  y←from equation (3.5)
              }
              else
              { x←P_{s_j}
                y←P_{s_i}
              }
              vector <Set >+= {(x,y,z)}
           }
        }
        L_k ← (x̄;ȳ;z̄)  in vector <Set >
        c ←stdev (s)  :  ∀_s  in agents j
        w_k(t+1) = 1/2 ( w_k(t) + (Σ_{i∈k} w_i(t))/k ∗ c ) + λ
    }
}
```

Equations (3.10) and (3.11) show how the first set of simultaneous equations is simplified for the values of a_1 and b_1. Flipping the values of x and y in these equations gives the formula to solve for a_2 and b_2 in the second set of simultaneous equations. Least squares point estimator finds that point at which the two regression lines intersect. Francis (1990) shows that this point coincides with the centre of mass of all the points that are recorded in the set of intersection points and points of closest approach, $(\bar{x}; \bar{y})$. That vector which originates from the origin towards the centre of mass $(\bar{x}; \bar{y})$ is the path the orientating ant agent must follow in the next step.

3.2.5.6 Message passing put together

Algorithm 3.4 summarizes the message passing and orientation processes. These processes are viewed as an autonomous primitive activity of every ant agent in this category - hence a primitive behaviour. Note that this primitive behaviour comprises of three key components namely; using shared direction vectors to create sets of intersection points and points of closest approach,determining the least squares point estimator which fairly represents the new direction to follow, and updating vector weights. We gave the details of how vector weights are updated in section 3.2.4.4.

Lemma 3 - *the ability of a message passing ant agent to orientate and update its knowledge regarding the direction vectors to follow, is a discrete autonomous action that is undertaken at individual levels* - hence a primitive behaviour in ant systems. - Let the mnemonic for this algorithm be $(MsP : v_c, v_j, v_j)$, which is an acronym for $Message\,Pass$. The first parameter of this primitive indicates the region in the agent memory where the vector weight parameter should be read. The second parameter indicates the memory block where the required vector should be taken from. Then the last parameter indicates the message block in which the results of the calculations are stored.

- For example; $(MsP : 0,0,1)$ tells an ant agent *to retrieve confidence weights that are stored in the ant agent's memory components number* 0*, as well as retrieve vector components that are stored in memory component* 0*, and accumulate these vectors and weights in order to get the resultant vector which would be stored in the orientating ant agent's memory component number* 1.

3.2.5.7 Ant agent movements

Upon successful orientation, an ant agent must relocates to the selected location. Movement is a critical ability in ant systems with which swarm level tasks are accomplished (e.g. locating food-like resources) (Chibaya and Bangay, 2007.; Panait and Luke, 2004a, 2004b.; Solimanpur et al., 2005). In line with the literature, we made an assumption that our ant agents remain in motion throughout the simulation - which is an autonomous activity that is undertaken at individual levels.

Lemma 4 - *ant agents' abilities to move around the environment in response direct or indirect interactions are discrete autonomous actions that are undertaken at individual levels* - hence a primitive behaviour in ant systems.

- Algorithm 3.5 interprets ant agent movement policies after successful orientation. This algorithm is inspired by the works presented in Panait and Luke (2004a, 2004b). Three parameters are required which indicate the offset of the preferred destination. These offset coordinates are acquired when the ant agent orientates. Let $(MvP : x, y, z)$ represent the mnemonics for $Move\_to\_a\_Preferred\ L$.

Algorithm 3.5 Ant agent movements

```
/* DEFINITION OF PARAMETERS

    d⃗_i(t) ← position or vector at time t
    x; y; z ← coordinates of current position
    d⃗_i(t+1) ←new position at time t+1
    x_i; y_i; z_i ← offset of new position
    */

(MvP : x_i, y_i, z_i)
{

    foreach agent i
    {
        d⃗_i(t)←(x; y; z)
        d⃗_i(t+1) ←(x + x_i ;y + y_i ;z + z_i)
    }

}
```

- In this context, the ant agent simply adds the offset values to the coordinates of its current position. For example; $(MvP : 1, -1, 0)$ tells an ant agent to relocate to the location that is south-east of its current position (one cell in the positive direction of x and one cell in the negative direction of y. The z component remains at 0 because we operate in 2D).

3.2.5.8 Stigmergic information update: Drop pheromone

Communication between stigmergic ant agents is environment mediated, and pheromone chemicals are the key ingredient for ant agent orientation (Dorigo et al., 1999.; Montgomery et al., 2007.; Nakamichi and Arita, 2004, 2005).

Algorithm 3.6 Drop pheromone

```
/* DEFINITION OF PARAMETERS

    p_i ← levels of pheromone to drop
    Q_L(t) ← quantity of p_i at L at time t
    Q_L(t+1) ← updated quantity of p_i
    q ← amount of pheromone to drop

*/
(Drp : p_i, q)
{

    foreach agent i at L
    {
        Q_L(t) ← quantity of p_i at L at time t
        Q_L(t+1) ← Q_L(t) + q
    }

}
```

These pheromone chemicals are placed on the environment by the ant agents (Dorigo et al., 1999.; Panait and Luke, 2004b.; Chibaya and Bangay, 2007). In line with the norm, stigmergic ant agents can update the levels of specific pheromones on their current location in each step.

Lemma 5 - *the ability of a stigmergic ant agent to update the levels of pheromones that are held at its current location is a discrete autonomous action that is undertaken at individual levels* - hence a primitive behaviour in ant systems.

- Algorithm 3.6 summarizes the semantics through which stigmergic ant agents retrieve and update these specific levels of pheromone in a movement step. This algorithm is also inspired by the works of Panait and Luke (2004a, 2004b). The algorithm requires two parameters, one which indicates the ID of the levels of pheromones to be detected and updated (p_i), and another which indicates the amount of pheromones the ant agent can place on the environment at the time (q).

- Let the mnemonic of this primitive activity be denoted as $(Drp : p_i, q)$, an acronym for $Drop\, specific\, levels\, of\, pheromone$. Precisely, a particular level of pheromone is stored in a cell tuple with two fields (ID and Quantity). For example; $(Drp : 1, 1)$ tells a stigmergic ant agent to retrieve the levels of pheromones whose ID is 1 that are held at location L at time t and top them up by 1 unit of the same levels. In this work, stigmergic ant agents can place a fixed amount of the same levels of pheromones. Variation of the amount of pheromone an ant agent can place on the environment in each step is not a subject of study in this work.

3.2.5.9 Stigmergic information update: Pheromone evaporation

In stigmergic systems, global update rules may be added which handle pheromone dissipation processes (Chirico, 2004.; Schoonderwoerd et al., 1996). This refers to pheromone updates through evaporation and diffusion. Pheromone evaporation in particular, refers to a process whereby a certain percentage of the levels of pheromone that are held on each location of the environment is lost without any ant agents or user intervention. It is, in fact, a fault tolerance control in the stigmergic model with which sub-optimal trails are cleared off from the environment (Dorigo et al., 1999).

Lemma 6 - *a pheromone update rule which triggers pheromone evaporation is a fault tolerance control which occurs at low levels* - hence a primitive behaviour in ant systems.

Algorithm 3.7 Pheromone evaporation

```
/* DEFINITION OF PARAMETERS

    p_i(t) ← levels of pheromone at time t
    α ← evaporation rate

*/
(Evp : α)
{
    foreach location L
        foreach pheromone p_i at L
            p_i(t) ← (1 − α) × p_L(t)
}
```

- Algorithm 3.7 describes the semantics which characterize pheromone evaporation. These semantics are built on the works of (Dorigo et al., 1999) and (Panait and Luke, 2004a, 2004b). Let the mnemonic for this primitive activity be denoted as $(Evp :)$. The algorithm requires one parameter (α), which indicates the evaporation rate supported at the time. All levels of pheromones that are on the environment are simultaneously dissipated. In technical terms, the algorithm retrieves the quantities of each level of pheromone at a particular location. A percentage of the retrieved quantity is taken off, and the remaining quantity overwrites the original quantity of the same levels of pheromone at that location.

3.2.5.10 Stigmergic information update: Pheromone diffusion

Stigmergic information is also updated through pheromone diffusion (Panait and Luke, 2004b.; Chibaya and Bangay, 2007). This is a process whereby a certain percentage of the levels of pheromone that are held on one

Algorithm 3.8 Pheromone diffusion

```
/* DEFINITION OF PARAMETERS

    Qj(t) ←quantity of pi at J at time t
    α ← diffusion rate

*/
(Dfs : α)
{

    foreach location L
      foreach location j around L
        foreach pheromone pi at  L
          if (Qj(t) < QL(t) and Qj(t) > 0)
            Qj(t) ← Qj(t) + α × QL(t)

}
```

location of the environment spills over to other locations without ant agents or user intervention. Precisely, pheromone diffusion integrates dynamics into ant agents' low level behaviour (Rajbhupinder et al., 2010), smoothing and widening the trails that are formed with time in simulation (Chibaya and Bangay, 2007).

Lemma 7 - *a pheromone update rule which triggers pheromone diffusion is a fault tolerance control which occurs at low levels* - hence a primitive behaviour in ant systems.

- Although pheromone dissipation is practically an environment manipulation control, it is a useful parameter of emergence worth exploring. Algorithm 3.8 describes the semantics which characterize pheromone diffusion. These semantics are also built on the works of (Dorigo et al., 1999) and (Panait and Luke, 2004a, 2004b). This algorithm, which we denote in mnemonics as (Dfs :), requires one parameter as well, which indicates the diffusion rate supported at the time.

- We make an assumption that pheromone diffusion is only possible when the levels of pheromone on the source location are higher than the same levels of pheromone on the receiving location. In addition, the receiving location must contain a minimum threshold amount of the same levels of pheromone before dissipation updates are allowed. Such controls prevent environment saturation which often occurs when pheromone diffusion controls are executed unconditionally (Panait and Luke, 2004b).

3.2.5.11 Stigmergic information update: A combined update rule

This section summarizes stigmergic information update. Precisely, Equation (3.12) shows how we put together the effects of the levels of pheromone that are placed on a location by ant agents, the levels that are acquired through diffusion, as well as the levels that are left on a location after evaporation. This equation connotes that the updated levels of pheromone at a particular location L at time $t+1$ are a result of adding the levels of pheromone that are acquired from neighbouring locations through diffusion ($\sum_{1}^{\forall_j(L)>L} \alpha \times Q_j(t)$), and the total amount of the same levels of pheromone that are placed on the same location by ant agents that visited the location at time t, which is ($\sum_{1}^{\forall_{Agent}(L)} q$). The quantity that is left at the same location L after pheromone evaporation, $(1-\alpha) \times Q_L(t)$, is added to the sum, giving $Q_L(t+1)$ in equation (3.12).

$$Q_L(t+1) = (1-\alpha) \times Q_L(t) + \sum_{1}^{\forall_{Agent}(L)} q + \sum_{1}^{\forall_j(L)>L} \alpha \times Q_j(t) \qquad (3.12)$$

Variable	Meaning of variable in equation (3.12)
α	dissipation rate
$Q_L(t)$	Quantity of pheromone at location L at time t
q	Quantity of pheromone that an ant agent can drop
$(1-\alpha) \times Q_L(t)$	Remaining quantity of pheromone after evaporation
$\sum_1^{\forall_{Agent}(L)} q$	Total quantity of pheromone dropped by ant agents at time t
$\sum_1^{\forall_j(L)>L} \alpha \times Q_j(t)$	Total quantity of pheromone diffused from neighbour locations

3.2.5.12 Message passing update: Normalizing vectors

Message passing orientation involves lots of calculations in which the resultant vectors thereof have magnitudes that are variable. These magnitudes can be smaller, equal, or greater than 1 depending on the direction vectors that are summed up. Assuming that these magnitudes translate to the confidence measures of the ant agents, then a contradiction arises where these magnitudes differ from the confidence weights that are stored in ant agents' memories. To avoid this possible mix up, we normalize the resultant vectors and restore the confidence factors. Practically, normalizing the resultant vector standardizes an ant agent's step size throughout the simulation.

Lemma 8 - *message passing ant agents' ability to normalize vectors is a discrete and autonomous action at individual levels* - hence a primitive behaviour in ant systems.

- Algorithm 3.9 presents the semantics for normalizing vectors in the message passing category. This algorithm accepts three parameters relating to the components of the resultant vector thereof. First, the algorithm computes the length of the vector, which is always anon-zero value when the orientating ant agent has neighbours around. A normalized

Algorithm 3.9 Normalize vector

```
/* DEFINITION OF PARAMETERS

    v_x, v_y, v_z ←vector components
    ρ ← a randomly generated vector component
    w_v(t) ←confidence weight of vector v at time t

*/
(Nrm : v_x, v_y, v_z)
{
    v_length← v_x^2 + v_y^2 + v_z^2
    if (v_length ≠ 0)
    {
        v_x ← v_x / v_length
        v_y ← v_y / v_length
        v_z ← v_z / v_length
    }
    else
    {
        v_x ← ρ (random x component)
        v_y ← ρ (random y component)
        v_z ← 0.0
        w_v(t) ← 0.0
        (Nrm : v_x, v_y, v_z)
    }
}
```

vector is found by dividing each component of the resultant vector by the length of the vector. However isolated ant agents would yield resultant vectors whose magnitudes are 0 because they do not have neighbours. In this case, our ant agents rather assume random direction vectors and follow them with the least confidence weight possible. These randomly picked vectors are normalized before they are used.

- Let the primitive routine with which vectors are normalized be denoted as $(Nrm :)$, an acronym for $Normalize$. For example $(Nrm : 0, 0, 1)$ tells an ant agent to: *normalize a vector whose components are stored in memory block* 0 *and store the normalized vector's components in memory block* 0*, while keeping possible random choices in the first memory block.*

3.2.5.13 Target detection

It is critical that ant agents possess abilities to detect the targets and any other objects of the environment (Cavalcanti et al., 2006a). In line with the work of Naeem et al. (2007), the positions of key objects, including targets, are marked using specific pheromone-like chemical indicators. These chemical indicators are neither produced by the ant agents nor by objects in the environment. They neither change in quantity, nor dissipate. Instead, they are set as constants when the environments are created and initialized.

While stigmergic ant agents are designed with implicit abilities to detect and interpret chemicals, message passing ant agents can only understand vectors. A mechanism is therefore required with which message passing ant agents can, at least, detect target indicators and interpret them.

Our message passing ant agents are therefore particularly designed with an extra agent level ability to detect and convert target indicators to vector information. This is not the first time vectors have been used to interpret the meaning of some levels of pheromone on the environment (Payton et al., 2001). In this case, the length of the vector thereof is non-zero if the chemical indicators are detected, connoting arrival around the vicinity of the target. However the same length will remain zero when the ant agent has not found the target sought. Whatever outcome, the confidence weight is updated accordingly (dropping or raising depending on vector length).

Algorithm 3.10 Detect target indicators

```
/* DEFINITION OF PARAMETERS

    Q_L ←quantity of target indicators at L
    x ←minimum levels of p_i required
    d_i←ant agent's current vector
    w_i(t)←confidence weight of agent i

*/
(PtV : p_i, x)
{
    for-each location L around agent i
    {
        if (Q_L > x)
        {
            d_i←(L_x;L_y;L_z)
            w_i(t) = 1.0
        }
    }
}
```

Lemma 9 - *message passing ant agents' ability to detect specific target indicators and convert these to corresponding vector information is an ant level ability* - hence a primitive behaviour in ant systems.

- Algorithm 3.10 presents the semantics of the routine with which message passing ant agents detect and convert target indicators to vector information. Two parameters are key in this algorithm, one which indicates the ID of the target indicator the message passing ant agent must detect, and another one which sets the minimum levels of the same target indicator that the ant agent must detect in order to trigger behavioural changes.

- We call this algorithm in mnemonics as $(PtV :)$, which is an acronym for $detect\_and\_convert\_Pheromone\ to\ Vector$. This algorithm captures the notion that an ant agent i that is situated at location L_i, where L_i is adjacent to location L_j, can detect the levels of a particular target indicator p_i at L_j provided the levels of p_i exceed a threshold quantity x. If this is the case, the ant agent overwrites the vector it is following, $\vec{d_i}$, and assumes the vector which points to location L_j where the target is likely placed. The ant agent's confidence weight is changed immediately, indicating that the ant agent has arrived at its target.

- For example, $(PtV : 3, 1)$ tells a message passing ant agent to *detect and convert target indicators whose ID is* 3 *to vector information, provided the levels of the same target indicators around the ant agent are above the threshold quantity of* 1.

3.2.5.14 No action

The design of the XSets we propose is such that a fixed number of instructions are required in each ant agent state. In the event of an ant agent requiring less instructions in one internal state, a No_Action instruction is used as a filler primitive behaviour. The mnemonic $(NOp :)$ tells an ant agent to do nothing. In departure from the norm, multiple inclusion of this behaviour in an XSet does not count as redundant. In computational terms, this is an algorithm with no code (see Algorithm 3.11). Although this completes (fills) the sets, ant agents would ignore these behaviours and jump to the next different behaviour in the sequence.

Lemma 10: *ant agents' ability to do nothing is a low level and autonomous skill* - and hence a primitive behaviour.

Algorithm 3.11 No action

```
/* DEFINITION OF PARAMETERS

    Algorithm without code

*/
(NOp :)
{

    //No action

}
```

3.3 Conclusion of the chapter

This chapter addresses two aspects of this thesis. First, it established the meta information and parameters which set forth the simulator. These are basically user defined. The system architecture is presented, along with the assumptions we make, as well as a discussion of the components and parameters of the simulation simulation system. Key design issues were also presented. The second aspect we address is the identification of primitive behaviours which characterize the activities of ant agents at individual levels, and presented routines which interpret these primitive behaviours in computational terms. It is important to note at this stage that these primitive behaviours are still claims (lemmas - claims whose purposes are help in proving a theorem) that will be verified and validated in Chapters 4, 5, and 6 regarding their usefulness to the XSet approach we propose. We make the following conclusions regarding these two aspects:

1. The key meta information defines agent design and environment setup parameters. First, our system requires us to make an explicit choice of

the type of ant agents we want to use at the time between stigmergic, message passing, and hybrid. It requires us to state the agent density upfront, agent memory, the number of internal states an ant agent supports, as well as the number of pheromone chemicals the same ant agents can perceive and interpret. It is also important to state the size of the environment we want to use, the dimensions in which we operate, the simulation time we are allowed to score the performances of particular XSets, as well as the number of instructions that are allowed in each internal state. Most importantly, we need to state the evaluation environment which describes the form of emergent behaviour we require. This information particularly spells out the design of the ant agent, the properties of the XSets sought, as well as the properties of the environment.

2. Key to both types of ant agents are primitive behaviours with which ant agents achieve *orientation* (i.e. $(MvH : \tau_i, \tau_i, \eta_i, w_\tau, w_\tau, w_\eta)$ and $(MsP : v_c, v_j, l_j)$). These instructions tell an ant agent to pick an appropriate direction to follow in the next step. A primitive behaviour for agent *movement* is also critical ($(MvP : x_i, y_i, z_i)$). This instruction relocates an ant agent towards the chosen direction. Ant agents in general, require abilities to *update* key system information. This is achieved in three ways in the stigmergic category ($(Drp : p_i, q)$, $(Evp : \alpha)$, and $(Dfs : \alpha)$). These instructions create and maintain the shared memory for the swarm. In the message passing context, vectors and confidence factors are concurrently updated when the $(MsP : v_c, v_j, l_j)$ instruction is executed. Most of the direction vectors that are yield as a result of sharing message blocks must be normalized ($(Nrm : v_x, v_y, v_z)$). In addition, all ant agents must update their internal states whenever it becomes necessary ($(StS : m, n, x)$). However message passing ant agents require an extra ability to detect and interpret chemical indicators ($(PtV : p_i, x)$). We can use a filler instruction when an agent is

required to do nothing $((NOp:))$.

3. The key outcomes of this chapter are the two findings that are stated in points 1 and 2 above. We can explicitly state the set of primitive behaviours - U as a collection of ten primitive behaviours. In line with our assumptions, it is possible to extract subsets of U in which all the primitive behaviours are inspired by stigmergic or message passing processes. However it is also possible to come up with mixed subsets of primitive behaviours - hence the three categories of ant type. The set U is the key input to the next chapter where we generate the power set and the XSets thereof. Figure 3.9 shows the composition of set U.

$$\boxed{U = \{\text{(NOp:),(MvH:),(Drp:),(MsP:),(MvP:),(Evp:),(Dfs:),(Nrm:),(PtV:),(StS:)}\}}$$

Figure 3.9: The set of primitive behaviours - U

A number of contributions emanate from this chapter, both to the thesis and to the board of knowledge. Some of these contributions are:

1. Ant agent control routines have always been "black box" processes with regard to explaining how the primitive behaviours of ant agents are implemented. This chapter presents ant agents' primitive behaviours, along with the computational routines which characterize the semantics of these primitive behaviours. This way we create relevant knowledge in the field.

2. The mechanisms in which ant agent orientation is achieved, both in the stigmergic and message passing ant agent systems, are innovative. In particular, the mathematical models we proposed for determining the attractiveness of locations around a stigmergic ant agent, and the geometric calculations we proposed for combining vectors and determine the desired target, are both of our own making. Besides creating

new knowledge in the field, these mathematical models and ant agent orientation processes may inspire the development of new ant systems with practical benefits.

3. The concept of orientation shows potentials for possible extension for use in scenarios where geometry is not available, for example, routing in graphs. As such, we create new avenues of research in the field.

4. Stigmergic ant agents reinforce the knowledge of the swarm by placing more and more levels of pheromone on the environment in each step. However the mechanisms in which message passing ant agents update each other's knowledge are innovative. Precisely, the knowledge of message passing swarms is held in the memories of swarm members in the form of vectors and levels of confidence. Again, the thesis creates knowledge in this respect, knowledge with potentials to attract more useful researches in the field.

5. The algorithms which interpret the semantics of different ant agent activities, along with the parameter values they take, form useful dictionaries from which ant agent systems can be programmed for different configurations in the future.

6. The mechanism in which the confidence weights of message passing ant agents are updated is creative. This may inspire the development of formal principles for updating similar parameters.

The next chapter validates the primitive behaviours we identified in this chapter, as well as investigating a methodology for searching for best combination of primitive behaviours and meta information which would create XSets of primitive behaviours that can allow emergent behaviour to be manifest in swarms of ant-like devices.

Chapter 4

Creation and Evaluation of XSets

4.1 Introduction

Figure 1.1 sub-divided the research problem of this thesis into five sub-problems, namely: (a) the identification of primitive behaviours, (b) investigation of strategies for combining primitive behaviours into XSets and genetically evolve these XSets into useful dictionaries for achieving predictable emergent behaviour, (c) evaluation of XSets using particular measures of emergence, (d) validation of primitive behaviours, XSets, and the measures of emergence that arise from using particular XSets, and (e) application of validated XSets to different problem domains (the reader is also referred to section 1.2 for a detailed presentation of each of these five sub-problems).

Chapter 2 went on to review works in which each one of these sub-problems is placed in the literature and board of knowledge. In Chapter 3, we addressed the first sub-problem of the thesis (identification of discrete candidate primitive behaviours of ant agents in the stigmergic and message passing categories), and proposed the design and implementation of ten such primitive behaviours in computational terms. In each design decision we made, a lemma was proposed whose purpose has been to set up a stepping stone on the path to proving the concept of XSets and the search for optimal solutions in this respect.

Although Chapter 3 made claims regarding the identification, design, and implementation of discrete candidate primitive behaviours of ant agents in the two categories we study, it neither verified nor justified the choices we made as appropriate, suitable, and sufficient constructs for defining a language for programming ant agents towards predictable emergent behaviour - explaining why we stated these candidate primitive behaviours as lemmas.

Creation of potentially useful XSets from the proposed candidate primitive behaviours, evaluation of these XSets for emergent properties, and the evolution of even better XSets using genetic programming principles, are the subjects of investigation in this Chapter. Our goal is to search for best performer XSets for predictable emergent behaviour (evolutionary algorithm search for optimal solutions). Although we may have insights into the structure of the XSets we want, it is difficult from a human's point of view to predict appropriate combination of the candidate primitive behaviours. It is hard to tell the suitable parameter values of these primitive behaviours. Worse still, we can not foretell appropriate sequences in which these primitive behaviours must be arranged in XSets by a naked eye.

In the light of these challenges, we propose a genetic programming system which is inspired by the work of Koza (1992). This genetic programming system forms the basis for our search for appropriate combination of primitive behaviours, search for proper sequences of primitive behaviours, and search for appropriate parameter values for the primitive behaviours that would define a language for programming ant agents towards deliberate engineering of particular forms of emergent behaviour. We believe that successful discovery of such XSets may provide useful insights and a baseline upon which researches aimed at assembling deterministic emergent products may arise.

4.1.1 A brief overview of genetic algorithms

Genetic algorithms are a searching tool which works well in search spaces where very little is known about the solution sought (Murphy, 2003). They use the principles of selection and evolution to produce several potential solutions (a genetic population) to a given problem and evaluate these solutions in order to identify the closest solution to the problem (Koza, 1992). Thus, a genetically generated solution may not be the exact solution to the problem but can be very close to the exact solution.

A genetic algorithm begins by creating an *initial population* of candidate solutions. Most works in the literature randomly generate the initial population of solutions (Koza, 1992). The algorithm then creates a sequence of new generations of populations through genetic evolution (Koza, 1992). In genetic terms, each member in the genetic population is referred to as an *individual*. In our case, every XSet which represents a possible solution to the problem at hand is an individual. The processes through which individuals are manipulated in order to create other *individuals* is called *genetic operations*. At each step, the algorithm uses the individuals that are in the current generation and population to create new individuals and the next population.

In creating a new population, the algorithm first ranks the individuals that are in the current population by computing some form of fitness values. In this work, we refer to this fitness value as the XSet's *index of merit* (see section 4.4.7 for details regarding how this value is calculated). The algorithm then selects individuals, called *parents*, biased by their fitness levels. Parent individuals are used to produce children individuals either by making random changes to their composition —*mutation*—or by combining the components of two parents—*crossover*. However there are always a few individuals that are directly promoted to the next population as elite members in order to enhance population diversity as we go down the generations (Murphy, 2003).

Note that the population size remains constant when new individuals and new populations are created, meaning that for each new individual, an old one is removed from the population. Generally, successful implementation and application of genetic algorithms depends on the following three criteria:

- It must be possible to evaluate the fitness of each individual relative to other potential solutions that are in the population (Murphy, 2003). Our work presents mechanisms in which we can quantify the amount of emergence that arise as a result of using a particular XSet (see section 4.4 for this aspect of the chapter).
- It must be possible to break a potential solution into discrete parts that can vary independently. The description of the structure of XSets allows us to explicitly reference each composite primitive behaviour and its parameters and vary these components independently (see section 4.2.1 for details regarding the structure of XSets).
- It must be possible to get a relatively “better” solution even if that solution is not the absolute best solution.

The research problem of this chapter focuses on searching for best performer XSets that would describe a language for programming ant-like devices towards predictable emergent behaviour. The definition of genetic algorithms which we presented in this section offers a suitable searching tool for these desired XSets.

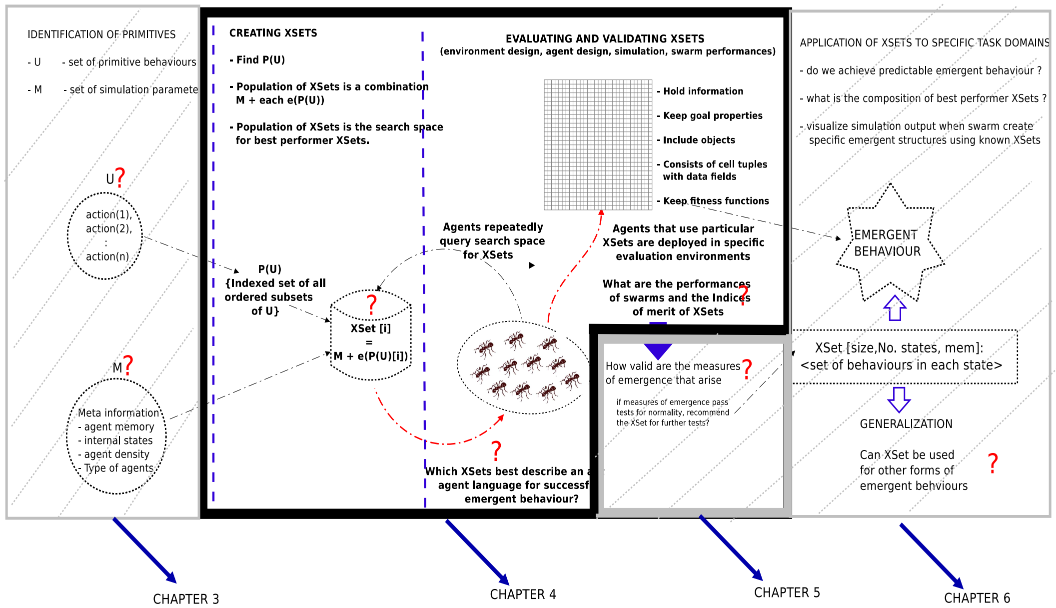


Figure 4.1: Creation and evaluation of XSets

4.1.2 Problem statement

Figure 4.1 reviews the research problem of this thesis and zooms into the focus of this chapter. First, it shows the two key outputs of Chapter 3 (set of candidate primitive behaviour U, and a collection of user defined meta information M). These two pieces of information are the main inputs to Chapter 4, with which we address the second (investigation of strategies for combining primitive behaviours into XSets and genetically evolve these XSets into useful dictionaries for achieving predictable emergent behaviour) and third (evaluating XSets) sub-problems of this thesis (see sections 1.2.2 and 1.2.3 for more details regarding these two sub-problems). We can therefore re-phrase the key aspects of this chapter into four separate tasks as follows:

1. **Creating XSets** - In addressing this first task, we are guided by the question which says: *how do we create XSets?* First, we present an

analogy from biology which motivates the design decisions we make regarding the structure of XSets. Thereafter, we describe the structure of an XSet in terms of the two pieces of information that are input from Chapter 3 (set of candidate primitive behaviours U, and collections of user defined meta information M). Most importantly, we show how the set of primitive behaviours U is manipulated in order to generate revised versions of the same set U depending on the *antType* and *cardinality* parameters in M (see section 3.2.3 for details regarding the parameters that are captured as meta information). Power sets are then described in terms of the revised U. These power sets consist of the possible ordered subsets of the revised version of U (see section 4.2.2 for an illustration of the concept of indexed and order subsets). Each element in the power set is a unique subset of U, a partial permutation, and has a unique index in the power set. In computational terms, our power set is a vector data structure (see section 4.2.4 for details regarding the computational representation of power sets). Merging each element of the power set with specific meta information (*antType, cardinality, number of internal states, and agent memory*) creates the initial population of XSets on which the genetic algorithm operates until optimum solutions are evolved.

2. **The search space and genetic population** - The second task of this chapter is to describe a mechanism in which the search space for best performer XSets is evolved. We describe genetic operations with which better and better XSets are added into the population. We identify mechanisms in which we can quantify the extent to which emergent behaviour is manifest as a result of using particular XSets, and motivate for five measures of emergence that are all linked to related works. These measures of emergence are scaled within the range $[0; 1]$ before they are averaged in order to determine each XSet's index of merit (see section 4.4 for details regarding these measures of emergence).

3. **Definition of evaluation environments** - The third task of this chapter is concerned with the definition and configuration of the platform on which ant agents that use particular XSets would reside during simulation (evaluation environments). We mainly show how the meta information that is provided by the user in the *XSets generator* sub-system (see section 3.2.3 for details regarding the parameters that are captured in this sub-system) influence the design of the environment on which we operate. Most importantly, we explain how the high level properties of the emergent behaviour sought are incorporated into the simulation system, mainly paying attention to the purposes of different environment parameters.

4. **Evaluation of XSets** - The last task of this chapter is to experimentally search (from the search space) for those XSets that best describe control rules for creating specific emergent behaviour. We particularly investigate the properties of XSets on a well known case study metaphor of path finding ant agents in swarms. In conducting this experiment, we firstly present the path finding problem and contextualize it to our ant agent metaphor and environment setup. Swarms of ant agents are deployed using particular XSets and their performances are scored. Each XSet is therefore ranked according to the index of merit it achieves.

The key outcome of this chapter are explicitly stated XSets which best describe ordered collections of primitive behaviours and parameters with which a particular class of emergent behaviour is guaranteed. We emphasize, again, that the choice to evaluate XSets on the case study scenario of path finding swarms is only to prove the concept and functionality of XSets. It is not the path finding outcome that matters, but rather the discovery of those XSets that give rise to the outcome.

4.1.3 Overview of the chapter

The breakdown of the content and sections of this chapter are outlined as follows:

- Section 4.2 comes first, describing five key aspects of this chapter. First, it emphasizes on explaining the structure of an XSets before we dwell on its creation and evolution (see section 4.2.1 for this aspect of the study). The design of how the initial genetic population is created follows thereafter (see section 4.2.2 for this aspect of the study). In section 4.2.3, we describe how parent XSets are selected and evolved in order to create children XSets as we move down the generations. Examples are given in this respect in order to clarify the genetic operations thereof. In addition, we explicitly show the representation of XSets in the genetic populations in section 4.2.4. The last sub-section of this section describes how ant agents use XSets (see section 4.2.5). In summary, this section responds to the first task of this chapter (see section 4.1.2 for the list of tasks of this chapter).

- Section 4.3 follows thereafter, explaining the design of evaluation environments based on the meta information and parameter values that are provided by the user at run-time (see section 3.2.3for this aspect of the chapter). First, we describe the global environment parameters in section 4.3.1. Section 4.3.2 then presents swarm level parameters. The last sub-section of this section presents targets related parameters (see section 4.3.3). In each case, details regarding how swarm information is kept on the environment, as well as how information is accessed when it is required are emphasized on. In summary, this section responds to the second task of this chapter (see section 4.1.2 for the list of tasks of this chapter).

- Section 4.4 goes on to describe particular measures of emergence with which we evaluate XSets for allowing emergent behaviour to occur. In doing so, we indicate how each measure of emergence is assessed, starting with the speed of emergence (see section 4.4.1), quality of emergence (see section 4.4.2), average delivery rate (see section 4.4.3), average end-to-end delays (see section 4.4.4), and then Shannon's information value (see section 4.4.5 for these measures of emergence). Key in some of these measures of emergence is the requirement to extract samples of ant agents to track and assess. This section also describes the sampling technique we use to extract significant samples (see section 4.4.6). The last part of this section illustrates how the index of merit (which tells the extent to which an XSet is useful) is calculated from the five measures of emergence we propose (see section 4.4.7 for this aspect of the study). In summary, this section responds to the third task of this chapter (see section 4.1.2 for the list of tasks of this chapter).

- Section 4.5 follows, in which we mainly administer an experiment for evolving and evaluating XSets for abilities to solve a specific tasks - in this case, the path finding problem. The measures of emergence that are prescribed in section 4.4 are used to rank all the XSets in the search space. To experimentally achieve this ranking, we firstly present the framework of the path finding problem (see section 4.5.1 for this framework). The experiment design in which we evaluate all XSets for allowing path finding behaviour is presented next (see section 4.5.2 for this aspect of this chapter). Then, the results thereof, which report the performances and configuration of best performer XSets in each category, as well as their relative indices of merit, close this section (see sub section 4.5.3 for these results). Precisely, this section responds to the fourth and last task of this chapter.

◦ Conclusions close the chapter in section 4.6, highlighting the key observations we make, as well as the contributions of the chapter to the board of knowledge and the thesis.

4.2 Generation of XSets

The design and envisioned functionality of the XSets we propose is inspired by an analogy in cell biology, where genes are described as “molecular units with particular roles in the cellular life of living organisms” (Singh et al., 2012). Organized collections of genes form chromosomes, and strands of chromosomes form DNA structures which encode the search space for organisms' abilities, behaviours, characteristics, and any other features.

Identifying the genes that form particular chromosomes in specific species, as well as describing the order in which these genes are configured in the chromosomes are active areas of research in medicine and genetics (Singh et al., 2012). The notion is that, if we get to understand gene configuration and gene sequences in chromosomes, then issues related to genetic disorder can be tackled and hopefully rectified with easy. The results of such researches are of direct benefits to any life, including human life.

This work is motivated by these biological relationships between genes and chromosomes. Discrete primitive behaviours (those that were identified in Chapter 3) are viewed as ”genes” - units with particular roles in the life and behaviours of ant agents. Organized collections of primitive behaviours (XSets) are created with the view of ”chromosomes” in mind - collections of genes in specific sequences. The key phrase in this analogy is “*organized collections in specific sequences*”, which connotes the requirement for order (similar to the requirement for order of genes in chromosomes (Singh et al., 2012)).

The genetic programming system we propose manipulates the primitive behaviours that form XSets (mutating these primitive behaviours, mutating the parameter values, crossing over primitive behaviours, reproducing primitive behaviours, or crossing over parameter values). Mutation, crossover, and promotion rates are set at 15%, 80%, and 5% respectively (see section 4.2.3 for details). We refer to collections of XSets as a *genetic population of XSets*. In this thesis, the size of a genetic population of XSets is sufficiently large in order to allow diversity in the search space (500 XSets at a time). Such a population hopefully forms the search space for ant agent abilities, behaviours, and any other features (similar to the roles of DNA in living organisms). Each XSet in the genetic population is allowed $10,000$ iterations as scoring time before it is ranked. Practically, the genetic population of XSets undergo 10 evolutions before best XSets are recommended, implying that the maximum evolution limit is $500 \times 10,000 \times 10$ (50,000,000 iterations).

We can therefore re-state the focus of this chapter in terms of this analogy as follows: we investigate the primitive behaviours (genes) which form XSets (chromosomes) which best describe a language for programming ant agent behaviours (DNA) - the same way biologists require solutions regarding the genes that form particular chromosomes in the DNA. To address this problem, we firstly describe the structure of an XSet in details.

4.2.1 The structure of an XSet

The key inputs with which an XSet is created are the two main outputs of Chapter 3, namely:

1. the set of discrete candidate primitive behaviours (set of genes) $U = \{(NOp :), (MvH :), (Drp :), (MsP :), (MvP :), (Evp :), (Dfs :), (Nrm :), (PtV :), (StS :)\}$

2. the set of user defined meta information $M = \{$*AntType, Cardinality, InternalStates, AntMemory, EnvironmentSize, EnvironmentDimensions, EvaluationEnvironment, ScoringTime, AgentDensity, Pheromones, GAPopulationSize, GASelectionMethod, GASelectionPressure, EliteGeneRate, CrossoverRate, MutationRate, EvolutionLimit*$\}$.

$$\boxed{\textbf{antType[}i\textbf{,}s\textbf{,}n\textbf{]} < (a_{0,1}), (a_{0,2})..(a_{0,i})|(a_{1,1})..(a_{1,i})|(a_{2,1})..(a_{2,i})|..|(a_{s,1})..(a_{s,i}) >}$$

Figure 4.2: Template structure of an XSet

In the views of this work, an XSet (an individual in the genetic population) consists of three components namely: (a) *antType*, (b) meta information (*cardinality, internalStates, antMemory*) and, (c) the list of primitive behaviours and parameter values which characterize ant agent behaviours over time (*selected elements of set U and their parameters*). For clarity in the explanations which follow, Figure 4.2 shows the template structure of an XSet. Note that i, s, and n are agent level meta information (maximum cardinality, number of internal states, and agent memory respectively). Also note that $(a_{p,q})$ represents a discrete primitive behaviour that is included on the list of collections of "genes" of the XSets.

We indicated that we support three categories of *antType,* namely; stigmergic (whose alias in this thesis is *stigXSet*), message passing (whose alias is *msgXSet*), and hybrid *antType* (whose alias is *hybXSet*). Stigmergic XSets contain lists of primitive behaviours that are only inspired by pheromone sensitive ant agent processes (such as *drop pheromone*, *evaporate pheromone, diffuse pheromone*, etc). On the other hand, message passing XSets consist of primitive behaviours that are only based on vector geometry (such as *normalize*, *message pass vectors*, etc). The last lot (hybrid XSets) consists of XSets whose primitive behaviours are taken from both the stigmergic and the message passing category. The choice of which category of the XSets (*antType*) one requires at the time is decided upon by the user in the *XSets generator* sub-system (see section 3.2.3 for details regarding the selection of these parameters).

Once the *antType* is chosen (*stigXSet, msgXSet, or hybXSet*), the next component in the structure of XSets is a list of *ant level meta information.* This information is also provided by the user. First, we state the highest cardinality the XSets can support. This (cardinality) is an integer value which indicates the maximum number of primitive behaviours an ant agent can execute in each internal state in each step. XSets with smaller cardinality which should desired properties are better since that would define simpler and naive ant agents - which would be in line with our dictum to develop simple autonomous and naive ant-like devices. Let the variable i represent the user-selected maximum cardinality of XSets.

Thereafter, the number of internal states each ant agent can support is required in the definition of XSets. Section 3.2.4.2 described the structure of ant agent internal states. We also indicated that each internal state has an integer ID. Let the parameter s indicate the number of internal states an ant agent can support. Therefore the IDs of the internal states supported would range between 0 and $(s-1)$ inclusively.

The last ant agent level meta information required indicates the amount of memory blocks an ant agent can carry at a time. This is a non-zero value since each ant agent must, at least, hold internal state information. The structure of our ant agents' memories was described in details in section 3.3. Let n indicate the number of blocks an ant agent can support at the time. These three ant agent level meta information are separated from the *antType* and the list of primitive behaviours using square brackets *(e.g. antType* $[i, s, n]$—.

The third component in the structure of XSets is the list of primitive behaviours and their parameter values. This list is separated from ant agent meta information by pointy brackets $< >$. A single primitive behaviour within these pointy brackets, along with its parameter values, are enclosed in parentheses *(e.g. antType [i, s, n] $<$ (StS : m,p,x) $>$).* Each primitive behaviour is separated from its parameter values by a colon sign. The parameter values of a primitive behaviour are separated by commas. Different

hybXSet [2,2,4] $<(MsP : 0, 0, 0), (StS : 1, 0, 1)|(Drp : 2, 1), (Nrm : 0, 0 : 0)>$

Figure 4.3: Hybrid XSet: 2 states: 2 instructions per state: 4 memory blocks

primitive behaviours are also separated by commas (e.g. *antType [i, s, n]* $<$ *(StS : m,p,x) , (MvP : x,y,z)* $>$). Lists of primitive behaviours that are executed in different internal states are separated by a vertical bar $|$.

Figure 4.3 illustrates an example of a hybrid XSet. This example can be used by ant agents which support two internal states, requiring two primitive behaviours in each internal state, and carrying at most four memory blocks at a time.

4.2.2 Initial population of XSets

Genetic programming systems begin by creating an *initial genetic population* from which genetic evolution is based. Most works in the literature randomly generate this initial genetic population of solutions to a problem (Koza, 1992). The same genetic algorithms then create sequences of new genetic populations (where children individuals are evolved and introduced in the new population) until a set evolution limit is reached. This section

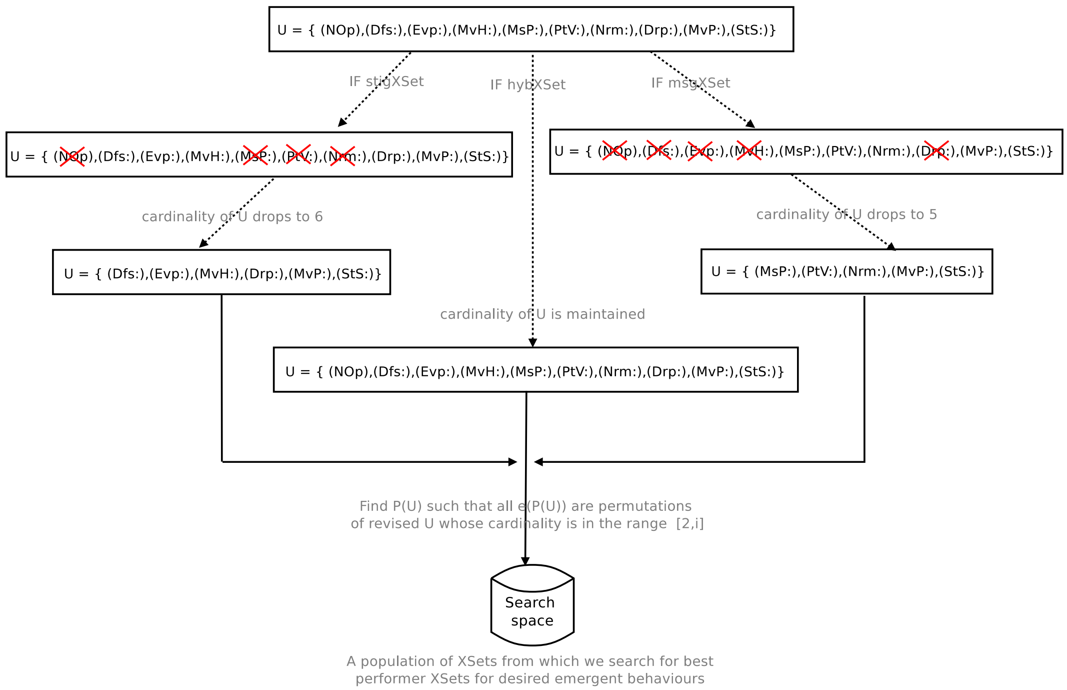


Figure 4.4: Effects of selecting ant type

discusses the processes through which we create the initial genetic population of XSets from which the rest of the genetic operations are based.

In the structure of XSets (see section 4.2.1), the first two components are meta information that are provided by the user when the *XSet generator* sub-system is invoked (*antType* and the parameters [i, s, n]). However the list of primitive behaviours that follow thereafter, their parameter values, and the sequences in which these primitive behaviours are arranged are system generated.

The choice of an *antType,* which the user makes, fires a rule on the set of candidate primitive behaviours U. This rule discards all the primitive behaviours that are outside the scope of the *antType* choice made and produce a revised set of candidate primitive behaviours (one where invalid primitive behaviours, at the time, are omitted). Figure 4.4 illustrates this view.

We defined cardinality as the maximum number of primitive behaviours that are supported by ant agents in each internal state at the time. The choice of this parameter (by the user) also fires a rule on the set of candidate primitive behaviours U, to allow the use of subsets of U whose cardinality are greater than 1 but less than or equal to the selected highest cardinality. We are saying that the XSets allowed at the time must comprise of collections of primitive behaviours in selected groups of 2 to i primitive behaviours - where i is the user-captured highest cardinality (see Figure 4.2).

Our work determines a set of partial permutations (combination with no repetitions) between the available candidate primitive behaviours. In this context, partial permutations reinforce the notion of "*ordered collections*" of primitive behaviours (where sets a, b and b, a are considered different). Partial permutations are also considered in order to enhance diversity in the potential solutions that would form the initial genetic population.

For illustration purposes, if a cardinality of say, $i = 4$, is selected by the user under a *stigXSet* category. Selection of *antType* ($stigXSet$) remains with 6 stigmergic primitive behaviours in U. XSets of cardinality 2, 3, and 4 are determined where $\frac{6!}{(6-2)!} = 30$ XSets will have two primitive behaviours in each ant agent's internal state, $\frac{6!}{(6-3)!} = 120$ XSets will have three primitive behaviours in each ant agent's internal state, and $\frac{6!}{(6-4)!} = 360$ XSets will have four primitive behaviours in each ant agent's internal state. A set of 510 XSets arises from which the initial genetic population can be picked at random.

Figure 4.1 illustrates the effects of firing rules on U based on the selected *antType*, as well as the effects of firing rules on U based on the selected cardinality. In this case, identification of partial permutations of U, and storage of the permutations thereof is handled by the *XSets generator* subsystem.

Seven other genetic related meta information are decided upon by the user as well. The first of these is the *genetic population size.* This is a constant number of XSets that can be in the genetic population at a time. Our system sets a default population size of 500 XSets in case the user decides not to change this parameter. These 500 XSets are randomly picked from the set of potential solutions (partial permutations) that are computed as shown in the previous paragraph.

The parameter values of the primitive behaviours are randomly set. However these random value are picked from within valid ranges that are pre-defined in the system. The hope is that these initial parameter values would undergo genetic mutation as we move across generations until the evolution limit is reached. We discuss the rest of the genetic related meta information in the next section where we describe how XSets are selected to assume parental roles, and how children XSets and new generations of XSets arise.

4.2.3 Evolution of new XSets

The initial genetic population of XSets is the input to the genetic system. The key activity in the genetic system is to evolve children XSets using the XSets that are currently in the genetic population as parents. The user has a choice between two parent *selection methods* (tournament selection, or roulette wheel selection (Jaadan et al., 2008)).

A tournament selection algorithm randomly picks a group of XSets from the existing genetic population (Koza, 1992). The picked group of XSets is put in a "tournament" in order to assess each tournament member's index of merit for a particular purpose, in this case, assessing the path finding behaviour.

A winner XSet in each tournament is assigned a parental role. The number of XSets that can compete in the same tournament at a time is called the *selection pressure* (another user defined parameter). A large selection pressure enhances the quality of the children XSets that arise because better fit parent XSets are likely picked and would win the tournaments (Bai et al., 2008). However this would create strange biases towards populations of highly fit XSets which lack diversity. We emphasize on the use of small selection pressures in order to give weaker XSets a chance to breed their traits into the new populations as well. These tournaments are repeated for as many times as the number of parent XSets required at the time.

Users may also decided to use a roulette wheel selection algorithm (Jaadan et al., 2008) when they select parent XSets. This approach arranges all XSets in the current genetic population in a sequence of interval, where the width of each interval is associated with a particular XSet's index of merit (Koza, 1992). To pick a parent XSet, a random interval selector is spinned. That XSet whose corresponding interval is hit by the random interval selector is assigned a parental role regardless of its relative fitness levels. This selection method has the advantage that a choice is made from the entire population of XSets rather than from a randomly selected group (Koza, 1992). However there is a danger that an XSet may serve as a parent in the same generation for more than once. Other XSets may never serve as parents even when they are highly fit, thus loosing the genetic traits they possess as we move down the generations.

In both cases, the selected parent XSets undergo genetic *crossover* or *mutation* in order to give rise to children XSets. Crossover refers to the combination or mixing of the components of two parent XSets (Koza, 1992). Figure 4.5 illustrates point crossover operation on two parent XSets. Although the crossover rate can be changed by the user at run-time, our illustrations assume a default 80% crossover rate. This is a common crossover rate that has been proposed in the literature (Bai et al., 2008.; Koza, 1992).

On the other hand, mutation refers to making random changes to the composition of a particular parent XSet (Koza, 1992). Figure 4.6 illustrates an example of mutation operation on a parent XSet.

The children XSets that arise replace old members in the current genetic population, thus defining a new and better population. Some XSets are

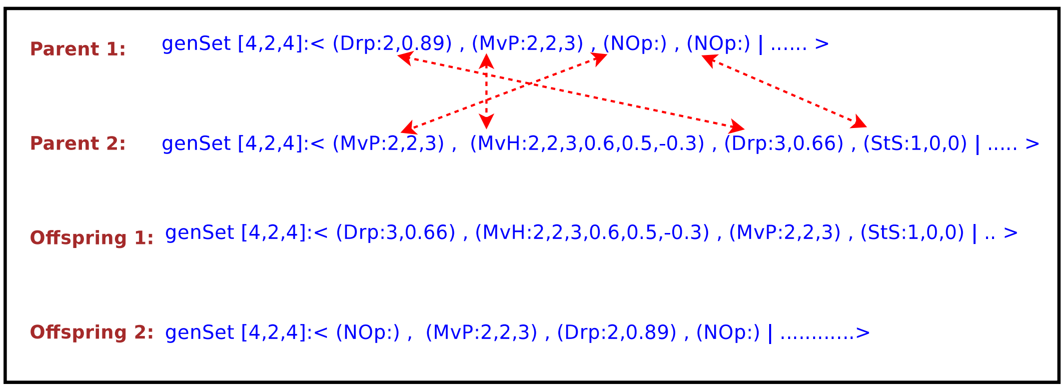


Figure 4.5: An example of crossover operation

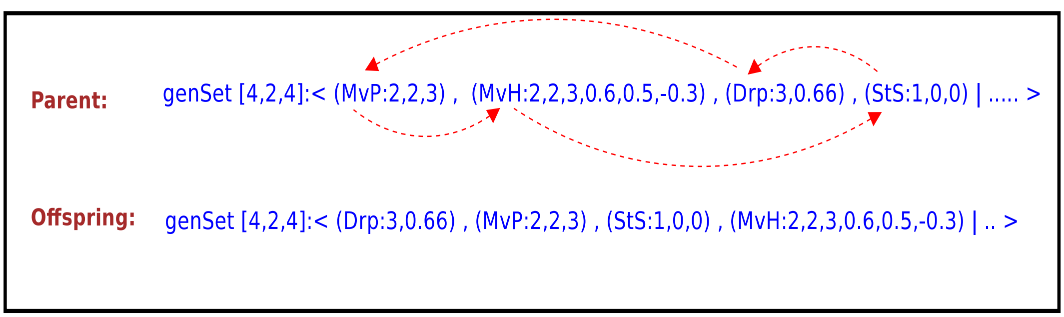


Figure 4.6: An example of mutation operation

picked as elite and transferred to the *new* generation. A 5% promotion rate is common in the literature (Bai et al., 2008.; Koza, 1992). Promoting

XSets ensures diversity in the new generation. However there is a danger of pushing weak XSets through - hence the low promotion rate we propose. Note that the parameter values of primitive behaviours are also mutated, crossed over, and promoted across generations. The last parameter is the evolution limit which sets the time the genetic system is allowed to evolve different generations of XSets before the best solution is picked.

4.2.4 Representation of XSets in the genetic population

Given the structure of XSets and how XSets are generated and evolved, this section briefly discusses how the resulting XSets are stored and represented in the genetic population. In this work, we store the XSets in a vector data structure where each node of the vector has three data fields, namely:

index - this is an integer field in which we store the ID of the XSet. This ID corresponds to the position of the XSet in the vector. Computationally all IDs are in the range $[0; populationSize - 1]$.

XSet - this field is of the type String. It records the full composition of the XSet in its mnemonic format. All primitive behaviours are stated together with their parameter values (e.g. msgXSet [5,4,8] : $<$ (PtV:0,1), (MvP:1,1,0), (MsP:0,0,1), (Nrm:0,0,0), (StS:1,0,0) $|$ (NOp:), (NOp:), (NOp:), (NOp:), (StS:2,1,0) $|$ (PtV:2,1), (MvP:1,-1,0), (MsP:2,2,3), (Nrm: 2,2,2), (StS:3,1,0) $|$ (NOp:), (NOp:), (NOp:), (NOp:), (StS:0,2,0)$>$).

indexOfMerit - this is a measure of how well the XSet achieved desired emergent formation. This is a float value which ranges between 0 and 1. We initialize this value to 0 until the XSet has been evaluated and ranked.

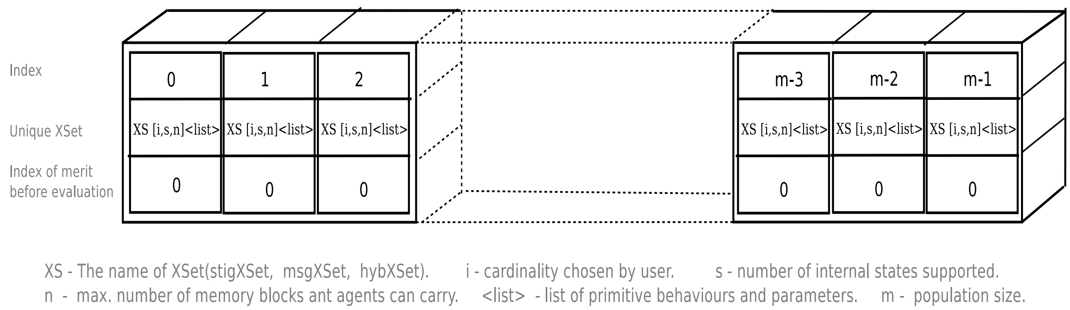


Figure 4.7: Representation of XSets in the search space

Figure 4.7 illustrates the representation of the data structure in which we keep XSets - the genetic population. As a result, we can sequentially access XSets within a selected range of indices, or directly pick a particular XSet using its index.

4.2.5 How ant agents use XSets

Upon deployment, all ant agents in a swarm use the same selected XSet at the time. Although this is the case, ant agents perceive the effects of the primitive behaviours in this XSet differently. For example, although two stigmergic ant agents may orientate using the same primitive behaviour, it is not guaranteed that they would spin the roulette wheel selector and get the same outcome. As a result, each ant agent would probabilistically head in its own perceived direction.

When using the information encoded in an XSet, ant agents read the list of primitive behaviours sequentially starting from left to right. For example, if an XSet is represented as follows: *hybXSet [2,2,4]:<(Drp:0,1) , (StS:1,1,0) | (MvP:1,-1,0), MvH:1,1,0,0.5,0,5,-1>*. A hybrid ant agent would first drop the levels of pheromone whose ID is 0 in unit quantities before considering to switch to internal state 1 if the levels of pheromone whose ID is 1 are above 0. In the next internal state, the same ant agent would first relocate to a cell which is south-east of the current cell before orientation (which would however characterize a random wandering swarm). This example also shows why order is a key issue in the study of the configuration of XSets.

The next section describes the environments on which we assess the performances of swarms of ant agents that use the XSets that are recorded in the genetic population at the time.

4.3 Definition of environments

The remaining tasks of this chapter are related to setting up the platform for evaluating the performances of swarms of ant agents when they use particular XSets that are recorded in the genetic population at the time. The assumption is that these swarms of ant agents would reside, and operate in particular evaluation environments where the targets, starting point, and any other objects thereof, are defined and set. In this work, the setup of evaluation environments is guided by the meta information that is provided by the user at run-time (when the *XSet generator* sub-system is invoked).

Figure 4.8 visualizes the internal design of an environment, showing at least the notion behind cells and cell tuples. Our environments are practically 2D arrays of N rows and N columns, where the intersection of a row and column form a cell. These cells are containers of data and building blocks of the shared memories for the swarms. Although the visualization we present in Figure 4.8 appears as $3D$ to an outsider, our definition of a $2D$ setup is that of an array of blocks of memory (blocks of containers) in rows and columns (without the height or matrix parameter) - hence still 2D.

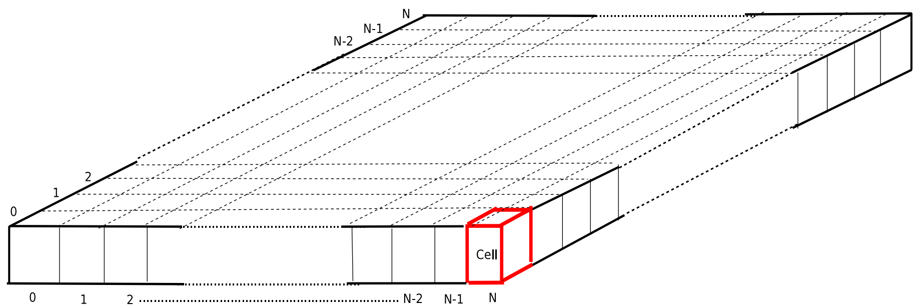


Figure 4.8: Evaluation environment

The information that is held in cells is what hints ant agents of what actions to take at a particular time and location. Similar environment setups have been proposed in the past (Negulescu et al., 2006.; Panait and Luke, 2004a.; Babaoglu et al., 2006.; Seevinck and Edmonds, 2008.; Mason, 2002) (see chapter 2, section 2.3.1 for details regarding different environment setups). We discuss the key environment parameters in the next sub-section.

4.3.1 Environment parameters

Successful launch of an environment requires the user to provide a number of optional parameters. This is done to allow system flexibility and adaptability to different task domains and scenarios. Key in their decisions, users must provide the number of cells of the environment in each direction (length), assuming square environments. However this value must not exceed the resolution of the screen in use. As a result, the values the users provide in this respect are validated by the central resource sub-system (which verifies that the architectures available are compatible with the parameter choices made). However, in the event of the user not decided on the size of the environment, our *XSet generator* assumes a default setup of 100×100 grid environments in 2D which we pre-tested as motivation to establish that the size does not detract from the behaviours exhibited.

The users are also free to decide on the number of internal states ant agents will support, as well as the number of the levels of pheromones each ant agent would be able to perceive. These two pieces of information guide the design of environment tuples. For example, the length of the XSet string can be predicted from knowing the cardinality and the number of internal states. That way appropriate field sizes are declared. Similarly, knowledge of the number of levels of pheromone an ant agent can perceive informs us of the likely structures of the records in which different pheromone details will be recorded on each cell. Therefore a cell is, in fact, a record with many fields, where particular information is stored about the activities completed at each location. Some typical examples of information that is stored on cells include the different levels of pheromones, target indicators, nest indicators, fitness parameters, and the IDs of ant agents that are at the cells at the time.

Most importantly, users decide on the forms of specific emergent behaviours they want the swarms to create. In our case, different forms of emergent behaviours are pre-coded separately, each with specific sets of fitness functions and constraints relative to the size of the environment. The choice of a particular form of emergent behaviour to simulate at the time incorporates the fitness functions and constraints on stipulated cells of the environment.

For example, suppose we decide to evaluate XSets for resolving the path finding problem. This choice, in particular, would include fitness functions which describe the parameters of a line between two points (the starting point and the target). This choice makes use of the coordinates of the targets and those of the starting point and perform geometric calculations to determine the cells of the environment through which a line that joints these two points would pass. Those cells would have in their tuples, an indicator of priority. However that indicator of priority does not, in any way, influence ant agents' decisions. Rather, it is used to compare how well an ant agent has performed against the preferred outcome.

It is also critical that the user sets the time frame in which swarms are allowed to score performances before the index of merit of the *particular* XSet they use is calculated. This time frame is stored as the "maximum possible age" of the environment. Time in simulation is measured in iterations (Hovda, 2008). This has the advantage of eliminating concerns regarding the measures we get being influenced by the processing power of the computer. As such, our simulation system allows ant agents to make as many steps as the maximum possible age of the environment before an index of merit for the XSet in use is determined. Longer times in simulation are desirable in order to allow swarms sufficient time to converge. However that may have a negative effect on the operating system's memory management schedules.

4.3.2 Swarm parameters

In addition, users decide on a preferred ant agent density. This parameter indicates the number of ant agents that are deployed into the environment at a time. Our ant agents can co-exist on the same cell. As a result, agent density is independent of environment size. Rather, when deciding on agent density, one has to consider the processing power of the PC in use. However in the event of the user not decided on this aspect, the default agent density is set to 5000, which we also pre-tested to establish that the density does not detract from the behaviours exhibited.

Information regarding the preferred agent density is relevant in the launching of environments since it informs the design of the tuples in which ant agent IDs are stored when they visit particular cells. In our case, ant agent IDs are merely indices in the array of agents. The format of the fields in which these ant agent IDs will be held must comply with the possible IDs that would arises based on the selected agent density.

4.3.3 Target parameters

We have designed our simulation system in such a way that the starting point is placed and hard-coded at a fixed position on the environment. This has the advantage of achieving fair experimental outcomes when we compare the performances of swarms that use different XSets. However, it does not matter where the starting point is positioned as long as that position is within the environment and the same position will be maintained throughout the experiments. Variation of the position of the starting point is not a subject of study in this work since environment complexity is not an agent level parameter of emergence (which is what we investigate), but rather a system level factor (which is not what we investigate).

On the contrary, placement of targets is based on the fitness functions that are pre-defined. In our case, the centre of the emergent structures sought must be located at the centre of the environment. Again, variation of the position of the centre of targets is not a subject of study for the same reason that environment complexity is not, at this point, an agent level parameter of interest.

Figure 4.1 shows that once the environment has been successfully launched, the next step is to deploy swarms of ant agents in specific agent densities, using particular XSets, and evaluate the usefulness of the selected XSets for achieving the emergent behaviour sought. Mechanisms are therefore required, with which we evaluate the performances of XSets and determine their indices of merits.

In the next section, we discuss five measures of emergence with which we can quantify the extent to which emergent behaviour is manifest in specific evaluation environments when swarms of ant agents are deployed using specific XSets as control dictionaries for particular emergent behaviours.

4.4 Measures of emergence

Our motivation for selecting the five particular measures of emergence we study is primarily based on their prevalence in the literature. In this context, a measure of emergence is a metric which indicates the extent to which an ant agent's behaviour is influenced by the behaviours of other members of the swarm or by some shared features in the environment as a result of following some known set of instructions (XSet).

Studying mechanisms in which to quantify emergence has been a popular research area in recent days. Generally, authors focus on the relationships that exist among events in simulation (Gore and Reynolds, 2008). For example, inputs are mapped to the outputs and the gap observed is regarded as the amount of emergent behaviour (Schaefer et al., 2002). In others cases, particular events such as the frequency of certain errors or entropy changes in the system are tracked (Hinchey et al., 2005.; Rouff et al., 2004.; Hamann et al., 2011.; Chan, 2011). Thus, there are many events that occur in simulation that can be measured as indicators of emergent behaviour. Our measures of emergence complement one another in establishing convergence time, throughput, and entropy changes.

First, we recommend a quantifier which infers throughput as a measure of emergence. Our choices of *quality of emergence and average delivery rates* are partly inspired by the works of Schaefer et al. (2002). Precisely, we assess the percentage of ant agents that arrive on their targets in a given time frame (which is similarly an investigation of system throughput).

Quantification techniques in which the actions of selected agents in discrete steps are tracked and used as indicators of emergence have also been proposed as well (Noble and Letsky, 2002.; Wang and Zhu, 2007). In these, the average behaviour of the selected group of agents is mapped to the behaviour of individual agents and the swarm (Minati, 2002.; Chan, 2011). Our design of *average end-to-end delays,* as well as *Shannon's information value* are partly inspired by these related assumptions. We particularly determine the average time it takes a selected group of ant agent to travel a complete journey and map this average to the swarm's performances in similar tests. Time in simulation is measured in iterations (Hovda, 2008).

Simpler quantifiers evaluate the time it takes the swarm to converge (Wang and Zhu, 2007). In these, the speed of emergence is the key issues. Our design of the same measure of emergence is inspired by these works, investigating how quickly ant agent swarms achieve their goals, if ever they do.

Given the focus of most quantifiers in the literature (system throughput, speed of emergence, degree of agent engagement on task, etc), issues of quality and timeliness of emergence become apparent (Noble and Letsky, 2002). Generally, quality assesses agents' adherence to schedules and the degree of engagement on task. We are inspired by the contextual definition of quality (as given in Noble and Letsky (2002)). Precisely, we assess the tendencies of ant agents to follow emergent paths as opposed to random wandering (quality of emergence).

In addition, how much information do ant agents have when they make path choices (Shannon's information value)? What is the extent of uncertainty in these decisions? Similar system fluctuation theories have been tested in which agent uncertainty has been quantified (Hamann et al., 2011.; Bavaud et al., 2005.; Martin, 2006.; Schneider, 2007.; Fernández et al., 2013). Our choice of Shannon's information value as a measure of emergence is motivated by these related works.

The discussions we present in this section emphasize on stating the steps we follow in order to extract each measure of emergence during simulation. In the end, we will explain how these measures of emergence are put together in order to determine that index of merit which ranks each XSet against the rest of the XSets in the genetic population and generation.

Although the validity of the measures of emergence we report in each case is tested in details in the next chapter, it is worth mentioning at this point that such validity heavily relies on replicated evaluations of the same tests in order to achieve centrally placed measures of emergence. Next, we discuss how each of these measures of emergence is determined.

4.4.1 Speed of emergence

In this measure of emergence, we assess the time it takes ant agents to converge as a result of using a particular XSet. Previous works have assessed speed of emergence in terms of the frequency of agent interactions in the swarm (Noble and Letsky, 2002), or in terms of the amount of change that is observed in the system over time (Wang and Zhu, 2007). Our premise in this assessment is that, if variations in speed of emergence are observed when different XSets are used, then there exist primitive behaviours in particular XSets which cause these variations.

We indicated that time in simulation is measured in iterations so as to eliminate concerns regarding the speed of emergence we observe being influenced by the processing power of the computer. In this context, an iteration is a single ant step for all ants. Smaller measures correspond to shorter times in simulation before system convergence, implying the use of better XSets.

To measure speed of emergence in a swarm of N ant agents that use a particular XSet, we go through the following steps:

1. We establish the time it takes the first ant agent in the swarm to hit the target. For illustration purposes, let this time be denoted as t_a.

2. We then determine the time it takes half of the swarm ($\frac{N}{2}$ ant agents) to find the same target as well. From observations, half of the agent density is a sufficiently large sample to track in order to reveal the collective behaviour of the swarm. For illustration purposes, let this time be denoted as $t_{a_{\frac{N}{2}}}$.

3. The time gap between $t_{a_{\frac{N}{2}}}$ and t_a indicates the speed of emergence of the swarm towards the target.

4. We also evaluate the time it takes the first ant agent to complete a round trip (travelling from starting point to the target and back). Let this time be denoted as t_b.

5. Speed of emergence towards the starting point is calculated as the time gap between t_b and the time it takes half of the agent density to complete return journeys as well.

6. The average between the speed of emergence towards the target and the speed of emergence in return trips indicates the overall speed of emergence of the swarm in one simulation test.

7. A similar test is re-conducted again and again in order to achieve the central tendency of the swarm (determining the average speed of emergence over many replications).

4.4.2 Quality of emergence

Quality of emergence establishes the frequency with which ant agents in a swarm successfully arrive at their intended targets within a set time frame.

Similar quality tests have been assessed in the literature in terms of the timeliness and efficiency of the outcomes sought (Noble and Letsky, 2002). In this work, quality of emergence assesses the tendencies of ant agents to follow the emergent paths as opposed to random wandering. High frequencies of arrivals indicate the use of well defined and straight paths between the targets and the starting point.

To determine the quality of emergence when a swarm of N ant agents has been deployed using a particular XSet, we follow the steps below:

1. A time frame is set in which the frequencies of ant agent arrivals on targets are recorded. The number of successful trips of ant agents in each direction within the evaluation time limit set are recorded. These frequencies of ant agent arrivals on targets indicate the relative quality of the paths that emerge in each direction.

2. The quality of emergence in one test cycle is, in fact, the average of the qualities of emergence in both directions.

3. The same quality measures are evaluated again and again in order to get a centrally placed quality of emergence that is achieved over many replications.

In this case large measures are better, indicating that more ant agents are able to find their targets within the time frame set.

4.4.3 Average delivery rate

An average delivery rate expresses the frequencies of arrivals of ant agents on their targets as a percentage of agent density. It establishes the rate of success, or throughput of the swarm within a set time limit. Similar evaluations assess average delivery rates in terms of the proportion of outputs that are achieved relative to the proportion of inputs that are fed into the system (Schaefer et al., 2002). Other works evaluate average delivery rate in terms of the frequencies of occurrence of specific parameters in the system over time (Rouff et al., 2004). Relatively bigger average delivery rates correspond to better swarm throughput and good net benefit for the swarm (Powell and Franks, 2007).

To determine average delivery rates in a swarm of N ant agents that are deployed using a particular XSet, we go through the following steps:

1. A time limit is set within which the qualities of emergence are evaluated.
2. The number of successful trips of ant agents in each direction are recorded.
3. The qualities of emergence are then expressed as percentages of agent density.
4. Similarly this is down over a number of replications in order to find average percentages.

4.4.4 Average end-to-end delay

This measure of emergence considers the time it takes an individual ant agent to travel between the starting point and the target or vice verse (including the time the ant agent takes wandering or lost). In this case, time refers to the number of steps an ant agent walks from the time it leaves one end until the time it arrives at the other end. Smaller average end-to-end delays correspond to ant agents walking relatively few steps between the two ends, implying the use of straight paths.

To determine the average end-to-end delays in a swarm of N ant agents that are deployed under the control of a specific XSet, we go through the following steps:

1. A sample swarm of n ant agents is randomly drawn from the deployed swarm, where $n \leq N$. The procedure through which we determine a significantly representative sample size is presented in section 4.4.6.
2. The number of steps each ant agent in the sample group walks from the starting point to the target and vice verse are tracked and recorded. An average number of steps walked in both directions per ant agent is calculated.
3. An average of the average steps of the n ant agents that are tracked is calculated. That average is assumed to centrally place the number of steps an ant agent walks in one simulation cycle.
4. The tests are replicated over many repeated simulations in order to achieve a centrally placed average end-to-end delays.

4.4.5 Shannon's information value

This measure of emergence is built on Shannon's measurement theories (Martin, 2006.; Schneider, 2007.; Fernández et al., 2013). It determines the average amount of information that is available to each ant agent when it makes path selection choices. Precisely, it evaluates the degree of uncertainty that is associated with an ant agent's path selection decisions in a movement step (Bavaud et al., 2005). Shannon's information value is an opposite of uncertainty.

To determine Shannon's information in a swarm of N ant agents that are deployed using a particular XSet, we follow the steps below:

1. A sample swarm of n ant agents is drawn from the deployed swarm (see section 4.4.6 on how a representative sample is drawn).

2. Uncertainty in each sampled ant agent's path choices is determined in each step. We denote uncertainty as H (Schneider, 2007). In the stigmergic category, $H = -A_L log_2 A_L$, where A_L is the attractiveness of a location L around the ant agent at the time. Chapter 3 explained how the attractiveness of locations around a stigmergic ant agent are calculated. The formula to determining H is taken from literature which discusses Shannon's measurement theory (Martin, 2006.; Schneider, 2007.; Fernández et al., 2013). In the message passing category, $H = 3 \times |w_i(t+1) - w_i(t)|$, where $w_i(t)$ and $w_i(t+1)$ are successive vector weights (levels of confidence in the vectors followed at time t and time $t+1$ respectively). This formula is mathematically derived from analogies in Shannon's measurement theories (Schneider, 2007).

3. Shannon's information value is denoted as I, and is calculated as the gap between the highest uncertainty measure possible and the uncertainty measure yield. In this case $I = 3 - H$ because each ant agent has at most 8 location around, and $log_2 8{=}3$ is the highest possible amount of information.

4. The average amount of Shannon's information around an ant agent in each step is of interest. This is found as follow: $\frac{\sum_0^k I}{k}$, where k is the number of steps an ant agent walked to this far.

5. The average amounts of information around all sampled ant agents are put together in order to determine the overall average amount of information around ant agents in this swarm in one simulation cycle.

6. The process is replicated a number of times as well in order to smoothen the average amount of information around each ant agent.

4.4.6 Sampling issues

Determining the average end-to-end delays, as well as finding Shannon's information values require us to extract samples of n ant agents to track. It is a statistical requirement that we extract a sample that fairly represents the entire swarm. This is a critical statistical problem which requires us to make a number of assumptions in order to determine that appropriate sample size.

First, we must stipulate an acceptable margin of error which indicates the biggest difference that is allowed between the performances of a sample ant agent and the performance of any other ant agent in the swarm. Let this margin of error be denoted as e (an acronym for *error*). This value is an estimate which, according to Lohr (2010), is calculated using the formula in equation (4.1).

To simplify equation (4.1), we decide on a significance level. This is a percentage of the sample from which we hope to achieve performances that would fall within the accepted margin of error (Lohr, 2010). In statistics , significance levels are denoted as α, where $\alpha = 1 - p$, and p is the accepted percentage of the sample that will give accepted performances. Commonly, p is set to 99%, 95% or 90% (Francis, 1990). However $p = 95\%$ is more popular because it is centrally placed. In line with the norm, we consider $p = 95\%$ as our accepted percentage of success of ant agents in the sample, which gives $\alpha = 0.05$.

Then, we determine the level of dispersion (standard deviation) that is expected and accepted in the performances of the sampled ant agents. This measure is denoted as S in equation (4.1). N is the agent density. As a result, it is possible to estimate S as $\frac{0.5 \times N}{4}$, because 95% of the values from a normally distributed population are within 2 standard deviations of the mean (Lohr, 2010). Thus, 95% of the values from a normal population are within the range $\bar{x} - 2 \leq \bar{x} \leq \bar{x} + 2$, hence the division by 4 (Lohr, 2010).

The remaining unknown variable in equation (4.1) is n, which represents the required sample size. Equation (4.2) expresses n as the subject of the formula by squaring and simplifying equation (4.1).

$$e = z_{\frac{\alpha}{2}} \sqrt{\left(1 - \frac{n}{N}\right)} \frac{S}{\sqrt{n}} \tag{4.1}$$

$$n = \frac{z^2_{\frac{\alpha}{2}} S^2}{e^2 + \frac{z^2_{\frac{\alpha}{2}} S^2}{N}} \tag{4.2}$$

For example - suppose we have an agent density of 5000. The appropriate sample size to extract from this swarm is calculated as follows:

$N = 5000$.
p = 95%. Therefore $\alpha = 0.05$.
From statistical tables: $z_{\frac{\alpha}{2}} = 1.96$.
$S = \frac{0.5 \times 5000}{4} = 625$.
The margin of error is the greatest possible difference between the average performances of sample ant agents and the performances of any ant agent in the population. If our environment is 100×100, the value of e is the average of the midpoints along each dimension of the environment.
Thus: $e = \frac{(50+50)}{2} = 50$.
Therefore $n \geq \frac{1.96^2 \times 625^2}{50^2 + \frac{1.96^2 \times 625^2}{5000}} = 535.91357$.
Rounding this result forward, $n \geq 536$

4.4.7 Determining the index of merit

The final task in quantifying emergent behaviour is to determine the index of merit that is associated with using a particular XSet. We defined an index of merit (IOM) as a value which indicates the extent to which an XSet allows specific emergent behaviour to occur relative to the performances of other XSets in the current genetic population and generation. In this section, we procedurally show how we arrive at this value.

1. We first scale the speed of emergence we found in order to have it lie within the range $[0; 1]$. This is done in order to standardize the weight of each of the five measures of emergence when we combine their effects. Let the speed of emergence that was yield when a particular XSet i was used be denoted as s_i. We indicated in section 4.4.1 that speed of emergence assesses the time it takes ant agents to achieve a specific task. We also indicated that smaller time measures are favourable. Our simulations are coordinated over a set time limit (provided by the user as initial conditions). Let this time limit be denoted as T. Therefore, 0$\leq s_i \leq$T and $0 \leq \frac{s_i}{T} \leq 1$. However in this measure of emergence, smaller speed measures correspond to ant agents walking fewer steps to get to their targets, and are favourable. Mathematically, the scaled speed of emergence we desired is therefore $(1 - \frac{s_i}{T})$, a float value within the range $[0; 1]$.

2. The quality of emergence that is achieved when the same XSet i was used is also scaled in order to prevent it from unfairly contributing its effects to the final index of merit. We indicated that quality of emergence establishes the frequency with which ant agents in a swarm successfully arrive at their intended targets within a set time frame, and bigger values are better. Let the quality of emergence achieved be denoted as q_i. These quality measures are extracted within a set time

frame T. Therefore $0 \leq q_i \leq$T, and $0 \leq \frac{q_i}{T} < 1$. The scaled quality measure we require is $\frac{q_i}{T}$ because bigger values are better. Note that the scaled quality measure is also a float value within the range $[0; 1]$.

3. Average average delivery rates are complementary to the qualities of emergence. They express quality as a percentage of agent density. Being percentages, they are therefore already scaled to lie within the range $[0; 1]$. If we denote average delivery rate of the i^{th} XSet as d_i, then $0 \leq d_i \leq 1$. The scaled quantity we require remains the same d_i expressed as a decimal float.

4. Scaling the average end-to-end delays is similar to weighting the speed of emergence. Let average end-to-end delays be denoted as a_i. If our simulation limit remains T. Then $0 \leq a_i \leq$T, and $0 \leq \frac{a_i}{T} \leq 1$. However, smaller measures correspond to ant agents walking fewer steps to get to their targets, and are favourable. As a result, the scaled average end-to-end delays we require is $(1 - \frac{a_i}{T})$, a float value within the range $[0; 1]$ as well.

5. Shannon's information is the last measure of emergence we consider and scale. Let this measure be denoted as I_i. We indicated that large measures indicate more information around an ant agent which is desirable. The maximum amount of information in the environment setup we provide is 3 because $log_2 8 = 3$, where 8 is the maximum number of possible destination locations around an ant agent. Therefore $0 \leq I_i \leq 3$, and $0 \leq \frac{I_i}{3} \leq 1$. The scaled value we require in this case is $\frac{I_i}{3}$ since bigger values are better, which is also a float value within the range $[0; 1]$.

6. Each scaled measure of emergence is an indication of how much emergence has occurred in the system, and these measures are complementary. Equation (4.3) calculates the average of these five scaled measures

of emergence which, in our context, is called the index of merit of XSet i. An average measure of the scaled measures is preferred since it depicts the common view in performances when a particular XSet is used. We are saying that the five measures of emergence all evaluate the relationships between particular events in simulation, and their average statistically tells the central tendency in these events - which is the reason why we evaluate XSets using more than one metric at a time. Note that all these measures are scaled on the scale in $[0; 1]$. The average value is thus used to rank different XSets in the same genetic generation and population.

$$IOM_i = \frac{\left(1-\frac{s_i}{T}\right)+\frac{q_i}{T}+d_i+\left(1-\frac{a_i}{T}\right)+\frac{I_i}{3}}{5} \tag{4.3}$$

The next section presents an experimental setup in which we validate the processes that are discussed in this chapter so far. It validates the generation of XSets, the creation of environments, the selection of XSet for use at a particular time in simulation, as well as the measures of emergence we propose. In the end, the next section provides an answer to the fourth task of this chapter (see section 4.1.2 for details regarding this task).

4.5 Evaluation of XSets

Three categories of genetic populations of XSets (*stigXSets, msgXSets, hybXSets*) can arise in which various member XSets are candidate control dictionaries for desired emergent behaviour. This section explores these three categories of XSets for best controller XSets for allowing a particular case study example of emergent behaviour to occur - the path finding behaviour in swarms of ant-like devices. Our premise in these investigations is that evidence of properties in a particular XSet, properties for allowing a particular form of emergent behaviour to occur will support the hypothesis which says: *there exists collections of particular primitive behaviours and parameters that are arranged in specific sequences which form an ant agent language for creating predictable emergent behaviour.*

First, we describe the context of the path finding problem we assume. Then we present the design of the experiment in which we search for those particular XSets that best allow path finding behaviour to occur in each category of populations of XSets.

The results of this experiment validate both the primitive behaviours we proposed, the measures of emergence we suggested as quantifiers of emergent behaviour, as well as the XSets themselves. We emphasize again that the choice to evaluate XSets on the path finding problem domain is only to prove the concepts of primitive behaviours and XSets. It is not the path finding outcome that matters, but rather the discovery of those XSets that give rise to a specific outcome - the path finding outcome.

4.5.1 The path finding problem: A case study

In the context of this thesis, the path finding problem can be re-defined as follows:

- *Ant agents in swarms of specific agent density are deployed in selected evaluation environments, using particular XSets as control toolboxes. These ant agents have the common task of locating food-like targets that are situated on the evaluation environment, and upon finding these food-like targets, travel back to the starting point. The trips between the starting point and the food-like targets are repeated again and again*

> *until the simulation time limit lapses. In each step, every ant agent has low level and autonomous tasks of orientating, updating swarm information, evaluating the local environment for target indicators and act accordingly, and moving to the next selected location.*

This is a known and common ant problem domain in the literature (Panait and Luke, 2004a.; Dorigo et al., 1999). Stigmergic ant agents, in particular, rely on the levels of pheromone they place on the environment for indirect interaction (Dorigo, 1992.; Panait and Luke, 2004a, 2004b). On the other hand, the novel model of message passing ant agents rely on implicit communication spaces that arise in which direction vectors are shared.

All ant agents remain in constant motion, moving at a constant speed of one grid cell per step, and executing a constant set of instructions throughout. Although ant agents are completely unaware of the environment in which they are deployed, paths emerge between the starting point and the target as emergent behaviour of the swarm (Gulyas et al., 2006). What do ant agents do as individuals in a swarm which give rise to path finding behaviour at swarm level?

4.5.2 Experiment design

We administer an experiment in which the key objective is to search for optimal XSets in each category which best allow a particular form of emergent behaviour to occur (in this case, a case study of the path finding behaviour). In other words, we evaluate the abilities of swarms of ant agents to establish the shortest path between a target and the starting point.

The null hypothesis that drives this experiment is that: *there are no significant differences between the average indices of merits that are achieved when* *different XSets that are taken from the same genetic population are compared* - ($H_0 : \mu_1 = \mu_2$). Therefore the alternative hypothesis is that: *there exist particular XSets in these search spaces which relatively best describe a language for programming ant agents towards achieving particular emergent behaviour, in this case, path finding behaviour.*

The only dependent variable in this experiment is the *index of merit* of an XSet. In this context, a dependent variable is that parameter which we measure or investigate in an experiment. We particularly investigate the indices of merits of XSets using the five measures of emergence we discussed in section 4.4. Our goal is to report the composition and properties of best performer XSets in each category.

Five independent variables are manipulated in this experiment. In this context, an independent variable is that parameter which we manipulate, vary, and monitor in an experiment. First, we monitor the category of XSets we use at a time. We indicated already that there are three categories possible, namely *stigXSets, msgXSets,* and *hybXSets*. Our aim at the end is to report best controller XSets in each category, and hopefully compare the performances of these best performer XSets in the next chapter.

The second independent variable of this experiment is the composition of the XSets we use. Section 4.2 explained how different partial permutations of the set of primitive behaviours U result in different composition and sequences in XSets. We also explained how genetic programming processes evolve new XSets and genetic populations. We monitor the composition and sequences of XSets with the goal of explicitly reporting, in the end, the composition, configuration, cardinality, and other properties of best controller XSets for, in this case, the path finding problem.

We also monitor the cardinality of the XSets as we go through the search spaces. In line with our dictum of proposing simple and naive ant agents, it is desirable to recommend XSets with the smallest cardinality possible.

Time in simulation is another key independent variable in this experiment. We particularly allow a default limit of 10000 iterations in which each XSet's measures of emergence are scored. However if desired, this limit can be changed by the user at run-time. Thus, a swarm runs for 10000 steps before an index of merit is determined. The same swarm is redeployed and allowed to run for another 10000 iterations, scoring new measures of emergence. The swarm is allowed to repeat the same tests for ten times, in each case determining an index of merit. The final index of merit of the XSet is the average of the 10 outcomes that are reported in each replication.

In addition, measures of emergence are extracted at intervals of 1000 iterations. This is also a parameter the user can change before the simulation commences. We call each interval or simulation stage a *control level* (1000, 2000, 3000 until the allowed scoring limit, in this case 10000). From observation, intervals or gaps of 1000 iterations between sample measures of emergence are sufficiently large to reveal the effects of time and that of the primitive behaviours thereof.

The rest of the variables we require in this experiment are controlled. We understand controlled variables as those parameters that are kept constant throughout the experiment because they are not the subjects of study at the moment. Key on the list of controlled variables are: agent density, environment configuration, and evolution limit. For illustration purposes, we use default settings of 5000 ant agents and 100×100 grid environments. Each category is allowed the same evolution limit of $10,000,000$ in which to evolve better XSets. We indicated that these variables can also be changed by the user at run-time. We also indicated that controlling these variables does not influence the results we report since the goal of this experiment is neither to investigate the effects of agent density nor to study the effects of environment complexity to emergent behaviour. Rather, we investigate the composition of XSets which give rise to emergent behaviour. From observation, 5000

Title: To identify XSets which best describe a language for programming ant agents towards achieving particular emergent behaviour (path finding towards a particular target).

Null-hypothesis: $H_0 : \mu_1 = \mu_2$)- there are no differences between the average performances of different XSets that are taken from the same search space.

Alternative-hypothesis: there exist particular XSets in the search spaces which best describe a language for programming ant agents towards achieving particular emergent behaviour, in this case path finding emergent behaviour.

Dependent variable : index of merit

Independent variables : Category of XSets ; Composition of XSets ; Cardinality (maximum of 10) ; Time in simulation (maximum of 10000 iterations) ; Control levels (every 1000^{th} iterations).

Controlled variables : Agent density (5000) ; Environment size (100×100) ; Position of starting point (fixed) ; Centre of target (centre of environment)

Procedure - Generator functions are invoked which define a path finding environment. Ant agents are deployed at random locations over ten replicated simulations. Average measures of emergence are calculated at each control level and reported. The algorithm below summarizes the procedure.

```
foreach category
  generate initial population of XSets : bestIOM = 0 : bestXSet=null
  foreach generation
    foreach XSet j in current generation : deploy n ant agents
       foreach replication i
```

Control level	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
speed										
quality										
delivery										
delays										
Shannon										
cumm. IOM	$x_1 += IOM_{1,i}$	$x_2 += IOM_{2,i}$	$x_3 += IOM_{3,i}$	$x_4 += IOM_{4,i}$	$x_5 += IOM_{5,i}$	$x_6 += IOM_{6,i}$	$x_7 += IOM_{7,i}$	$x_8 += IOM_{8,i}$	$x_9 += IOM_{9,i}$	$x_{10} += IOM_{10,i}$

```
       until 10 replications
```

$index\,of\,merit = \frac{\sum_{k=1}^{10} \frac{x_k}{10}}{10}$

```
       update the XSet's field which holds the index of merit
        if current index of merit > bestIOM
            bestIOM =current index of merit : bestXSet = ID of current XSet
       endif

     next XSet in current population
     genetic programming processes : Evolve next generation of XSets
  until EvolutionLimit
  Report bestXSet in the category
next category
```

Figure 4.9: Experiment design

ant agents are sufficiently many to reveal the properties we require in each XSet, especially on 100×100 grid environments, over a large evolution limit we propose. The placement of the starting point and targets was discussed in section 4.3.

To enhance reliability in the indices of merits we report, each experiment picks an XSet from a particular genetic population and repeatedly evaluate this XSet for ten times (where each cycle has ten control levels). The measures of emergence that are extracted at each control level of each of these ten evaluations, and the indices of merits that are computed at each of these control levels, are accumulated and averaged over the ten replications ($x_k = x_k + IOM_{k,i}$). A standard deviation measure at each control level is tracked which validates the central tendencies we observe. Thus, each x_k is, in fact, the sum of the ten indices of merits that are achieved at control levels $k \times 1000$. Dividing these sums by 10 give centrally placed indices of merits at each control level. In this work, the overall index of merit of the j^{th} XSet in the search space is found by determining the average of the centrally placed indices of merits per control level, i.e. $IOM_j = \frac{\sum_{k=1}^{10} \frac{x_k}{10}}{10}$. This is the value we compare with the performances of the rest of the XSets in the same search space. The composition of the XSet with the best *index of merit* value is reported.

Our simulation system sequentially picks the next XSet for evaluation from the same genetic population until all XSets are assessed. Every time a new XSet is chosen for evaluation, the simulation restarts the whole experiment all over again. At the end, the best XSet in each category is identified. This process is repeated for all the three categories of XSets. Many generations are recommended in order to allow the system more time to evolve better and better XSets.

Figure 4.9 summarizes this procedure, showing the key variables of the experiment, the steps through which the final index of merit of each XSet is calculated, as well as how the best performer XSets in each category are identified.

4.5.3 Results

The key outputs of the experiment we administered are: characterization of the genetic populations that arise, discussion of the composition of best performer XSets, verification of the performances of best performer XSets, and demonstrating the visual outputs that arise when ant agents are tracked for isolation (how frequent do ant agents remain isolated?).

4.5.3.1 Properties of the genetic population

The choice of *antType* has a number of influences over the characteristics of the genetic population that arises. Three categories of genetic populations are therefore possible. These are determined by the number of primitive behaviours that are valid in each category. Six primitive behaviours are valid in the stigXSet category. On the other hand, there are 5 primitive behaviours that are valid in the *msgXSet* category. The biggest chunk of XSets are in the *hybXSet* category. In each case, the (NOp :) instruction is added on the list. Precisely:

- The initial population in the *stigXSet* category is randomly picked from: $\frac{7!}{(7-2)!} + \frac{7!}{(7-3)!} + \frac{7!}{(7-4)!} + \frac{7!}{(7-5)!} + \frac{7!}{(7-6)!} + \frac{7!}{(7-7)!} = 42 + 210 + 840 + 2,520 + 5,040 + 5,040 = 13,692$ possible XSets whose cardinality inclusively range between 2 through 7. This is a large search space in which a random selection may miss those XSets with better traits for allowing desired emergent behaviour, hence the need to allow the user to choose the highest cardinality.

- The initial population in the *msgXSet* category is selected from : $\frac{6!}{(6-2)!} + \frac{6!}{(6-3)!} + \frac{6!}{(6-4)!} + \frac{6!}{(6-5)!} + \frac{6!}{(6-6)!} = 30 + 120 + 360 + 720 + 720 = 1,950$ XSets whose cardinality inclusively range between 2 and 6. Randomly picking 500 XSets out of this sample of $1,950$ candidates may also miss XSets with better traits to evolve a good population.

- The *hybXSet* category has even a larger sample from which to pick the initial population. It consists of $\frac{10!}{(10-2)!} + \frac{10!}{(10-3)!} + \frac{10!}{(10-4)!} + \frac{10!}{(10-5)!} + \frac{10!}{(10-6)!} + \frac{10!}{(10-7)!} + \frac{10!}{(10-8)!} + \frac{10!}{(10-9)!} + \frac{10!}{(10-10)!} = 90 + 720 + 5,040 + 30,240 + 151,200 + 604,800 + 1,814,400 + 3,628,800 + 3,628,800 = 9,864,090$ XSets. However $8,877,690$ of these XSets contain the $(NOp :)$ instruction somewhere in their sequences, defining redundant combination. This finding that large cardinality often result in redundant combination is consistent with redundancy in evolutionary computation which also finds that large genomes contain redundancy. In addition, we note that the (NOp:) instruction potentially affects timing / synchronization of interaction operations, so may affect outcomes. Of the remaining $986,400$ XSets, $13,692$ are purely stigmergic while $1,950$ are purely message passing XSets. These XSets are not reconsidered in this category again. Although $970,758$ XSets remain in the search space thereof, most of these XSets are still invalid in that they may contain instructions that cancel the effects of one another, or invalid collections such as trying to evaporate pheromones when the rest of the instructions in the XSet are message passing. However we observe that penalizing such collections before assessment may compromise population diversity.

We make three observations regarding the generation of the initial population:

- Although long evolution limits require more machine time to evolve better XSets, the quality of the final genetic population thereof heavily relies on this evolution limit parameter. It is a better choice to consider quality at the expense of more time in simulation.

- A mechanism is required in which to select useful combination when we pick the initial population of XSets. Two ways are possible: (a) we can increase the GA population size so that as many possible combination are included in the initial population as possible. However, large GA population sizes slow down the ranking and evaluation of XSets in each genetic generation. The system would take too long to evolve better generations of XSets. (b) We may also consider to discard the (NOp :) instruction when we determine the partial permutations of primitive behaviours in each category. That would reduce the number of possible permutations and increase the quality of the initial population thereof. This would reduce the search spaces in the stigmergic and message passing categories to 1950 and 320 XSets respectively. Thus, most useful combination of XSets would be considered in the initial population.

- The use of partial permutations (combination without repetition) when we create the initial population enhances diversity. It prevents populating the initial population with XSets that are similar or close to each other in configuration. We observe that the quality of subsequent generations of XSets relies on the diversity we emphasize on in the initial population.

4.5.3.2 Configuration of best performer XSets in each category

Understanding the composition and properties of the best performer XSets is the critical outcome sought. The results we discuss in this section are achieved when arbitrary choices of parameters were made as shown in table 4.1.

AntType	**stigXSet**	**msgXSet**	**hybXSet**
Highest cardinality supported	6	6	6
Internal states	4	4	4
Agent memory (in no. of blocks)	4	8	8
Environment size	100×100	100×100	100×100
Environment dimensions	2D	2D	2D
Environment name	pathFind	pathFind	pathFind
Scoring time (in iterations)	10,000	10,000	10,000
control level intervals (in iterations)	1,000	1,000	1,000
Agent density	5,000	5,000	5,000
Max. No. of pheromone	5	2	5
GA population size	500	500	500
Selection method	Tournament	Tournament	Tournament
Selection pressure	5	5	5
Elite gene rate	5%	5%	5%
Cross over rate	80%	80%	80%
Mutation rate	15%	15%	15%
Evolution limit (iterations)	10,000,000	10,000,000	10,000,000

Table 4.1: Parameter settings

Top on the list of the observations we make is that good performer XSets occur when cardinality ranges between 3 and 5. In this context, a good performer XSet is one that demonstrates evidence of emergent properties although it may not be the best XSet in the category. Those XSets whose cardinality are below 3 are completely insufficient controllers, particularly for the path finding problem we investigate.

Another key observation is that the stigmergic and message passing XSets widely out-class hybrid XSets. This observation suggests that good swarms are specialist with respect to the interaction techniques that are used by the ant agents in these swarms. This observation is in line with biological views of worker ants and reproductive queen ants that are specialists in their colonies (Tsutsui and Suarez, 2003).

We observe that hybrid XSets that demonstrate some evidence of emergent properties occur when cardinality is relatively high. However the ant agents in this category remain more of generalist ant agents (Imai, 1966) that would not solve particular tasks. Worse still, most of such XSets comprise of unnecessary instructions that are not useful to the ant agents, such as supporting diffusion in almost message passing XSets, or normalizing vectors in almost stigmergic XSets. In our views, XSets that include unnecessary primitive behaviours in their composition are against our dictum of keeping the ant agents simple and naive.

We remind the reader that this work has been referred to as a preliminary study of XSets as an approach for defining an ant agents language (see section 1.1 for this reference). We also remind the reader that there is potential to carry out similar studies on other ant agent metaphors and extend the set of primitive behaviours for ant agents, thus increasing the search spaces in which, hopefully, *hybXSets* may be useful. As a result, the results we report next ignore hybrid XSets, thus going with the idea of specialist swarms.

After an evolution limit of $10,000,000$ iterations, the best performer *stigXSet* for the path finding problem was identified. Key in the configuration of this XSet (see Figure 4.10 for the configuration of this XSet) is that:

- ant agents in this category must be able to drop specific levels of pheromone in specific quantities - $(Drp : p_i, q)$. In particular, they must drop the levels of “home pheromone” (whose ID is 2) when they are in the seek mode (seek mode is the internal state an ant agent assume when it is travelling from the starting point towards the target. See Figure 3.4), essentially marking trails that are useful to ant agents that are travelling towards the starting point. In the return mode (when an ant agent is travelling from the target to the starting point), the same ant agents must drop the levels of “food pheromone” (whose

ID is 3). Pheromone 3 marks trails with directional cues towards the target. When ant agents in this category arrive on the target or at the starting point, they do nothing but flip back to the opposite internal state, after which they commence the reverse trip all over again. Note that the amount of pheromone an ant agent drops at a time is a float parameter greater than 0. We observe that quantities below 1 slow down the swarm's convergence time. On the other hand, quantities above 1 saturate the environment too soon in simulation.

stigSet[4,4,4]: $< (Drp:2,1), (MvH:3,3,2,0.5,0.5,-1), (MvP:x,y,0), (StS:1,0,0) \mid (NOp:), (NOp:), (NOp:), (StS:2,0,0) \mid (Drp:3,1), (MvH:2,2,3,0.5,0.5,-1), (MvP:x,y,0), (StS:3,1,0) \mid (NOp:), (NOp:), (NOp:), (StS:0,1,0) >$

Figure 4.10: Best performer XSet in the stigXSet category

- these ant agents require a mechanism for achieving informed orientation based on the relative attractiveness of locations around - $(MvH : \tau_i, \tau_i, \eta_i, w_\tau, w_\tau, w_\eta)$. In the seek mode, stigmergic ant agents are attracted to the levels of pheromones 3, which mark trails towards the target. The same ant agents would penalize movements towards locations with high levels of pheromones 2 which mark trails with directional cues towards the starting point. In the return mode (when an ant agent is travelling from the target to the starting point), stigmergic ant agents are attracted to the levels of pheromone 2, while at the same time penalizing movements towards locations with high levels of pheromone 3. As a result, ant agents keep travelling between the starting point and the target for the entire scoring time. We observe that a fifty fifty weighting of the effects of attractive and repulsive levels of pheromone ($w_\tau = 0.5, w_\tau = 0.5, w_\eta = -1$) achieves more realistic orientation outcomes.

- the best performer XSet in this category also contains a primitive behaviour with which ant agents make probabilistic movement choices. The direction the ant agents follows is selected during orientation, implying order in the arrangement of the primitive behaviours (with the primitive behaviour for orientation coming first before the primitive behaviour for ant agent movement) - ($MvP : x_i, y_i, z_i$).

- All ant agents in this category must be able to detect targets in their vicinity, also implying order between that ability to detect targets and agent movement. After completing all other low level tasks, an ant agent must be able to switch between different internal states when it becomes necessary - ($StS : m, n, q$).

- We observe that dissipation controls are not causal actions in this category. Rather, they enhance the quality of the paths that arise. However that alone compromises the simplicity sought in the system we propose.

- In summary, the smallest stigmergic XSet which out-perform the rest in allowing path finding behaviour to occur has a cardinality of 4. Ant agents in this category require four internal s to solve the path finding task (seek mode, on target mode, return mode, and on starting point). However an ant agent can only be in one of these four internal states at a time in simulation. A minimum of four memory blocks can sufficiently hold all the state information these ant agents require. There is no requirement for specific order between the instructions to drop the levels of pheromone and the one for orientation. However these instructions must both occur before agent movement.

Figure 4.11 shows the configuration of the best performer XSet that is found in the *msgXSet* category after the same evolution limit of $10,000,000$.

msgSet[5,4,8]**:** $<$ $(MsP : 0,0,1)$, $(PtV : 0,0)$, $(Nrm : 0,0,0)$, $(MvP : x,y,0)$, $(StS : 1,0,0)$ $|$ $(NOp :)$, $(NOp :)$, $(NOp :)$, $(NOp :)$, $(StS : 2,0,0)$ $|$ $(MsP : 2,2,3)$, $(PtV : 1,1)$, $(Nrm : 2,2,2)$, $(MvP : x,y,0)$, $(StS : 3,1,0)$ $|$ $(NOp :)$, $(NOp :)$, $(NOp :)$, $(NOp :)$, $(StS : 0,1,0)$ $>$

Figure 4.11: Best performer XSet in the msgXSet category

A notable difference between this XSet and the stigmergic counterpart is that the message passing XSet comprises of five primitive behaviours in each of the four internal states supported. In addition, message passing ant agents require more memory to hold more message blocks in which geometric vectors are held and processed. Precisely, a message passing ant agent's key activities in each internal state are summarized as follows:

- ant agents in this category must exchange or share specific geometric vectors in every step, where each vector is weighted by some confidence weight in that vector - $(MsP : a, a, b)$. In the seek mode, message passing ant agents must share "target vectors" (whose components are stored in memory block 0, and the weights are stored in memory block 1, hence $(MsP : 0, 0, 1)$). Ant agents that are in the return mode must share "home vectors". These are stored in memory block 2, and the corresponding weights are recorded in memory block 3.

- message passing ant agents must be able to detect target indicators in each step, whose specificity is determined by the ant agent's current internal state at the time. If detected, these target indicators must be converted to vectors information - $(PtV : a, b)$ which message passing ant agents can interpret. In this case, target indicators have an ID of 0 and starting point indicators are labelled as 1. When in the seek mode, ant agents seek to detect target indicators and, if possible, convert these to target vectors which point directly towards the target.

Control level	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000	Overall
IOM in stigXSet	0.3887	0.5045	0.6212	0.6853	0.5724	0.5601	0.5343	0.4787	0.4681	0.4228	0.5236
stddev(stigXSet)	0.00434	0.00231	0.00262	0.00023	0.00132	0.00119	0.00054	0.00024	0.00013	0.00021	0.001313
IOM in msgXSet	0.2619	0.4372	0.5684	0.6882	0.7120	0.7969	0.8318	0.8782	0.9167	0.9626	0.7054
stddev(msgXSet)	0.02131	0.01932	0.03103	0.00281	0.00312	0.00123	0.00012	0.000113	0.00018	0.00010	0.0079333

Table 4.2: Performances of best performer XSets

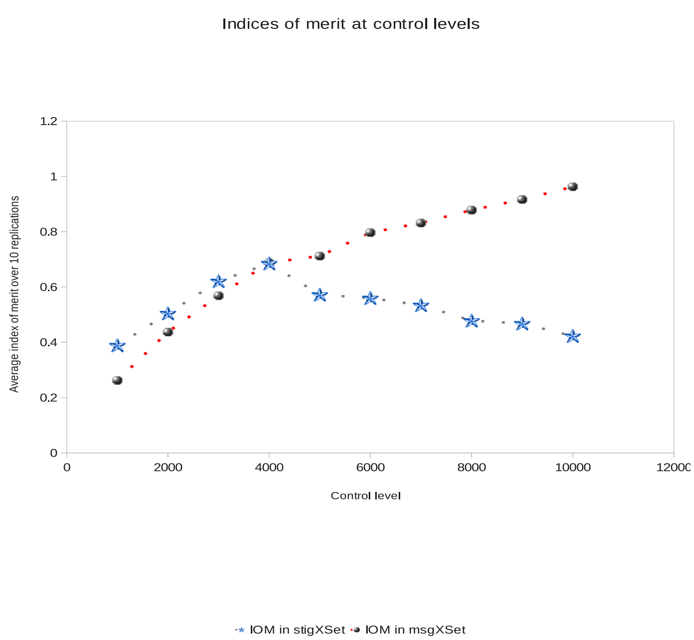


Figure 4.12: IOM in best performer XSets

In the return mode (when an ant agent is travelling from the food source to the starting point), message passing ant agents seek to detect starting point indicators and convert these to home vectors which directly point towards the starting point.

- these ant agents must be able to normalize resultant vectors - $(Nrm : x, y, z)$ in order to standardize ant agent movement steps.
- biased movement steps arise in which message passing ant agents follow the normalized vectors - $(MvP : x, y, z)$ that arose during message passing - $(MsP : a, a, b)$ or the target vector that arose during the detection of target indicators - $(PtV : a, b)$. In the seek mode, ant agents in this category follow the vectors that are held in memory component 0, while in the return mode, they follow vectors that are held in memory component 2.
- similarly, these ant agents must be able to switch between different internal states when it becomes necessary - $(StS : m, n, q)$.
- In summary, the smallest message passing XSet has a cardinality of 5. Ant agents in this category similarly require four internal states to solve the path finding task (seek mode, on target mode, return mode, and on starting point). A minimum of eight memory blocks can sufficiently hold the vectors, weights, and state details that are required for ant agents to complete their tasks. Order is also a requirement between orientation, normalization and then movement.

4.5.3.3 Indices of merits of best performer XSets

Table 4.2 reports the average indices of merits that are achieved at each of the ten control levels when best performer XSets in the *stigXSet* and *msgXSet* categories are used for solving the path finding problem. XSets in the hybrid category are completely out-classed. Worse still, promising XSets in the hybrid category have high cardinality which compromise agent simplicity.

The reliability of the results we report (meaning the average indices of merits per control level) is based on the values being computed over ten replications of the same experiment. We also track the standard deviation measures in order to identify any abnormal dispersions. In addition, all the measures of emergence we use for calculating the indices of merits have direct links to previous works in the literature, hence reliable.

The procedure through which we repeat the simulation, record, and accumulate results before we determine the overall indices of merits has been discussed in details in section 4.5.2. The meanings of these average indices of merits are further elaborated in Figure 4.12 which shows the bigger picture or trend changes over time. We make the following observations regarding these results:

- successful identification of particular XSets which best describe languages for programming ant agents towards particular emergent behaviour (path finding behaviour) in the stigmergic and message passing categories is a milestone in this thesis. This finding is evidence that *there exist particular XSets in the search spaces we study, which best describe languages for programming ant agents towards desired emergent behaviour.*

- the quality of the indices of merits we achieve from using a particular XSet relies on the composition of the XSet, the sequences of the primitive behaviours involved, the number of internal states that are supported at the time, cardinality, as well as the number of memory blocks an ant agent can hold at a time. Best and smallest (in cardinality) performer XSets in the stigmergic category consist of at most four primitive behaviour in each internal state (see Figure 4.10). Ant agents in this category require four internal states to solve the path finding problem. The same ant agents successfully solved the path finding problem using four memory blocks. In the *stigXSets*, the instruction for agent movement strictly follows after agent orientation and pheromone update. On the other hand, best performer XSets in the message passing category consist of at most five primitive behaviour in each ant agent internal state (see Figure 4.11). Ant agents in this category similarly require four internal states to solve the path finding problem. They successfully solved the path finding problem using eight memory blocks. Similarly, the instruction for agent movement strictly follows after agent orientation, target detection, and vector normalization. This observation is in line with the analogy of genes and chromosomes we presented earlier on, where describing the order in which genes (primitive behaviours) are configured in chromosomes (XSets) are important findings in medicine and genetics (Singh et al., 2012).

- Best performer XSets in the stigmergic category allow swarms of ant agents to improve in performances early in simulation time until a threshold turning point is reached, after which the model depletes. This is because the levels of pheromones that are deposited onto the environment would reach a point when they saturate the environment. The paths thereof would get wider and wider with time in simulation, thus reverting ant agents into random wandering. This is a side effect which can be rectified by allowing pheromone dissipation controls. However, as we mentioned already, pheromone dissipation properties are merely enhancers, and not causes of emergent behaviour. On the contrary, the best performer XSets in the message passing category can propel swarms of ant agents towards deterministic paths. This is because, once vector fields are established which point towards the targets with high vector weights, ant agents often travel shorter distances between the two points (starting point and food sources).

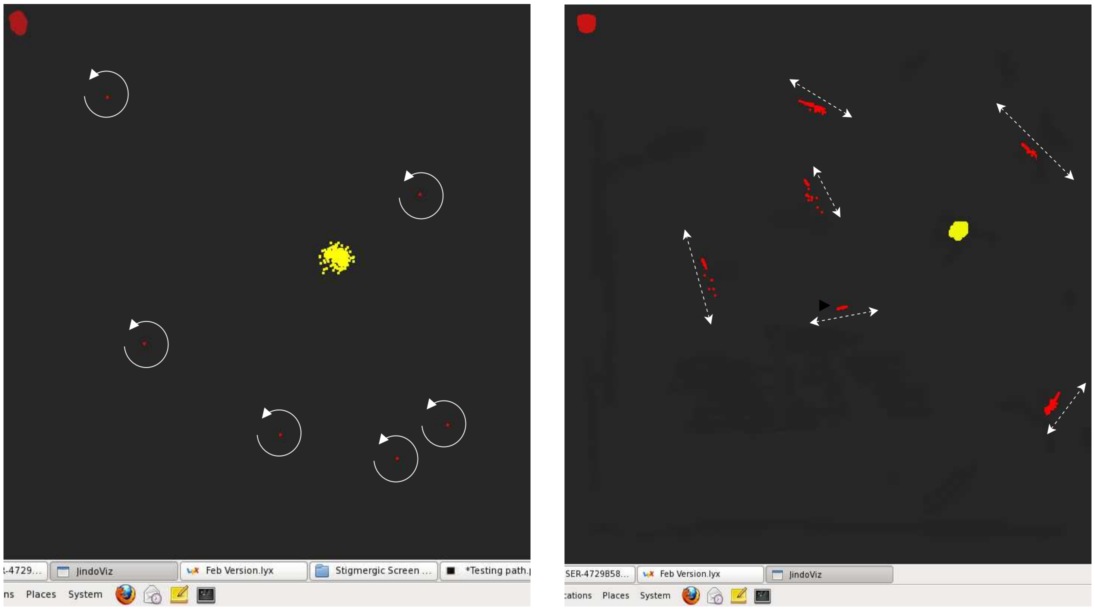
- Stigmergic swarms have less tendencies to follow the paths that emerge. This is because path choices remain probabilistic regardless of how well the ant agents are performing. As a result, chances that the same ant agents derail off the paths are highly common. However, although this sounds like a flaw, it is in fact a fault tolerance property in the event of the emergent path being detoured for one reason or another. On the contrary, message passing path choices are determined by the knowledge held in neighbouring ant agents. If the swarm is performing well, it is therefore likely that ant agents would perform better as well, thus learning from one another. As a result, the swarm can converge on desired vector fields. In general, message passing ant agents achieve better quality paths than the stigmergic counterparts.

4.5.3.4 Visual evidences of the performances of best XSets

Figure 4.13 visualizes the spread of ant agents that never found the target throughout the simulation period of 10000 iterations when each best XSet was used. Ant agents are indicated by the small red spots. Most of the ant agents we visualize are isolated from the rest of the swarm that converged on the shortest path. We indicate the direction of motion of these ant agents or clusters of ant agents in order to further clarify the visual effects of each category of XSets.

The screenshots also show the density of hits on the targets by those ant agents that converged on the emergent paths. Precisely the yellow spots indicate the spots at which ant agents perceived target indicator and flipped from the search to the return internal state.

We make the following observations from these visualized performances:



(a) Stigmergic swarms (b) Message passing swarms

Figure 4.13: Density of ant agents that remain isolated

- Isolated ant agents in the stigmergic category are often trapped in sub-optimal solutions. They are often trapped in circular movements because a local maxima arose. A local maxima occurs when some locations contain relatively higher levels of attractive pheromone than their neighbours. As a result, ant agents that visit those locations would consider travelling back to the same locations in every next step because the location is far much more attractive. This flaw is rectified when other ant agents create paths which pass by these local maxima (creating a gradient which repairs the local maxima). However these are rare cases in this category for two reasons. First, local maximas arise when ant agents are centrally deployed at the same location, and are all allowed to drop the same levels of pheromones on that location before commencing the search trips. Ant agents that would visit the location where ant agents were deployed would be trapped. We have resolved this flaw by deploying the ant agents at random locations on the environment (thus avoiding the creation of local maximas). However our ant agents can still co-exist. The few cases of isolated ant agents we see in Figure 4.13(a) may be a result of coincidental distribution of ant agents at the same location during deployment. In this particular simulation, only 6 out of 5000 ant agents remained isolated in the stigmergic category (which amounts to 99.88% success rate). The second reason why it is rare to observe isolated ant agents in the stigmergic category is because the path selection policies that are followed emphasize on ant agents repelling away from the levels of pheromones they themselves dropped in favour of locations with those levels of pheromone that were dropped by ant agents in the opposite internal state. It is therefore common that ant agents would penalize movements towards the locations they once visited. In this way, the stigmergic model demonstrate better robustness, adaptability, and fault tolerance.

- The message passing model creates emergent clusters in which common vector fields are followed. Ant agents that hit the target can influence other ant agents in the same cluster to exhibit a following behaviour towards the same targets. Eventually deterministic paths arise between the target and the starting point (only for those ant agents that are in the successful clusters). However clusters of lost ant agents also arise. These would create common but sub-optimal vector fields in which ant agents keep searching for the targets without success. However the confidence factors in the vectors the ant agents follow in such clusters would drop to lowest levels. In the event of ant agents in such clusters merging with more successful clusters, they would abandon the vectors they follow in favour of the vectors being followed in successful clusters because they have less confidence in their own performances. However if they never get to meet with more successful clusters, isolated ant agents in this category may remain following falsely agreed vectors for the entire scoring time. In this case, we observe six clusters of 29, 16, 7, 38, 15, and 19 isolated ant agents. This gives a success rate of 97.52%, which is still attractive (obtained by subtracting isolated ant agents from agent density, divide the answer by agent density, and convert the result to a percentage). Most isolated message passing ant agents travel in the same linear direction in the same internal state.

◦ The two observations we made above suggest that failures of ant agents at individual levels do not affect the completion of swarm level goals. This is a known advantage and property of successful swarm intelligence systems. Thus, the XSets we identified show known properties in swarm intelligence systems.

◦ Stigmergic ant agents demonstrate better and cooperation (evident from the number of ant agents that remain isolated). We attribute this observation to the mechanism in which information is held in each model. Stigmergic ant agents, in particular, create shared memories on the environment, thus allowing isolated ant agents to indirectly interact with the rest of the swarm and self-organize. Message passing ant agents would require one-on-one interactions in order to achieve informed path choices. As a result, isolated ant agents in the message passing category would remain lost until they coincidentally merge with clusters of highly confident ant agents and drop their own views and directional tips for the better.

4.6 Conclusion of the chapter

This chapter addressed four aspects of this thesis. First, we described the generation of XSets. In doing so, we first described the structure of an XSet.

Then we explained how sets of primitive behaviours are manipulated in order to create the initial genetic population of XSets with which the search spaces for optimum XSets are evolved. In these explanations, we indicated how the XSets that arise are represented in the genetic population, as well as how ant agents use these XSets to perform the tasks sought.

Thereafter, the chapter described the setup of evaluation environments on which the performances of XSets are assessed. In doing so, we emphasized on describing the key design parameters, as well as those other variables that are stored in environment tuples (e.g. swarm information and target indicators).

The third aspect this chapter addressed is the quantification of emergent behaviour and determination of the indices of merits of XSets. We presented five measures of emergence which indicate the extent to which emergent behaviour is manifest as a result of using a particular XSet. First, we described speed of emergence which assesses the time it takes ant agents to converge. Then we considered the quality of emergence which establishes the frequency with which ant agents successfully arrive at their intended targets within a set time frame. Average delivery rates expressed these frequencies of arrivals as a percentage of ant agent density. We also considered average end-to-end delays which evaluates the time it takes ant agents to travel between the starting point and the target. Lastly, we built the fifth measures of emergence on Shannon's measurement theories.

Some measures of emergence are applied on samples of ant agents. We illustrated how representative sample sizes are determined, and showed through an example how the final sample size is calculated.

As a case study, the chapter administered an experiment in which we evaluated various XSets for properties with which to resolve the path finding problem. The main aim of the experiment was to search for best performer XSets for this purpose. The search relied on genetic programming processes where better and better XSets were evolved over time until the best performers were identified.

Two XSets were singled out as relatively best performers, one in the stigmergic, and another in the message passing categories. A number of observations arise from these findings. We particularly highlight the following:

- There exist particular XSets with the best properties for allowing path finding behaviour in swarms of ant-like devices. These XSets are shown in Figures 4.10 and 4.11.
- The order in which primitive behaviours are arranged in XSets has a bearing on the quality and usefulness of the XSets thereof. Thus, order is a parameter of emergence.
- Although relatively high cardinality XSets may competitively achieve good performances, these XSets often consist of redundant instructions which defeats our goal to design simple and naive ant agents.
- prominent XSets serve best when particular time limits are set. Precisely, stigmergic swarms deplete with time in simulation because the levels of pheromone that are continuously dropped on the environment would reach a point when they saturate the same environment and revert ant agents back to random wandering. On the contrary, message passing swarms gain in performances with time in simulation. They may even achieve deterministic agent movements when target vector fields develop with time.

The value of this chapter is further emphasized by a number of contributions it makes to the board of knowledge, as well as to the thesis. In particular:

- identification of best XSets for the path finding problem is a big milestone towards investigations related to the creation of other forms of emergent behaviours using the XSet method.

- The design principles with which we create partial permutations and the initial genetic population of XSets are innovative. Besides creating diverse genetic populations, these principles may inspire the design of similar XSets for industrially compatible forms of emergent behaviour.

- The measures of emergence we proposed, as well as the procedures through which each measure of emergence is determined are innovative. These measures of emergence provide a unique way of evaluating the performances of swarms of agents, as well as assessing the usefulness of specific XSets. As a result, this may inspire the development of formal agent evaluation standards in the future.

- Most researchers in the field ignore the importance of sampling and extracting representative samples. This chapter presented an innovative technique for extracting samples of ant agents for tracking when it is necessary. This is potentially a useful hint in future experiments where sampling is apparent.

- The formula we proposed for calculating the indices of merits of XSets is unique. This formula is of our own making. That alone may inspire the development of XSets assessment standards in the field as well.

In the next chapter, we emphasize on investigating the relationships between the measures of emergence we reported in this chapter.

Chapter 5

Relationships Between Measures

5.1 Introduction

Chapter 3 identified collections of candidate primitive behaviours which characterize the low level activities of stigmergic and message passing ant agents. The primitive behaviours we identified were stated as lemmas because we neither verified nor validated their influences to emergent behaviour (see lemmas 1 to 10 in section 3.2.4 and in section 3.2.5). However our assumption in Chapter 3 and Chapter 4 remained that these ant agent activities are problem independent.

Mechanisms in which the primitive behaviours we identified are grouped together into XSets and stored into collections (populations) of XSets were investigated in Chapter 4. The collections of XSets that emanate from partial permutations on the set of primitive behaviours U (see section 4.2.1) were evaluated for abilities to allow deliberate formation of a particular class of emergent behaviour (as proof of concept) - the path finding behaviour. These evaluations were based on five measures of emergence that are all inspired by related works in the literature (see section 4.4).

A mechanism was defined in which measures of emergence are put together in order to determine the index of merit of each XSet in the search space, an index which indicates the extent to which emergent behaviour is manifest as a result of using a particular XSet. In the end, Chapter 4 singled out the best performer XSets in each category (those that yield relatively best indices of merit for the path finding problem).

Figure 5.1 positions the research problem of this chapter in the context of this thesis. Understanding the relationships that exist between the measures of emergence that arise when a particular XSet is used as an ant agent language for engineering predictable emergent behaviour is the critical goal of this chapter. Besides validating the XSets we study, tests that are conducted in this chapter verify different primitive behaviours as useful constructs for programming ant agent behaviours. Such knowledge of the relationships that exist between measures of emergence may potentially allow us to deliberately engineer other forms of emergent architectures with practical benefits to human life.

5.1.1 Problem statement

The desire to establish the relationships that exist between the measures of emergence that arise when particular XSets are used for resolving the path finding problem (as a case study) drives the research we present in this chapter. We particularly address the following question:

- What relationships exist between the sets of measures of emergence that arise when swarms of ant agents use particular XSets to resolve the path finding problem? In responding to this question, we particularly establish the correlation coefficients that arise between pairs of sets of measures of emergence that are recorded at different control levels.

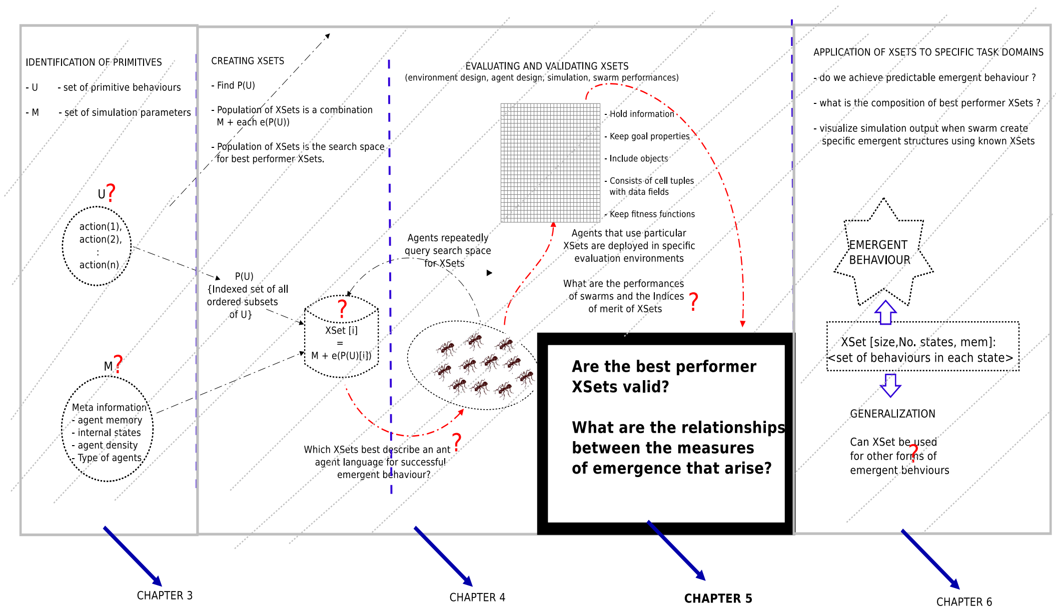


Figure 5.1: Positioning the research task of chapter 5 in the thesis

We defined control levels as intervals in simulation or stages in simulation when we extract swarm performances (see section 4.5.2). Besides establishing correlations, we also compare the means and variances that arise in the indices of merits that are yield in these sets of measures of emergence.

Our null hypothesis in correlation measures is that there are no relationships between the sets of measures of emergence that arise. In a layman's language, we are saying that the measures of emergence we see are merely a random phenomena with no basis in the XSets we use.

We also hypothesize that the variances we see in the sets of measures of emergence are similar (no difference between variances), and that the mean performances are also similar. The motivation for these choices of null hypotheses stems from known propositions in statistics that one can only falsify statements but cannot not prove them (Lohr, 2010). Thus, a logically consistent statement is one which tests for evidence to refute the claim we make. If we get to see some evidence of relationships, some evidence of differences in variances, or differences between means in the sets of measures of emergence, that alone would falsify the null hypothesis is favour of the alternative hypothesis.

Note that the problem we address in this chapter is not to compare the results that are achieved from different ant system metaphors (stigmergic versus message passing, or other traditional models). Rather, we compare the trends we observe in the measures of emergence that are recorded at different control levels when we use the categories of ant agents we study.

5.1.2 Overview of the chapter

The following is a breakdown of the sections of this chapter, all of which are collectively aimed at providing a solution to the research question of this chapter (see section 5.1.1). Section 5.2 describes the procedure we follow in order to determine correlation coefficients between different sets of measures of emergence, as well as the procedure we follow in order to determine the significance of the results thereof. The same section also highlights the key statistics we want to find and interpret when we compare variances and mean performances in two independent sets of measures of emergence.

The biggest part of this chapter presents the results we achieve regarding the relationships that arise (section 5.3). Conclusions close the chapter in section 5.4, along with highlights of the main contributions of this chapter to the board of knowledge and the thesis.

5.2 Determining relationships between measures

The key input to this chapter are the XSets we reported in Chapter 4 as relatively best control dictionaries for deliberate path finding behaviour. However are the best performer XSets valid toolboxes? Are the measures of emergence that are reported at different control levels related in any way to indicate that these XSets are influencing coherent behaviour over time?

In this thesis, the validity of an XSet is determined by the degree of relationships we observe between sets of measures of emergence that are achieved as a result of using the XSets. These measures of emergence are extracted at different control levels of the same simulation.

Three groups of variables are key in the experiment we administer in this chapter. The main dependent variables (those key variables which we measure) are the average measures of emergence that are reported at different control levels. Figure 5.2 shows that these measures of emergence are accumulated over ten replications before an average measure of emergence is calculated at each control level. Thus, every value in each cell of the table that is presented in Figure 5.2 is a sum of ten scaled measures of emergence at that control level (see section 4.4.7 regarding how measures of emergence are scaled. The formulas for scaling different measures of emergence are also indicated in the table in Figure 5.2).

However the results which we statistically analyze and compare, are the averages of the measures of emergence that are reported at control levels over the ten replications. We also indicated that, from observation, ten replications (which is way above the norm of a minimum of three replications that are generally recommended for most science experiments) are sufficiently many to reveal the trends we investigate. Although other non-parametric measures can be used to get these centrally placed measures of emergence

Title: To investigate the relationships between measures of emergence that arise at different control level when a particular XSet is used for path finding purposes.

Hypothesis: the sets of measures of emergence that arise at different control levels have no relationships

Dependent variable : average measures of emergence at different control levels

Independent variables : control levels, time in simulation (maximum of 10000 iterations).

Controlled variables: agent density (5000), environment size (100 × 100), position of starting point (fixed), centre of target (centre of environment) , XSets (best performers).

Procedure - Generator functions are invoked which define path finding environments. Ant agents are distributedly deployed and allowed to score performances. Results are recorded in intervals of 1000 iterations for ten replications. Average measures of emergence are calculated and reported at each interval. The algorithm below summarizes this procedure.

```
foreach best performer XSet
   foreach swarm of 5000 ant agents
     foreach replication of experiment i
```

Control level	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
speed	$s[1]+=(1-\frac{s_i}{T})$	$s[2]+=(1-\frac{s_i}{T})$	$s[3]+=(1-\frac{s_i}{T})$	$s[4]+=(1-\frac{s_i}{T})$	$s[5]+=(1-\frac{s_i}{T})$	$s[6]+=(1-\frac{s_i}{T})$	$s[7]+=(1-\frac{s_i}{T})$	$s[8]+=(1-\frac{s_i}{T})$	$s[9]+=(1-\frac{s_i}{T})$	$s[10]+=(1-\frac{s_i}{T})$
quality	$q[1]+=(\frac{q_i}{T})$	$q[2]+=(\frac{q_i}{T})$	$q[3]+=(\frac{q_i}{T})$	$q[4]+=(\frac{q_i}{T})$	$q[5]+=(\frac{q_i}{T})$	$q[6]+=(\frac{q_i}{T})$	$q[7]+=(\frac{q_i}{T})$	$q[8]+=(\frac{q_i}{T})$	$q[9]+=(\frac{q_i}{T})$	$q[10]+=(\frac{q_i}{T})$
delivery	$d[1]+=d_i$	$d[2]+=d_i$	$d[3]+=d_i$	$d[4]+=d_i$	$d[5]+=d_i$	$d[6]+=d_i$	$d[7]+=d_i$	$d[8]+=d_i$	$d[9]+=d_i$	$d[10]+=d_i$
delays	$a[1]+=(1-\frac{a_i}{T})$	$a[2]+=(1-\frac{a_i}{T})$	$a[3]+=(1-\frac{a_i}{T})$	$a[4]+=(1-\frac{a_i}{T})$	$a[5]+=(1-\frac{a_i}{T})$	$a[6]+=(1-\frac{a_i}{T})$	$a[7]+=(1-\frac{a_i}{T})$	$a[8]+=(1-\frac{a_i}{T})$	$a[9]=+=(1-\frac{a_i}{T})$	$a[1o]+=(1-\frac{a_i}{T})$
Shannon	$I[1]+=(\frac{I_i}{3})$	$I[2]+=(\frac{I_i}{3})$	$I[3]+=(\frac{I_i}{3})$	$I[4]+=(\frac{I_i}{3})$	$I[5]+=(\frac{I_i}{3})$	$I[6]+=(\frac{I_i}{3})$	$I[7]+=(\frac{I_i}{3})$	$I[8]+=(\frac{I_i}{3})$	$I[9]+=(\frac{I_i}{3})$	$I[10]+=(\frac{I_i}{3})$

```
  until 10 replication
end swarm

Find average measure at each control level by dividing sum by 10
```

Figure 5.2: Experiment design

(such as mode and median), averages have the advantage of considering all the scores that are achieved in the ten replications - thus giving a better indication of the central tendencies thereof.

More over, although it is possible to extract results from more control levels (both in between the control levels we propose or after the cut-off simulation period), these are continuous stochastic processes in which one can only take a representative sample of the result. In line with this thought, and from observation, ten control levels in 10000 iterations are sufficiently many to reveal the trends we investigate.

Time in simulation is the key independent variable in this experiment (the variable we manipulate and monitor), and is measured in iterations. We explained in chapter 4 that an iteration is equivalent to an ant agent step. Precisely, we monitor the performances of swarm at specific control levels or time intervals in simulation. The rest of the variables we require in this chapter are controlled (those variables that are kept constant).

Of the key controlled variables are agent density and environment size. We explained how users can change these two variables at run-time. For illustration purposes, we continue to deploy swarms of ant agents in colonies of 5000 on 100×100 grid environments (merely to prove the concepts we investigate).

The positions of targets and starting points are kept at fixed locations throughout the experiment. This is done in order to ensure fair outcomes when we compare the results. Please note that investigations aimed at assessing the effects of agent density or environment complexity to path finding are outside the scope of this chapter. In the next three sections, we discuss the statistics we propose for establishing relationships between sets of measures of emergence.

5.2.1 Correlation coefficients

As part of our investigations for relationships between sets of measures of emergence that arise when specific XSets are used, we elaborate on how we calculate sample correlation coefficients (r) and map these to the population correlation coefficients (ρ). Correlation coefficients indicate the strength and direction of linear association between two data sets, in this case sets of measures of emergence. They indicate the proportion of common variation between two data sets (Trochim, 2006.; David, 2008).

Although different correlation coefficients exist in the literature (Francis, 1990), we particularly consider Pearson's correlation coefficient because of its emphasizes on quantifying the strength of association between two data sets (David, 2008). Detailed discussions regarding the application, merits, and demerits of using other forms of correlation coefficients is outside the scope of this work.

Equation (5.1) shows how we calculate Pearson's sample correlation coefficient (r) between two sets of measures of emergence (this formula is taken from Francis (1990)). In this formula, x and y are sets of different measures of emergence that are compared at the time. Each data set, in this case, consists of ten scores (where each score is an average of ten replicated performances at each control level). The sample correlation coefficient r that arises inclusively lies in the range $[-1; 1]$ (Trochim, 2006. ; David, 2008).

The significance of the sample correlation coefficient we get, r, is recommended at a 99% level of confidence because the sample size we use is statistically too small ($n = 10$ measures of emergence in each data set, where each measure represents the average performances of ant agents at each control level). In this case, the significance level we get is $\alpha = 0.01$. Thus, this significance level is also known as the alpha value (Lohr, 2010.; Francis, 1990).

$$r = \frac{n\sum xy - \sum x \sum y}{\sqrt{\left(n\sum x^2 - \left(\sum x\right)^2\right)\left(n\sum y^2 - \left(\sum y\right)^2\right)}} \tag{5.1}$$

Variable	Meaning of variable in equation (5.1)
n	number of scores that are recorded for each measure of emergence
x	set of scores corresponding to the first measure of emergence
y	set of scores corresponding to the second measure of emergence
r	Pearson's sample correlation coefficient

Two-tailed correlation tests are recommended in this case because we do not know, in advance, the sign of the correlation coefficients we will get. Statistical tables in Francis (1990) show that a critical correlation value of 0.765 arises when such a sample size of 10 is used along with an alpha value of 0.01. This is the minimum sample correlation coefficient we can expect between two sets of measures of emergence in order to conclude that the association implied in the population correlation coefficient ρ is not by chance (Miles, 2008). We are saying that, if the absolute value of r is above 0.765, we have sufficient evidence to reject the null hypothesis which states that: there are no significant relationships between two sets of measures of emergence ($H_0 : \rho = 0$) in favour of the alternative hypothesis ($H_1 : \rho \neq 0$). Note that in this notation, ρ refers to the population correlation coefficient while r denotes the sample correlation coefficient.

5.2.2 Variances

More investigations regarding the relationships that may exist between pairs of sets of measures of emergence are required in order to validate the XSets we propose, as well as to verify the individual primitive behaviours that form the XSets. In particular, are the measures of emergence we see in each category of comparable origins?

An F-test is suitable for assessing the significance of the similarities we observe between the variances that arise in two different sets of measures of emergence (Jayaraman, 1999). For example, is the variation we see in the speed of emergence over time significantly different from the variation we see in the quality of emergence? The null hypothesis which drives the F - tests we conduct is that the variances we observe in two different sets of measures of emergence are similar ($H_0 : \sigma_1^2 = \sigma_2^2$). In this case, the alternative hypothesis is that the two data sets are not similar ($H_1 : \sigma_1^2 \neq \sigma_2^2$).

5.2.3 Means

Relationships between pairs of sets of measures of emergence are further assessed using T - tests (Jayaraman, 1999). This statistic compares the central tendencies that are observed between two independent data sets. Note that T - tests are built on F - test results, assuming equal or unequal variances upfront, depending on the outcomes observed in F-tests. The null hypothesis which motivates the T - tests is that the mean performances we observe in two sets of measures of emergence are identical (H_0: $\mu_1 = \mu_2$). Thus, the alternative hypothesis proposes unequal means (H_1: $\mu_1 \neq \mu_2$).

5.3 Results

Table 5.1 reports the weighted average measures of emergence that are achieved at the 10 control levels we sample when the best performer XSet in the stigmergic category was used for resolving the path finding task (see section 4.5.1

	Speed : $1-\frac{s_i}{T}$	Quality : $\frac{q_i}{T}$	Delivery : d_i	Delays : $1-\frac{a_i}{T}$	Information : $\frac{I_i}{3}$
1000	0.7108	0.0451	0.0902	0.9603	0.1033
2000	0.7868	0.0978	0.1956	0.9737	0.2333
3000	0.8703	0.2143	0.4286	0.9809	0.4433
4000	0.9145	0.3442	0.6884	0.9873	0.6300
5000	0.9219	0.3536	0.7072	0.9910	0.7367
6000	0.9063	0.3695	0.7390	0.9887	0.7033
7000	0.8857	0.3704	0.7408	0.9817	0.6433
8000	0.8808	0.3708	0.7416	0.9794	0.5867
9000	0.8791	0.3719	0.7438	0.9767	0.4967
10000	0.8789	0.3732	0.7464	0.9633	0.4367

Table 5.1: Weighted stigmergic measures of emergence

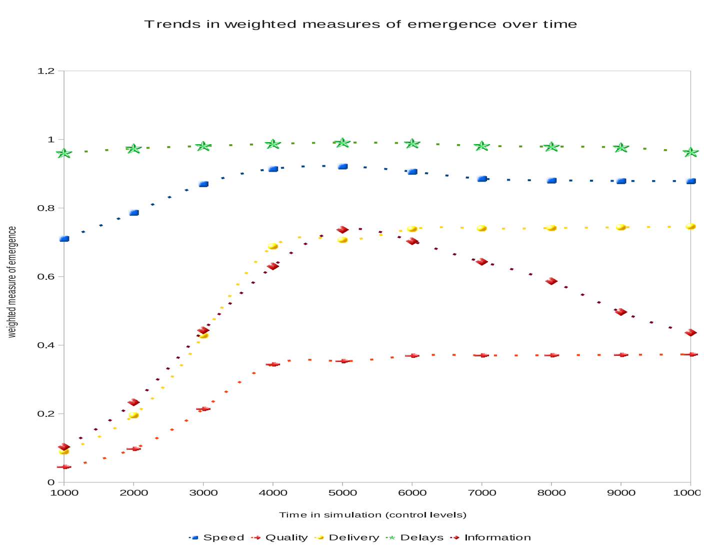


Figure 5.3: Trends in weighted measures of emergence: stigmergic XSet

	Speed : $1-\frac{s_i}{T}$	Quality : $\frac{q_i}{T}$	Delivery : d_i	Delays : $1-\frac{a_i}{T}$	Information : $\frac{I_i}{3}$
1000	0.6327	0.0213	0.0426	0.2320	0.0700
2000	0.7047	0.0353	0.0706	0.4240	0.2333
3000	0.8703	0.0677	0.1354	0.5990	0.4433
4000	0.9145	0.1534	0.3068	0.6610	0.6300
5000	0.9389	0.2734	0.5468	0.7340	0.7367
6000	0.9688	0.3599	0.7198	0.8040	0.7767
7000	0.9789	0.4133	0.8266	0.8570	0.8433
8000	0.9811	0.4418	0.8836	0.8740	0.9340
9000	0.9838	0.4756	0.9512	0.8770	0.9633
10000	0.9879	0.4809	0.9618	0.8930	0.9933

Table 5.2: Weighted message passing measures of emergence

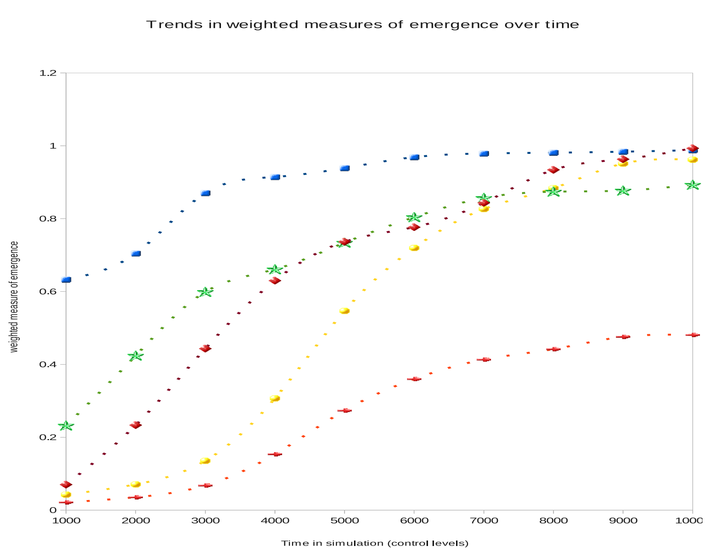


Figure 5.4: Trends in weighted measures of emergence: message passing XSet

for details regarding the formulation of the path finding problem). Figure 5.3 graphically interprets the trends that arise.

Likewise, Table 5.2 reports the weighted average measures of emergence that are recorded at each of the 10 control levels we sample when the best performer XSet in the message passing category is used for the same purpose under similar simulation conditions. Figure 5.4 graphically interprets the trends thereof.

Note that these readings are presented as weighted average measures of emergence for comparison purposes. We make the following observations from these results:

- Generally, measures of emergence that are sampled from the stigmergic category improve with time in simulation until a turning point is reached, after which the model depletes. This observation is consistent with the one we made in Chapter 4 when we analyzed merit trends (see section 4.5.3.3 for a similar observation). We observe that these turning points commonly occur when the simulation environment gets saturated with the levels of pheromones ant agents drop in every step. Environment saturation reverts ant agents into the random wandering mode. A partial remedy to this flaw would be to consider dissipation effects (see section 6.3.5 for this aspect of the thesis).

- The best achievement of every stigmergic ant agent is with respect to the average number of steps an ant agent walks between the starting point and the target (see Figure 5.3). We attribute these achievements to the mechanism in which ant agents in this category are propelled from one location to another. These ant agents favour movements away from locations which contain high levels of the pheromones they are dropping at the time. Thus, stigmergic ant agents try to avoid heading backwards. Although probabilistic path choices often derail ant agents off the emergent paths, speed of emergence is generally good (the time it takes ant agents to converge).

- Poor qualities of emergence (tendencies of ant agents to follow the emergent paths and the frequencies of arrivals of ant agents on targets) are observed early in simulation time in the stigmergic category because this model requires ant agents to have time to build shared memories through which they can interact.

- Similarly, measures of emergence that are sampled from the message passing category improve with time in simulation in a Sigmoid-like pattern (s - shaped curves). As such, ant agents in this category are generally poor in performances in the early stages of the simulation. They would improve with time, and then maintain their achievements when they converge on deterministic paths.

- Both models demonstrate relatively high uncertainty measures when ant agents make path choices. However information levels increase slowly with time. We notice that the average amount of information around an ant agent in each step is highly dependent on how popular the ant agent's current location is to the swarm.

- Time in simulation is an important parameter of emergence in both categories. Sufficient time time allows ant agents in the message passing category to sufficiently build vector fields which create deterministic paths towards targets. In these vector fields,vector propagation occurs through which searching ant agents get directional cues sooner. On the contrary, lots of time in simulation is detrimental to the stigmergic model when environments get saturated.

The rest of the sections that follow in this chapter establish the relationships that exist between the measures of emergence that are reported in these two tables, if any.

5.3.1 Tests for normality

It is hard to tell the general situation regarding the distribution of data by looking at it from the tables (for example, we cannot see from an eye scan whether the data that is presented in Tables 5.1 and 5.2 follow a normal distribution or not). Although descriptive statistics may give a picture of the central tendencies in these data sets, they often fail to detect, or they often ignore the effects of tail cases (Francis, 1990).

Kolmogorov-Smirnov tests for normality compare samples of data with the standard normal distribution and establish whether these sample distributions differ from theoretical expectations (Lohr, 2010). Precisely, these tests compare sample data with the prediction of a Gaussian distribution, thus determining the "goodness of fit" of the data sets to a normal distribution (Lohr, 2010). As a result, Kolmogorov-Smirnov tests report two critical outcomes; the d statistic and the p value, where: for all $x - axis$ values of the distribution curve, d is the maximum or biggest vertical difference (along the $y - axis$) between the predicted Gaussian distribution curve and the sample distribution curve (Lohr, 2010). Then, p is the calculated probability that the d statistic is larger in the population than is observed in the sample. If $p > 0.05$, the sample distribution passes the test for normality (Lohr, 2010).

Although other normality tests are possible (such as the Shapiro - Wilk test, or Q-Q plots), the Kolmogorov-Smirnov test for normality has a number of advantages that are in favour of the way our data is sampled and reported in this chapter. First, it is not sensitive to identical data values in the sample data sets or any ties (as in the case of Shapiro-Wilk test). We can not guarantee different data values in the scores we report in this work. In addition, transformation of data values, scaling them, or weighting the values, does not change the d statistic (Lohr, 2010). We are saying that actual data analysis is equivalent to rank analysis. Our data values are scaled to lie

Category of results	K-S d	p-value	Remark
Stigmergic speed of emergence	0.34142	p<0.2	Pass
Message passing speed of emergence	0.26103	p>0.2	Pass
Stigmergic quality of emergence	0.36351	P<0.15	Pass
Message passing quality of emergence	0.17993	p>0.2	Pass
Stigmergic average delivery rates	0.36351	p<0.15	Pass
Message passing average delivery rates	0.17993	p>0.2	Pass
Stigmergic average end-to-end delay	0.14280	p>0.2	Pass
Message passing average end-to-end delays	0.18775	p>0.2	Pass
Stigmergic Shannon's information values	0.17625	p>0.2	Pass
Message passing Shannon's information values	0.19243	p>0.2	Pass

Table 5.3: Kolmogorov-Smirnov tests for normality

within the range [0; 1] for comparison purposes, hence suited for Kolmogorov-Smirnov tests for normality. In these views, this approach is fairly robust (Lohr).

Table 5.3 reports Kolmogorov-Smirnov tests for normality on the measures of emergence that are reported in Table 5.1 and Table 5.2. We make the following three observations from the outcomes of these tests:

- All measures of emergence passed the Kolmogorov-Smirnov tests for normality because their observed p values are all greater than 0.05 (Lohr, 2010). This allows us to state that no significant departure from normality is noted in the data sets we reported. Most important in this work is that normality is a requirement for justifying the use of inferential statistics such as F- Tests and T-Tests when we determines whether there are statistically significant differences between the means and variances that are observed in pairs of sets of measures of emergence.
- We stated earlier on that the d value indicates the maximum vertical differences between the sample distribution and the predicted Gaussian

normal distribution. We observe that generally, the message passing data sets are closer to perfect normality because their d values are relatively smaller than the d values that are reported in the stigmergic category. However there are higher chances that these d values would be larger in message passing population because the p values are relatively larger.

- We also stated earlier on that one of the key advantages of Kolmogorov-Smirnov tests is that transformation of data values, scaling data, or weighting data in any way, does not change the d statistic. We observe a similar d value between the qualities of emergence and the average delivery rates in both categories. This outcome connotes that the two measures of emergence are, in fact, the same measure of emergence that is expressed in two different ways.

In the next three sections, we compare the variations we observe in pairs of sets of measures of emergence that are taken from the same category. Then we establish the similarities between the means, as well as the correlation coefficients thereof. The findings that arise from these statistics can be mapped to population statistics because the data sets we use are normally distributed.

5.3.2 Correlation coefficients

We indicated in section 5.2.1 that correlation coefficients determine the strength and direction of linear association between pairs of sets of measures of emergence. In other words, it indicates the proportion of common variation between two data sets (Trochim, 2006.; David, 2008).

We report sample correlation coefficients (r) that arise between pairs of sets of measures of emergence in each category. We stated that two-tailed correlation tests are recommended because we have no clue of the direction of

	Speed	Quality	Delivery	Delays
Quality	$r = 0.9093$			
	p=0.000			
Delivery	$r = 0.9093$	$r = 1.000$		
	p=0.000	p=—		
Delays	$r = 0.7655$	$r = 0.5349$	$r = 0.5349$	
	p=0.010	p=0.111	p=0.111	
Information	$r = 0.9421$	$r = 0.8785$	$r = 0.8785$	$r = 0.8501$
	p=0.000	p=0.001	p=0.001	p=0.002

Table 5.4: Correlations between stigmergic measures of emergence

	Speed	Quality	Delivery	Delays
Quality	$r = 0.8582$			
	p=0.001			
Delivery	$r = 0.8582$	$r = 1.000$		
	p=0.001	p=—		
Delays	$r = 0.9020$	$r = 0.9230$	$r = 0.9230$	
	p=0.000	p=0.000	p=0.000	
Information	$r = 0.9628$	$r = 0.9527$	$r = 0.9527$	$r = 0.9877$
	p=0.000	p=0.000	p=0.000	p=0.000

Table 5.5: Correlations between message passing measures of emergence

correlation that will arise. An alpha value of 0.01 and $N = 10$ give a critical correlation value of 0.765. This critical value is the minimum sample correlation coefficient we expect between two sets of measures of emergence in order to conclude that the association we see in the samples reflect the association in the population. Remember, our null hypothesis states that there are no significant relationships between pairs of sets of measures of emergence ($H_0 : \rho = 0$).

Table 5.4 reports the correlation coefficients that are yield between pairs of sets of stigmergic measures of emergence. In each case, we show the sample correlation coefficient r, as well as the probability p (a probability that this sample correlation coefficient is different in the population).

Table 5.5 then reports the sample correlation coefficients that are obtained between sets of message passing measures of emergence. Similarly, we show the sample correlation coefficient r, as well as p. We make the following observations from these two tables:

- all sample correlation coefficients in the message passing category are above the critical correlation value of 0.765. In addition, the p values we report in this category connote that there are very rare chances that the population correlation coefficients are different from these sample correlation coefficients ($r \approx \rho$ or say, $r \equiv \rho$). This piece of information provides sufficient statistical evidence with which to reject the null hypothesis ($H_0 : \rho = 0$) in favour of the alternative hypothesis.

- sample correlation coefficients in the stigmergic category are weak when average end-to-end delays are considered against quality measures. Similarly, the p values that are yield in these analyses are relatively highest (suggesting higher chances of achieving different population correlation coefficients in the same categories). We conclude that stigmergic ant agents' throughput has a weak association with the average number of step ant agents walk in simulation.

- what stands out in these two tables is that correlation values between the quality of emergence and any other set of measures of emergence are the same as the correlation values that are achieved when the average delivery rates are considered. Similarly, the p-values thereof are the same. This piece of information is consistent with the outcome we observed when we conducted the Kolmogorov-Smirnov tests for normality of these data sets. This outcome connotes that the two measures of emergence are, in fact, the same measure of emergence that is expressed in two different ways.

- Shannon's information values are significantly correlated with all the other measures of emergence in both categories. This implies that if speed of emergence is good, we can infer ant agents walking shorter distances between the starting point and the target. We can also conceive that better quality of emergence and throughput arise, suggesting the presence of sufficiently reliable information around ant agents when they path find.

- Perfect linear relationships are observed between qualities of emergence and the average delivery rates, confirming our previous observation that these two measures of emergence are the same.

- On a general note, most sample correlation coefficients are commonly above the critical correlation value in both categories. Thus, inference and regression analysis are possible. This is a critical observation which suggests that the relationships we see between sets of measures of emergence in both categories have a very small probability that they occur by chance. As such, we have sufficient evidence to reject the null hypothesis at the 1% level of significance.

Significant sample correlation coefficients between sets of measures of emergence suggest that the proportion of common variation between these data sets is high. One data set may infer the other, indicating that these data sets are of similar origins. This outcome is consistent with the finding that these sets of measures of emergence follow a normal distribution.

5.3.3 Analysis of variances

The results we report in this section further discuss the extent to which the sets of measures of emergence that are recorded in each category are related to one another with regards to the variances we see with time in simulation.

Speed and Quality	
F	0.269
P value (one tail)	0.032
F Critical value	0.1871

Speed and Delays	
F	49.23
P value (one tail)	0.000
F Critical value	5.3511

Speed and Delivery	
F	0.067
P value (one tail)	0.0002
F Critical value	0.1871

Quality and Delivery	
F	0.25
P value (one tail)	0.026
F Critical value	0.1871

Speed and Information	
F	0.101
P value (one tail)	0.001
F Critical value	0.1871

Quality and Delays	
F	151.23
P value (one tail)	0.000
F Critical value	5.3511

Quality and Information	
F	0.377
P value (one tail)	0.081
F Critical value	0.1871

Delivery and Delays	
F	604.91
P value (one tail)	0.000
F Critical value	5.3511

Delivery and Information	
F	1.509
P value (one tail)	0.274
F Critical value	5.3511

Delays and Information	
F	0.002
P value (one tail)	0.000
F Critical value	0.1831

Table 5.6: Analysis of variances in the stigmergic category

We indicated earlier on that F- tests are suitable for comparing variations in two data sets that are normally distributed. The null hypothesis that drives

	F-ratio variance	P-value for F-tests	Levene F(1,df)	df Levene	P-value Levene
speed vs quality	3.7162	0.063708	5.54332	18	0.030107
speed vs delivery	14.8647	0.000436	14.36323	18	0.001341
speed vs delays	40.6942	0.000006	7.38563	18	0.014112
speed vs information	9.8540	0.002204	7.97414	18	0.011246
quality vs delivery	4.0000	0.051003	5.31970	18	0.033192
quality vs delays	151.2265	0.000000	22.50613	18	0.000162
quality vs information	2.6517	0.162477	1.70570	18	0.207989
delivery vs delays	604.9058	0.000000	24.56048	18	0.000102
delivery vs information	1.5085	0.549994	0.75868	18	0.395211
delays vs information	401.0017	0.000000	16.17396	18	0.000800

Table 5.7: Variance comparisons in the stigmergic category

these tests is that the variances are similar ($H_0 : \sigma_1^2 = \sigma_2^2$) - where σ represents standard deviation, and henceσ^2 represents variance.

The key outcome sought in F - tests are three values namely the F - value, the critical F - value, and the P - value (Jayaraman, 1999). If the F - value we obtain is greater than the critical F - value at a chosen level of significance, it is implied that we have sufficient statistical evidence to reject the null hypothesis ($H_0 : \sigma_1^2 = \sigma_2^2$) in favour of the alternative hypothesis ($H_1 : \sigma_1^2 \neq \sigma_2^2$) (Jayaraman, 1999). However, if the F - value we obtain is less than the critical F - value, we have no evidence to reject the null hypothesis. Thus we would rather proceed to believe that the variances we see between the two sets of measures of emergence are similar.

The P - value is an alternative decision tool regarding the relationships between the variances we see in the two data sets. If the P-value is less than the significance level (which is in this case 0.01), it is implied that we have sufficient statistical evidence to reject the null hypothesis.

Speed and Quality	
F	0.456
P value (one tail)	0.129
F Critical value	0.1871

Speed and Delays	
F	0.326
P value (one tail)	0.055
F Critical value	0.1871

Speed and Delivery	
F	0.114
P value (one tail)	0.002
F Critical value	0.1871

Quality and Delivery	
F	0.25
P value (one tail)	0.026
F Critical value	0.1871

Speed and Information	
F	0.159
P value (one tail)	0.006
F Critical value	0.1871

Quality and Delays	
F	0.715
P value (one tail)	0.313
F Critical value	0.1871

Quality and Information	
F	0.348
P value (one tail)	0.066
F Critical value	0.1871

Delivery and Delays	
F	2.861
P value (one tail)	0.067
F Critical value	5.3511

Delivery and Information	
F	1.392
P value (one tail)	0.315
F Critical value	5.3511

Delays and Information	
F	0.486
P value (one tail)	0.149
F Critical value	0.1871

Table 5.8: Analysis of variances in the message passing category

Table 5.6 reports the F - test results that arise between the sets of measures of emergence that are achieved in the stigmergic category when ant agents are tasked to path find. Table 5.8 reports the F - test results that arise between the sets of measures of emergence that are achieved in the message passing category when ant agents are deployed to resolve the same path finding problem. We further summarize these statistics in Tables 5.7 and 5.9 respectively.

	F-ratio variance	P-value for F-tests	Levene F(1,df)	df Levene	P-value Levene
speed vs quality	2.216140	0.251526	3.83640	18	0.065830
speed vs delivery	8.864560	0.003293	17.91674	18	0.000500
speed vs delays	3.098219	0.107397	2.80214	18	0.111426
speed vs information	6.370493	0.010977	7.28478	18	0.014682
quality vs delivery	4.000000	0.051003	8.94419	18	0.007843
quality vs delays	1.398025	0.625729	0.05510	18	0.817064
quality vs information	2.874590	0.131596	2.44271	18	0.135482
delivery vs delays	2.861179	0.133244	5.81984	18	0.026735
delivery vs information	1.391503	0.630538	0.93081	18	0.347436
delays vs information	2.056179	0.297890	1.47845	18	0.239723

Table 5.9: Variance comparisons in the stigmergic category

We make the following observations from these outcomes:

- In four stigmergic cases, we observe F - values that are lower than the critical F - value: comparison between speed of emergence and average delivery rate, speed of emergence and information value, average delivery rate and information value, as well as between average end-to-end delays and information. We therefore have no evidence to reject the null hypothesis in these sets and rather proceed to believe, at a 1% levels of significance, that the variances we see between these pairs of sets of measures of emergence are similar.

- Six stigmergic cases achieved F - values that are greater than the corresponding critical F - value at a 1% level of significance or their P values are less than or equal to the alpha value. We therefore have sufficient statistical evidence to reject the null hypothesis ($H_0 : \sigma_1^2 = \sigma_2^2$) in favour of the alternative hypothesis ($H_1 : \sigma_1^2 \neq \sigma_2^2$).

- Two message passing cases achieved p-values that are less that the significance level ($\alpha = 0.01$) that is between speed of emergence and average delivery rate, as well as between speed of emergence and information value. These outcomes provide reasonable evidence to reject the null hypothesis ($H_0 : \sigma_1^2 = \sigma_2^2$) in favour of the alternative hypothesis ($H_1 : \sigma_1^2 \neq \sigma_2^2$).

- We fail to reject the null hypothesis in eight message passing cases, implying that the variations we observe in eight of the pairs of sets of measures of emergence in this category are not by chance.

- From a general point of view, most similarities in variations do not occur by chance. This observation suggests sufficient statistical evidence for failing to reject the null hypothesis ($H_0 : \sigma_1^2 = \sigma_2^2$). We are saying that the variations we see in the sets of measures of emergence in the message passing category show evidence of common origins. This outcome is consistent with the correlation analyses we presented earlier on.

5.3.4 Comparisons between means

The last tests we conduct are aimed at comparing the central tendencies in pairs of sets of measures of emergence that are recorded in each category. We indicated earlier on that T- tests are suitable for this purpose (comparing means). The null hypothesis which drives the T - tests we conduct is that there are no significant differences between the mean measures of emergence we observe (H_0: $\mu_1 = \mu_2$).

Speed and Quality	
P (T <=t) one-tail	0.0000
t Critical one-tail	2.552
P (T <=t) two-tail	0.0000
t Critical two-tail	2.878

Speed and Delays	
P (T <=t) one-tail	0.0000
t Critical one-one tail	2.552
P (T <=t) two-tail	0.0000
t Critical two-tail	2.878

Speed and Delivery	
P (T <=t) one-tail	0.0015
t Critical one-one tail	2.552
P (T <=t) two-tail	0.003
t Critical two-tail	2.878

Quality and Delivery	
P (T <=t) one-tail	0.0021
t Critical one-one tail	2.552
P (T <=t) two-tail	0.004
t Critical two-tail	2.878

Speed and Information	
P (T <=t) one-tail	0.0000
t Critical one-one tail	2.552
P (T <=t) two-tail	0.000
t Critical two-tail	2.878

Quality and Delays	
P (T <=t) one-tail	0.0000
t Critical one-one tail	2.552
P (T <=t) two-tail	0.0000
t Critical two-tail	2.878

Quality and Information	
P (T <=t) one-tail	0.0064
t Critical one-one tail	2.552
P (T <=t) two-tail	0.013
t Critical two-tail	2878

Delivery and Delays	
P (T <=t) one-tail	0.0000
t Critical one-one tail	2.552
P (T <=t) two-tail	0.0000
t Critical two-tail	2.878

Delivery and Information	
P (T <=t) one-tail	0.221
t Critical one-one tail	2.552
P (T <=t) two-tail	0.441
t Critical two-tail	2.878

Delays and Information	
P (T <=t) one-tail	0.0000
t Critical one-one tail	2.552
P (T <=t) two-tail	0.0000
t Critical two-tail	2.878

Table 5.10: Analysis of means in the stigmergic category

Two important pieces of information are important when we interpret T - tests, namely the P - value and the significance level. The P - value indicates the strength of evidence in support of the null hypothesis. If the observed P

	Mean Group1	Mean Group2	T-value	df	P-value for T- tests	N1	N2	Std dev Group1	std dev Group2
speed vs quality	0.8635	0.2911	12.7672	18	0.000000	10	10	0.06529	0.12586
speed vs delivery	0.8635	0.5822	3.4214	18	0.003044	10	10	0.06529	0.25172
speed vs delays	0.8635	0.9783	-5.4929	18	0.000032	10	10	0.06529	0.01023
speed vs information	0.8635	0.5013	5.3247	18	0.000046	10	10	0.06529	0.20495
quality vs delivery	0.2911	0.5822	-3.2708	18	0.004247	10	10	0.12586	0.25172
quality vs delays	0.2911	0.9783	-17.2102	18	0.000000	10	10	0.12586	0.01023
quality vs information	0.2911	0.5013	-2.7645	18	0.012772	10	10	0.12586	0.20495
delivery vs delays	0.5822	0.9783	-4.9726	18	0.000099	10	10	0.25172	0.01023
delivery vs information	0.5822	0.5013	0.7875	18	0.441260	10	10	0.25172	0.20495
delays vs information	0.9783	0.5013	7.3504	18	0.000001	10	10	0.01023	0.20495

Table 5.11: Comparisons between means in the stigmergic category

-value is less than or equal to the significance level, it is inferred that we have sufficient statistical evidence to reject the null hypothesis (H_0: $\mu_1 = \mu_2$) in favour of the alternative hypothesis (H_1: $\mu_1 \neq \mu_2$). In this case, any conclusions to accept or reject the null hypothesis is made at a 1% level of significance because the sample size we use is statistically small (10 measures of emergence in each set).

Table 5.10 reports the T - test results that arise when we compare central tendencies in the sets of measures of emergence that are recorded in the stigmergic category. These statistics are further interpreted in table 5.11.

Table 5.12 reports the T - test results that arise when we compare central tendencies in the message passing category. Similarly, these results are elaborated in table 5.13.

Speed and Quality	
P (T <=t) one-tail	0.0000
t Critical one-one tail	2.552
P (T <=t) two-tail	0.0000
t Critical two-tail	2.878

Speed and Delays	
P (T <=t) one-tail	0.0115
t Critical one-one tail	2.552
P (T <=t) two-tail	0.023
t Critical two-tail	2.878

Speed and Delivery	
P (T <=t) one-tail	0.0058
t Critical one-one tail	2.552
P (T <=t) two-tail	0.0116
t Critical two-tail	2.878

Quality and Delivery	
P (T <=t) one-tail	0.0274
t Critical one-one tail	2.552
P (T <=t) two-tail	0.0548
t Critical two-tail	2.878

Speed and Information	
P (T <=t) one-tail	0.0222
t Critical one-one tail	2.552
P (T <=t) two-tail	0.0444
t Critical two-tail	2.878

Quality and Delays	
P (T <=t) one-tail	0.0001
t Critical one-one tail	2.552
P (T <=t) two-tail	0.0002
t Critical two-tail	2.878

Quality and Information	
P (T <=t) one-tail	0.0018
t Critical one-one tail	2.552
P (T <=t) two-tail	0.0036
t Critical two-tail	2878

Delivery and Delays	
P (T <=t) one-tail	0.1437
t Critical one-one tail	2.552
P (T <=t) two-tail	0.2874
t Critical two-tail	2.878

Delivery and Information	
P (T <=t) one-tail	0.229
t Critical one-one tail	2.552
P (T <=t) two-tail	0.458
t Critical two-tail	2.878

Delays and Information	
P (T <=t) one-tail	0.395
t Critical one-one tail	2.552
P (T <=t) two-tail	0.790
t Critical two-tail	2.878

Table 5.12: Analysis of means in the message passing category

We make the following observations from these results:

- Eight out of the ten pairs of sets of measures of emergence in the stigmergic category are in favour of the null hypothesis. We are saying

	Mean Group1	Mean Group2	T-value	df	P-value for T- tests	N1	N2	Std dev Group1	std dev Group2
speed vs quality	0.89516	0.272226	8.72211	18	0.000000	10	10	0.12593	0.18747
speed vs delivery	0.89516	0.54452	2.80345	18	0.011749	10	10	0.12593	0.37494
speed vs delays	0.89516	0.69550	2.47665	18	0.023416	10	10	0.12593	0.22166
speed vs information	0.89516	0.66239	2.15303	18	0.045126	10	10	0.12593	0.31785
quality vs delivery	0.27226	0.54452	-2.05386	18	0.054808	10	10	0.18747	0.37494
quality vs delays	0.27226	0.69550	-4.61033	18	0.000217	10	10	0.18747	0.22166
quality vs information	0.27226	0.66239	-3.34325	18	0.003619	10	10	0.18747	0.31785
delivery vs delays	0.54452	0.69550	-1.09616	18	0.287458	10	10	0.37494	0.22166
delivery vs information	0.54452	0.66239	-0.75832	18	0.458075	10	10	0.37494	0.31785
delays vs information	0.69550	0.66239	0.27020	18	0.790080	10	10	0.22166	0.31785

Table 5.13: Comparisons of means: message passing category

that besides observing strong correlations between the sets of measures of emergence in this category, as well as observing similar variations in these sets, the central tendencies in the same sets of measures of emergence are generally identical. These properties are sufficient to validate the stigmergic XSet as an appropriate dictionaries for achieving path finding behaviour in swarms of an-like devices.

- Seven out of ten possible pairs of sets of measures of emergence in the message passing category are in favour of the null hypothesis. We are similarly saying that the central tendencies we see in the message passing category are identical. Message passing sets of measures of emergence that are significantly correlated, sets which demonstrate similar variations and consistent central tendencies have common origins. These properties equally validate the message passing XSet as an appropriate control toolbox for achieving path finding behaviour in swarms of ant-like devices.

5.4 Conclusion of the chapter

The chapter mainly discussed the relationships that exist between different sets of measures of emergence that arise from using particular XSets for resolving the path finding problem. Three statistical tests are described in details namely correlation analyses, comparisons between means, and analyses of variances. These statistics mainly relied on data that was sampled over ten control levels. The following three results are outstanding in this chapter:

1. Significantly strong correlation coefficients are noted between the sets of measures of emergence that are recorded in both categories. These results justified the rejection of the null hypothesis ($H_o : \rho = 0$) in favour of the alternative hypothesis. Thus, there is a very small probability that the relationships we observe between different sets of measures of emergence when particular XSets are used for resolving the path finding problem occur by chance. Strong correlation coefficients between different sets of measures of emergence connote similar origins of these sets of measures, thus suggesting validity of the XSets that are used to generate these measures of emergence.

2. Variations in the sets of measures of emergence that are recorded in both categories showed similar trends in both categories. Similarly, sufficient evidence was noted with which we justified these similarities. This outcome indicate that there are very slim chances that the variances we see in different sets of measures of emergence occur by chance.

3. T - tests consolidated the general outcome we saw by demonstrating similarities between the mean performances that arise in the same sets of measures of emergence. Sufficient evidence was presented in this

respect, also indicating that there are slim chances that the similarities we observe between these mean performances occur by chance.

The value of this chapter is further emphasized by the following contributions:

1. To the best of our knowledge, correlation analyses, comparisons between means, as well as analyses of variances are innovative and novel approaches for validating ant agent XSets. Similar validation tools can be used in other agent control architectures in the field.

2. The validation processes we present in this chapter can be followed when we want to compare different agent models with regards to the quality of the outcome they achieve.

In the next chapter, we extend the application of the path finding XSets to the multiple target problem, primarily investigating the effects of manipulating some of the variables we stated as controlled variables in this chapter.

Chapter 6

Multiple Targets Location

6.1 Introduction

Ten discrete candidate primitive behaviours of foraging ant-like agents were proposed in Chapter 3. Inspired by related works in the literature, each of these candidate primitive behaviours was described and designed in algorithmic form in order to depict the implementation issues of the primitive behaviour in computational terms.

Chapter 4 went on to present mechanisms in which these discrete primitive behaviours are put together into useful collections which, in this work, are called XSets. Collections of XSets are proposed which define genetic populations and search spaces for best performer XSets for particular purposes. Thus, the concepts of XSets, notation, and the structure of XSets, mechanisms in which XSets are created and evaluated, motivation for using the XSets method, techniques for creating diverse initial genetic populations of XSets, representation and storage of XSets in genetic populations, mechanisms in which ant agents use XSets, as well as mechanisms in which new genetic populations of XSets are evolved over time were discussed in details in Chapter 4.

To augment the discussions thereof, Chapter 4 went on to design and administer an experiment in which we generated the initial genetic population of XSets in the stigmergic, message passing, and hybrid categories with the goal of identifying the XSets in these genetic populations with abilities to allow swarms of ant-like agents to deliberately engineer desired emergent behaviour - particularly addressing a case study scenario of path finding swarms. Mechanisms were put in place with which to quantify the extent to which emergent behaviour is manifest (path finding) as a result of using each particular XSet in the genetic population at the time. These evaluations (association of indices of merits to each XSet in the genetic population) formed the basis for selecting parent XSets for crossover operation, mutation, or promotion .

XSets in different generations were successfully ranked using indices of merits and parents XSets were successfully chosen whenever they were required. At the end of the genetic evolution limit, particular XSets were identified which best describe languages for allowing swarms of ant agents to deliberately engineer predictable emergent behaviour. The composition of of these best performer XSets were explicitly stated, as well as reports regarding the performances of swarms of ant agents that used these XSets.

Identification of particular XSets as best performer XSets motivated the work that is presented in Chapter 5. Precisely, the goal of Chapter 5 has been to statistically verify the relationships that exist between the sets of measures of emergence that were reported when these best path finding XSets were used. It seeks to evaluate and test sets of measures of emergence for normality, validity, and reliability.

We demonstrated that all sets of measures of emergence that were reported

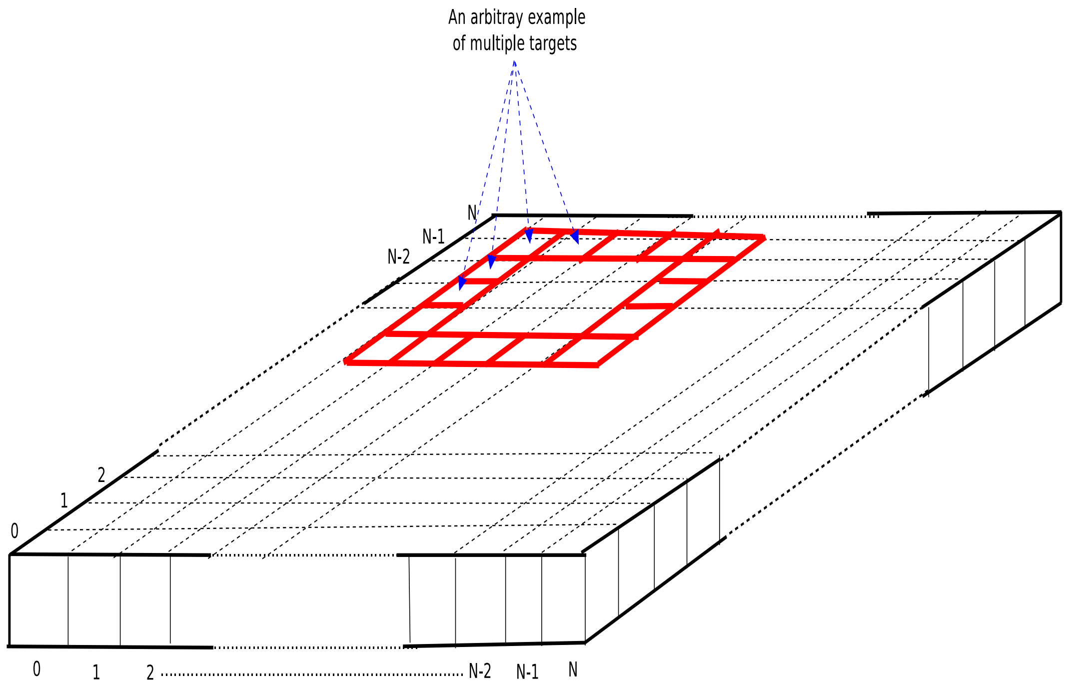


Figure 6.1: The concept of multiple targets

from the experiment that was administered in Chapter 4 passed tests for normality (see section 5.3.1). The same sets of measures of emergence demonstrated clear correlation between pairs of sets of measures of emergence that were extracted from the same category (see section 5.3.2). These measures also demonstrated similar variations (see section 5.3.3) and common mean performances over time (5.3.4). Collectively, Chapter 5 validated and justified the results thereof.

The purpose and motivation for the work that is presented in this Chapter (Chapter 6) emanates from the validations that are presented in Chapter 5. Having passed normality tests, we seek to investigate the properties of the same best path finding XSets for abilities to solve emergent problems beyond path finding. Precisely, can ant agents under the control of these best path finding XSets achieve other forms of emergent behaviour? Our hope is that, if these XSets can be adapted to deliberately cause other forms of emergent assemblies, then these XSets can potentially form the basis and building blocks of ant agent languages for deliberate engineering of other emergent object assemblies in the future.

We particularly assess the abilities of best path finding XSets for allowing another case study scenario of emergent behaviour, particularly multiple targets location. In this context, multiple targets are collections of adjacent cells of the environment in which target indicators are set. It refers to more than one cell of the environment containing target indicators, cells where ant agents would acknowledge successful target search and flip from the seek mode to the return mode (see Figure 3.4 for a detailed illustration of the concept of ant agent internal states). Therefore, in these cases, some of the fields of the tuples that hold information at these cells are designed to record the concentration of target indicators. Figure 6.1 illustrates an arbitrary setup and example of our contextual view of the arrangement of multiple targets.

In Chapter 4, we also administer experiments which investigate the effects of different variables that were controlled in previous experiment (variables that were kept constant). First, we assess the effects of varying agent densities to the indices of merits of best performer XSets. We also assess the influences brought about by including or excluding each primitive behaviour in these XSets. Likewise, an experiment is administered which evaluated the effects of altering the sequences in which primitive behaviour are arranged in these best controller XSets. The last experiment assesses the extent to which pheromone dissipation controls enhance emergent behaviour in the stigmergic category.

6.1.1 Problem statement

The question which drives the research and the experiments we administer in this chapter is the desire to investigate the extent to which best path finding

XSets can allow multiple targets location as another example of a different form of deliberate emergent behaviour. We can break this question into two tasks as follows:

1. *Evaluation of best path finding XSets for allowing multiple targets location* - In resolving this task, we administer an experiment in which we evaluate the indices of merits of best path finding XSets for allowing multiple targets location. Nine different multiple targets setups are proposed. As a result, swarms of ant agents of each category are deployed in these evaluation environments and their performances are reported. These performances are recorded for future comparisons with related path finding performances. In addition, evidence of successful multiple targets location are similarly gathered using tests for normality, correlation analyses, analyses of variances, comparison between the means, as well as using visual screenshots of the successes or failures of the swarms. Our motivation for selecting the particular design of multiple targets setups is described in details in section 6.3.

2. *What aspects of best path finding XSets influence general emergent behaviour?* - This question requires us to investigate the effects of different controlled variables of the previous experiment. Precisely, it requires us to administer the four experiments referred to in the previous section.

Responses to these two tasks, presentation of the results yield from the five experiments, as well as the analyses thereof may potentially open up new research avenues in the field.

6.1.2 Overview of the chapter

The rest of the sections of this chapter proceed as follows: First, we describe the multiple targets location problem within the context of this thesis (see section 6.2 for this aspect of the study). Then we motivate for the evaluation environments which we use for testing swarms of ant agents' abilities to allow multiple targets location (in section 6.3 for this aspect of the thesis).

The bulk of the work that is present in this chapter relates to the design of the five experiments we propose, along with the results which report the indices of merits of best path finding XSets for multiple targets location (see section 6.4 for this aspect of the chapter). We close this chapter in section 6.5, highlighting the observations and its contributions to the thesis and board of knowledge.

6.2 The multiple targets location problem

The motivation for the multiple targets location problem is to challenge path finding swarms of ant agents to solve a new class of emergent problems. This problem involves deploying swarms of ant agents in specific evaluation environments where groups of adjacent cells (locations) are marked with target indicators as a region of goals. Each of these adjacent cells (locations) define a region at which ant agents can acknowledge successful target search by flipping to the return internal state (see Figure 3.4 for a detailed illustration of the concept of ant agent internal states). In our context, ant agents acknowledge successful target search by leaving yellow marks on these regions of cells that contain target indicators before they commence return trips (see Figure 4.13 for this aspect of the work).

Ant agents are similarly deployed in the default seek mode (see Figure 3.4 for a detailed illustration of the concept of ant agent internal states). Likewise, initial placement of ant agents into the environment is random (distributed) in order to avoid the emergence of undesired local maximas in which clusters of ant agents may get trapped on sub-optimal solutions.

We indicated in the previous experiment (see section 4.5.2 for this aspect of the thesis) that the placement of the starting point and that of the centre of the region that is occupied by multiple targets are hard-coded at fixed positions on the environment in order to achieve fair experimental outcomes when we compare the performances of swarms of ant agents that used different XSets (see section 4.3.3 for this aspect of the study). We also indicated that variation of the position of the starting point and that of the centre of the region that is occupied by multiple targets is not a subject of study in this work since environment complexity is not an agent level parameter of emergence.

In this chapter (meaning Chapter 6), the goal of the swarms of ant agents that are deployed under the control of each best path finding XSet is to locate these multiple targets, and score performances in this regard. Upon hitting the regions of multiple targets, ant agents would flip to the return internal state (see Figure 3.4 for a detailed illustration of the concept of ant agent internal states) and commence return trips towards the starting point. On arriving at the starting point, the same ant agents would flip back to the search internal state and commence the search trips all over again. Ideally, these up and down movements between the starting point and the region of multiple targets are repeated for as many times as possible within the set time frame.

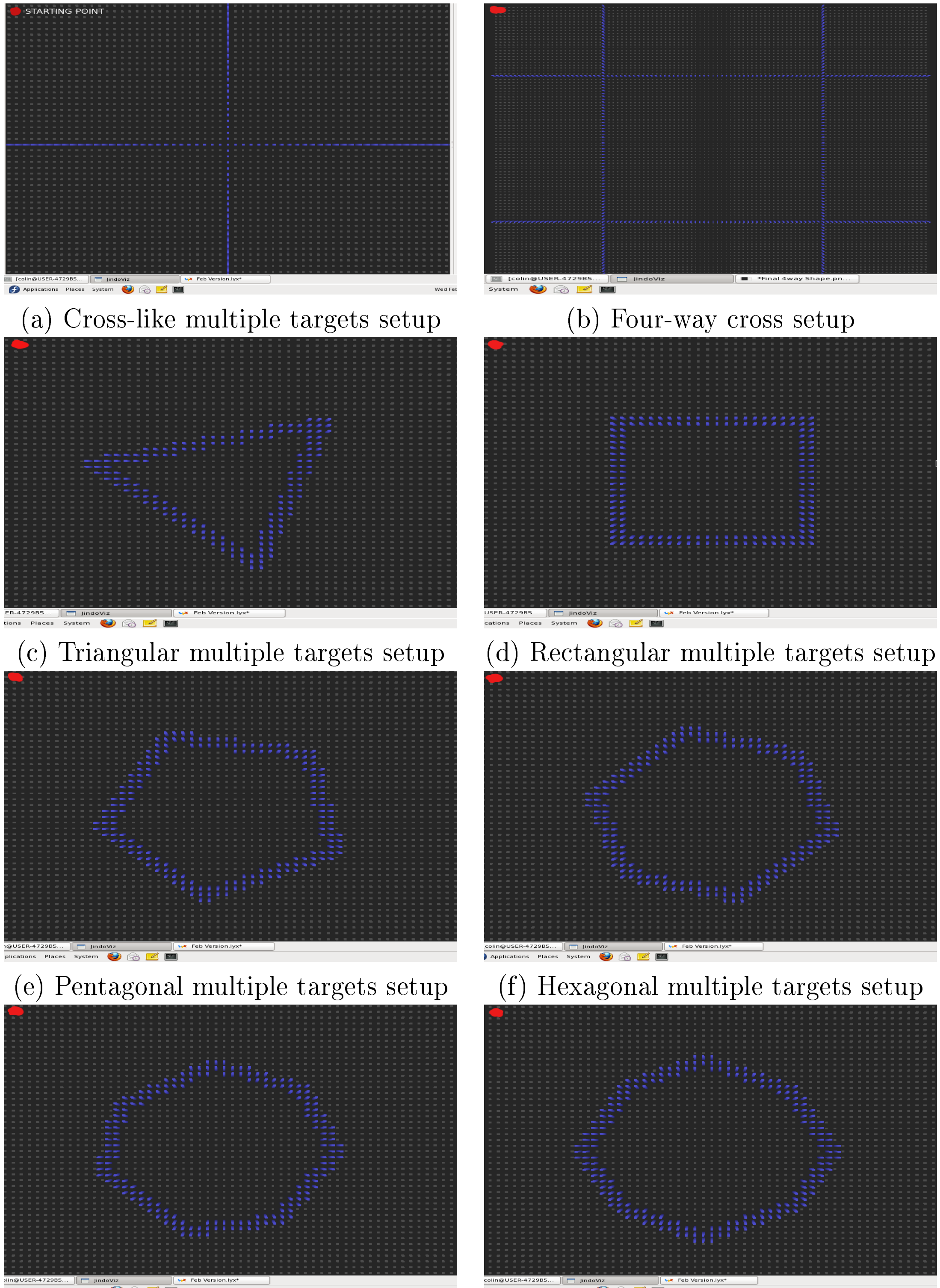
The design of the ant agents is not changed regarding memory (see section 3.2.4.1), internal states (see section 3.2.4.2), and any other abilities (see section 3.2.5). These ant agents are neither aware of the locations of the multiple targets, nor are they aware of the global outcome that would arise as the emergent behaviour of the swarm. They neither have any clue of the direction towards the multiple targets, nor knowledge of the direction towards the starting point. Each ant agent continues to use local information to decide on any actions thereof.

6.3 Definition of environments

The choices of the design and setups of multiple targets on the evaluation environments is motivated by various works in the literature. We cite a number of examples of similar scenarios where multiple targets have been layed down in triangular setups, rectangular setups, or many other multiple targets setups. Thus, the aim of this section is to justify that this is not the first time laying out multiple targets in the form of geometric shapes has been considered. For example, self-assembling shapes that are based on circle growth principles have been proposed for locating cross-like multiple targets (Nagpal et al., 2002) (see Figure 6.2(a) for the setup of cross-like multiple targets).

Similarly, scaffolding DNA origami has successfully been used to locate and mark triangular, rectangular, and even star shaped multiple targets setups (Rothemund, 2006). More polygonal multiple targets have also been reported in the work of Werfel (2002). We refer the reader to the following sources for more examples of works in which multiple targets are laid out in geometric shapes: Kaewkamnerdpong et al. (2007), Seevinck and Edmonds (2008), Mason (2002), Burke and Kendall (1999), Parrish et al. (2001), Couzin and Franks (2002), Butera (2002), Eyiyurekli et al. (2013), Bai et al. (2008), Kondac (2003), and Nagpal (2006).

Inspired by these many examples of multiple targets setups, we propose nine arbitrary case study scenarios in which multiple targets are arranged in the



(a) Cross-like multiple targets setup

(b) Four-way cross setup

(c) Triangular multiple targets setup

(d) Rectangular multiple targets setup

(e) Pentagonal multiple targets setup

(f) Hexagonal multiple targets setup

(g) Heptagonal multiple targets setup

(h) Octagonal multiple targets setup

Figure 6.2: Multiple targets setups

form of cross structures, four-way cross structures, triangles, rectangles, polygonal structures, as well as circular shapes. These nine case study scenarios are selected as merely proofs of concept. From observations, these are a sufficiently large sample of examples from which the properties sought in the XSets we study can be revealed.

Figure 6.2 shows the arrangements of multiple targets in eight of the evaluation environments we propose. The blue dots we see in these screen shots indicate the cells which contain target indicators (where ant agents would leave yellow spots when they detect target indicators). Note that the number of targets that are involved in each design decision are calculated differently depending on the shapes that arise when the continuous regions of targets are marked. That numbers depends on the radius of the region of targets, R, the number of sides of the shape, n, the length of each side, s, as well as the width of the edges of the regions of targets, w. We mathematically deduce the number of targets on each polygonal region to be $\frac{1}{2}nsw$. However the number of targets arranged in a cross shaped pattern are ($length + width$ of environment), while that of the targets arranged in a four-way cross shaped pattern are $(2 \times (length + width) - 4)$. In circular regions, there are $\pi(w^2 - 2Rw)$ targets.

6.4 Experiments and Results

The main hypothesis which drives the research and experiments we administer in this section is that there are no significant relationships between the sets of measures of emergence that are achieved when best XSets are used for path finding purposes, and the sets of measures of emergence which arise when the same XSets are used for multiple targets location ($H_0 : \rho = 0$), where ρ is the population correlation value. This hypothesis suggests that the sets of measures of emergence we see in these two case study scenarios connote path finding XSets as problem specific.

In the event of H_0 being tested and falsified, we further hypothesize that there are neither significant differences between the variations we see in the same data sets ($H_0 : \sigma_1^2 = \sigma_2^2$), nor are there significant differences between the mean performances thereof ($H_0 : \mu_1 = \mu_2$). The motivation for these choices of null hypotheses stems from known propositions in statistics that one can only falsify statements but cannot prove them - hence the negativity we impose in the formulation of hypotheses (Lohr, 2010).

To test this hypothesis, we assess the relationships that exist between sets of measures of emergence that are achieved when the best performer XSets are used for path finding behaviour, with the sets of measures of emergents that arise when the same best XSets are used for multiple targets location. Precisely, we investigate the correlation coefficients between the data sets that arise in these two scenarios. We also assess similarities in variances and means. Like in the previous experiments, the validity of the results we report rely on replicated simulations of the same experiments for a number of times in order to report centrally placed performances.

Note that our concern in this work is not to compare the outcomes of swarms with regard to the quality of the product or throughput as emphasized on in traditional ant based models. Rather, we are concerned with the reverse aspect of ant simulations, investigating the sources of emergency. We want to demonstrate that there exist explicit collections of primitive behaviours (that are used at ant agent individual levels) which describe a language for programming ant-like devices towards deliberate and predictable emergent behaviour. The findings sought are the identification of explicit XSets and their composition (not the performances of swarms of ant agents as in the traditional cases). These investigations are at preliminary stages (defining baseline studies upon which more researches on the XSets methodology would arise) such that any evidence of an understanding of the origins of emergency is sufficient (hence the reason why this work is not compared to traditional techniques and outcomes).

6.4.1 Experiment 1: Multiple Targets Location

The two key questions we respond to in this experiment are as follows: (a) are path finding XSets problem specific? (b) Can path finding XSets allow multiple targets location? In answering these two questions, our goal is to further understand the properties of path finding XSets, properties which give rise to deliberate and predictable emergent behaviour. We assess these properties on a different problem domain which involves multiple targets location. To the best of our knowledge, this is the first time explicit XSets have been investigated for these properties on such a problem task. The procedure we follow in order to answer the questions posed in this section is summarized in the experiment design that is presented in Figure 6.3 in the next section.

6.4.1.1 Experiment Design

The purpose of this experiment is to compare the sets of indices of merit that are reported in Table 4.2 with the sets of indices of merit that arise when best path finding XSets are used for multiple targets location. Our hypothesis is that there are insignificant relationships between these data sets ($H_0 : \rho = 0$). The motivation for this choice of null hypothesis has been explained to stem from known propositions in statistics that one can only falsify claims but cannot not prove them (Lohr, 2010).

Three classes of variables arise in this experiment. The main dependent variable (the variable we measure in the experiment) are the indices of merit

Title: To compare indices of merits that are achieved during path finding, to those that are achieved when the same best XSets are tested for allowing multiple targets location.

Hypothesis: there are no relationships between the sets of indices of merits that are observed during path finding and those that arise when the same XSets are tested for allowing multiple targets location.

Dependent variable : average indices of merits that are achieved at the control levels we consider.

Independent variables : time in simulation, targets setups, control levels, number of replications.

Controlled variables: agent density (5000), environment size (100 × 100), position of starting point (fixed), centre of the region of targets (centre of environment), XSets (best path finding performers), dissipation rates (0%), radius of region of targets (fixed), sequence of primitive behaviours in XSets (fixed), number of perceivable levels of pheromone (max. of 5), number of perceivable vectors (max. of 4), internal states(search, return, on target, on starting point).

Procedure - Generator functions are invoked which load the parameters of the selected evaluation environment. The region of targets is initialized with target indicators. Starting point is set. An XSet is selected between the stigXSet and msgXSet. Swarms of ant agents are deployed at random locations on the environment. Performances are scored and reported at specific control levels. Average indices of merits are computed over ten replications. These average indices of merits are the basis for statistical comparisons sought . The algorithm below summarizes this procedure.

```
foreach multiple targets setup
   initialize environment parameters
   foreach category of XSets
     repeat
       deploy a swarm of 5000 ant agents
       score performances as follows:
```

Control level	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
speed										
quality										
delivery										
delays										
Shannon										
cumm. IOM	$x_1+=IOM_{1,i}$	$x_2+=IOM_{2,i}$	$x_3+=IOM_{3,i}$	$x_4+=IOM_{4,i}$	$x_5+=IOM_{5,i}$	$x_6+=IOM_{6,i}$	$x_7+=IOM_{7,i}$	$x_8+=IOM_{8,i}$	$x_9+=IOM_{9,i}$	$x_{10}+=IOM_{10,i}$

```
    until 10 replications
  next category
 next multiple targets setup
end experiment
```

Figure 6.3: The design of experiment 1

of XSets when ant agents are tasked to locate multiple targets. These indices of merit are required for comparisons with the results reported in Chapter 4 in Table 4.2. We also require these sets of indices of merit for statistical analyses including correlation tests, analyses of variations, as well as comparisons between means for validation purposes.

Time in simulation remains the key independent variable of the experiment (the variable we manipulate and monitor). We indicated that time is measured in iterations (see Figure 6.3 for the period that is allowed for this variable). In this experiment, we also manipulate and monitor multiple targets setups (see Figure 6.2 for these setups), control levels (see Figure 6.3 for the control levels we monitor), and the number of replications that are administered on each experiment in order to achieve centrally placed indices of merit at different control levels.

A number of controlled variables are also required. On top of the list is the agent density (which remains set at a default level of 5000 ant agents in order to achieve comparable results to the results reported in Chapter 4 in Table 4.2). We also keep the environment size controlled (which remains set as a 100×100 grid). The position at which the starting point is set, as well as the centre of multiple targets remain controlled as well.

Another controlled variable of this experiment is the configuration of the XSets we use. We repeatedly use the best path finding XSets for all the tests we administer with the goal of verifying their application on different problem domains. Pheromone dissipation effects in the stigmergic category are also kept constant (at 0%), the same way as agent density and the sequence of primitive behaviours in the XSets. However the effects of most of these controlled variables are investigated in upcoming experiments in this Chapter.

Upon running this experiment, we first invoke environment generator functions which incorporate target indicators in specific cells of the environment.

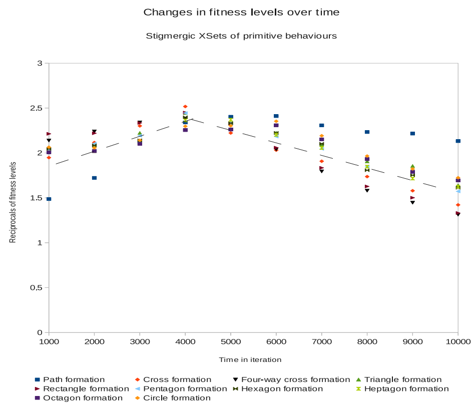
The same generator functions handle the placement of the starting point. Best path finding XSets are inputs to this experiment, which control the actions of ant agents over time. Precisely, swarms of 5000 ant agents are deployed in selected evaluation environments in order to score performances for allowing specific multiple targets location. Similarly and in line with the setup of the experiment administered in Chapter 4 Section 4.5.2, the scoring time is limited to an arbitrary period of 10000 iterations, and measures of emergence are extracted in every 1000^{th} iteration. We indicated earlier on that this is a sufficiently long simulation period to reveal the properties sought in the XSets we study.

We also repeat the same experiment for ten replicated simulations in order to report indices of merit that are averaged over ten trials. In each case, standard deviations are tracked which justify the reliability of the outcomes we report. These averaged indices of merit that are reported at different control levels are the results we statistically compare with the results reported in Chapter 4 in Table 4.2.

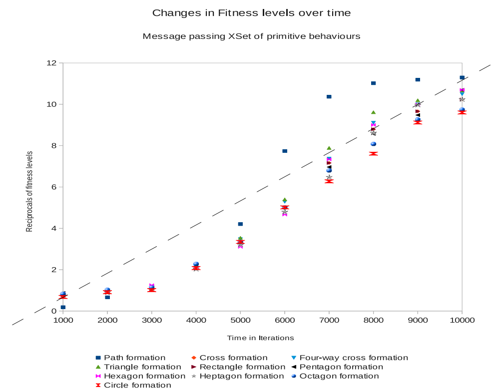
6.4.1.2 Results from experiment 1

Figure 6.4(a) plots the average indices of merit that are achieved when the best stigmergic XSet was used for allowing multiple targets location on various evaluation environments. Similarly, Figure 6.4(b) plots the average indices of merit that are achieved when the best message passing XSet was used for the same purposes.

Of interest in these results are the general trends that arise in the changes we observe in the average indices of merit over time. An overlapped high level trend curve shows that best performer XSets in the stigmergic category allow swarms of ant agents to improve in performances early in simulation time until a threshold turning point is reached, after which the model starts



(a) Stigmergic XSets



(b) Message passing XSets

Figure 6.4: Average indices of merits

to degrade. We indicated the reason for this trend in section 4.5.3.3 as related to the levels of pheromones that are dropped onto the environment reaching a point when they saturate the environments.

On the contrary, best performer XSets in the message passing category consistently show upward trends, which indicates that the XSet successfully propels swarms of ant agents towards deterministic paths. We explained the reason for these upward trends to be associated with the emergence of deterministic vector fields (see section 4.5.3.3 for similar observations).

Correlation analyses in our context, quantify the strength of association between two data sets (Francis, 1990). Our null hypothesis in these analyses is that there are no relationships between the sets of indices of merit that are reported in this experiment and the sets of indices of merit that are reported in Chapter 4 in Table 4.2.

	Cross	Four way cross	Triangle	Rectangle	Pentagon	Hexagon	Heptagon	Octagon	Circle
StigPath finding	0.033	0.175	0.177	0.169	0.118	0.128	0.188	0.264	0.242
MsgPath finding	0.972	0.969	0.961	0.957	0.960	0.965	0.963	0.970	0.975

Table 6.1: Correlation coefficients

We similarly choose an alpha value of $\alpha = 0.01$ because the samples we use of 10 outcomes (taken from the 10 control levels) are statistically small. Two-tailed correlation tests are recommended because we do not know the signs of the correlation coefficients we will get. Francis (1990) gives the critical correlation value of 0.765 when $N = 10$ and $\alpha = 0.01$. This is the minimum correlation coefficient we can expect in order to conclude that the correlation coefficients we get are not by chance.

Table 6.1 reports the correlation coefficients we get when we compare the indices of merit reported in Table 4.2 and the indices of merit that are achieved when best XSets in each category are used for multiple targets location. For example, the value of 0.033 which is reported in the second row and the second column of Table 6.1 indicates the correlation coefficient which arises between sets of indices of merit that are report in Table 4.2 under the stigmergic category, and the sets of indices of merits that arise when swarms of stigmergic ant agents are deployed in an environment where multiple targets are arranged in the form of cross structures (see Figure 6.2 for the setup of environments in which targets are set in a cross shaped pattern).

The correlation coefficients we observe in the stigmergic category are all below the critical correlation value of 0.765. This observation implies that, in this case, we fail to reject the null hypothesis that there are no significant relationships between the indices of merit that arise when stigmergic XSets are used for path finding purpose and the indices of merit we observe when the same XSets are used for multiple targets location. This outcome suggests that any relationships we may see between these data sets are merely by chance, connoting stigmergic XSets as problem specific or rather, specialist XSets.

We can explain the outcome in the previous paragraph to be associated with the way in which shared memories arise in this category. Precisely, placement of specific levels of pheromones on the environment when swarms of ant agents path find promotes the emergence and development of narrow but highly attractive paths between the starting point and the target. On the contrary, wider and weaker paths arise when multiple targets are considered because ant agents have more path options towards evenly accessing the multiple targets within the set time limits. We conclude that the indices of merit of stigmergic XSets are sensitive to the number of the multiple targets, as well as sensitive to the radius of the region where multiple targets are placed. In this case, few and closely packed multiple targets allow the emergence of narrower and stronger paths that would share relationships with the outcomes of path finding XSets in Table 4.2.

However although the findings that are reported in these observations connote weak relationships between path finding outcomes and the performances that are achieved during multiple targets location in the stigmergic category, we cannot make any conclusive remarks regarding these relationships since this is only one statistical outcome which is not sufficient evidence in to accept or refute the overall hypothesis.

The correlation coefficients we observe in the message passing category are all above the critical correlation value. Statistically, we have sufficient evidence to reject the null hypothesis in favour of the alternative hypothesis (significant relationships are observed between pairs of sets of indices of merit in the message passing category).

	Cross	Four way cross	Triangle	Rectangle	Pentagon	Hexagon	Heptagon	Octagon
Four-way cross	0.967							
Triangle	0.963	0.915						
Rectangle	0.968	0.998	0.927					
Pentagon	0.976	0.939	0.996	0.948				
Hexagon	0.952	0.925	0.985	0.933	0.989			
Heptagon	0.937	0.901	0.979	0.911	0.979	0.990		
Octagon	0.874	0.832	0.946	0.839	0.936	0.962	0.968	
Circles	0.875	0.838	0.947	0.845	0.938	0.963	0.968	0.998

Table 6.2: Correlations: multiple to multiple targets (stigmergic)

We can explain the outcomes we see in the previous paragraph as follows: The ability of message passing ant agents to hold directional information in their memories allows ant agents to globally develop vector fields which they can follow and rely on unconditionally regardless of the number of targets that are set on the environment. The strength of a path an ant agent follows is held in the ant agent's memory. The ant agent's knowledge is problem independent. Overall, message passing XSets are insensitive to the number of the multiple targets set, as well as insensitive to the radius of the region where multiple targets are placed. The findings we present from the message passing category provide sufficient statistical evidence to reject the null hypothesis in favour of the alternative hypothesis. These findings connote slim chances that the relationships we see in these data sets occur by chance. Thus, message passing XSets are viewed as problem independent. However we do not make conclusive remarks, as yet, because this is just one statistical outcome which requires further investigations before we confirm the observations we make.

	Cross	Four way cross	Triangle	Rectangle	Pentagon	Hexagon	Heptagon	Octagon
Four-way cross	0.999							
Triangle	0.999	0.999						
Rectangle	0.999	0.999	0.999					
Pentagon	0.996	0.995	0.998	0.998				
Hexagon	0.996	0.995	0.996	0.997	0.999			
Heptagon	0.999	0.999	0.999	0.999	0.998	0.998		
Octagon	0.999	0.999	0.999	0.999	0.996	0.996	0.999	
Circles	0.999	0.999	0.997	0.998	0.995	0.995	0.998	0.999

Table 6.3: Correlations: multiple to multiple target (message passing)

Table 6.2 and Table 6.3 present correlation coefficients that arise between the indices of merit that are reported when we compare the outcomes of the best stigmergic XSet when it was used for multiple targets location. For example, the value of 0.999 which is reported in Table 6.3 in the second row and second column is the correlation coefficient that arises between the sets of indices of merit that are recorded when message passing swarms are deployed in environments where multiple targets are arranged in the form of crosses, and the sets of indices of merits that are recorded when message passing swarms are deployed in environments where multiple targets are arranged in the form of four-way crosses (see Figure 6.2 for these target setups).

In both cases, significantly strong correlation coefficients are observed between the sets of indices of merit in each category when best XSets are used for multiple targets location. In this case, the null hypothesis is penalized in favour of the alternative hypothesis (significant relationships exist between different sets of indices of merit that arise when ant agents work towards multiple targets location). This is the case in the stigmergic category because the setups of multiple targets are similar with regards to numbers and distribution. This outcome contradicts with our previous observation regarding stigmergic XSets being problem specific. Rather, it suggests that stigmergic XSets are sensitive to the number and spread of multiple targets but not problem specific. As a result, we also hold on to any conclusive remarks regarding the validity of the relationships we observe until we establish the outcomes of other statistical analyses.

Analysis of variances in our context, compare the variations in sets of indices of merit that are recorded in different data sets. They tell us how significantly different the sets of indices of merit in two data sets are dispersed. Sets of indices of merit whose variances are significantly different imply that the XSets thereof are problem specific. They imply that the indices of merit we compare are not of similar origins. Therefore the null hypothesis we propose in these tests states that there are no significant differences between the variances we observe in sets of indices of merit that are achieved under different problem domains. The motivation for this hypothesis similarly arise from known statistical propositions of being able to falsify rather than prove claims (Lohr, 2010).

Similarly, an alpha value of $\alpha = 0.01$ is used because the samples we use are statistically small (taken from the 10 control levels). We indicated in Chapter 5 that F - tests are suitable for these comparisons.

Table 6.4 presents the F - test outcomes that are yield between the sets of indices of merit that arise when swarms of ant agents path find using the best performer XSets and the indices of merit that are observed when multiple targets are considered using the same XSets. The critical F - value in the stigmergic category that is read from statistical tables is 0.1659. This is the largest F - value we require in order to accept the null hypothesis. On the other hand, the critical F - value in the message passing category is observed

	Cross	4way cross	Triangle	Rectangle	Pentagon	Hexagon	Heptagon	Octagon	Circle
Stigmergic	0.292	0.194	0.668	0.207	0.525	0.558	0.583	0.884	0.884
Message passing	1.321	1.397	1.325	1.292	1.416	1.467	1.363	1.589	1.379

Table 6.4: F - tests : single vs multiple targets

	Cross	4way cross	Triangle	Rectangle	Pentagon	Hexagon	Heptagon	Octagon
Four-way cross	**0.64**							
Triangle	2.28	3.56						
Rectangle	**0.68**	1.05	**0.29**					
Pentagon	1.79	2.80	**0.78**	2.66				
Hexagon	1.90	2.97	**0.83**	2.82	1.06			
Heptagon	1.99	3.11	**0.87**	2.95	1.11	1.04		
Octagon	3.02	4.70	1.32	4.46	1.67	1.58	1.51	
Circles	3.02	4.71	1.32	4.47	1.68	1.57	1.52	1.00

Table 6.5: F-tests : multiple to multiple targets (stigmergic)

	Cross	4way cross	Triangle	Rectangle	Pentagon	Hexagon	Heptagon	Octagon
Four-way cross	1.06							
Triangle	1.06	1.00						
Rectangle	1.04	**0.97**	**0.97**					
Pentagon	1.15	1.08	1.08	1.11				
Hexagon	1.15	1.08	1.08	1.11	1.00			
Heptagon	1.08	1.01	1.01	1.04	**0.94**	**0.94**		
Octagon	1.21	1.14	1.13	1.17	1.05	1.05	1.12	
Circles	1.02	**0.96**	**0.96**	**0.98**	**0.89**	**0.88**	**0.95**	**0.84**

Table 6.6: F-tests : multiple to multiple targets (message passing)

to be 6.0289, indicating the largest F - value we require in order to accept the null hypothesis in this category.

Generally, stigmergic F - values are bigger than the critical F - value. This implies that the variances we observe in the sets of indices of merit thereof are significantly different from those we see when multiple targets are considered. This outcome is consistent with the outcome reported in the correlation analyses on the same data sets. Thus, we also lack evidence to accept the alternative hypothesis and rather continue to believe that there are no relationships between the indices of merit that are reported when ant agents path find, and when multiple targets are considered using stigmergic XSets. We noted earlier on that this outcome is related to the sensitivity of stigmergic XSets to the number of multiple targets, as well as the radius of the region on which multiple targets are set.

On the contrary, the F - test results we observe in the message passing category are all smaller than the corresponding critical F - value. This implies that the variances we observe in the sets of indices of merit thereof are similar to one another when path finding or locating multiple targets. These results are also consistent with the outcome reported in correlation analyses. We justified this outcome as arising from the insensitivity of message passing XSets to the number of multiple targets, and the radius of the region on which multiple targets are set. Similarly, we lack sufficient evidence to reject the hypothesis that message passing XSets are, in fact, problem independent.

Table 6.5 and Table 6.6 show the F-test results that are yield when we compare pairs of sets of indices of merit that are taken from the same category when multiple targets are considered. We note that triangular setup of multiple targets yield significantly different indices of merit to the rest in the stigmergic category. This is explained by the depletive nature of complex environments to the stigmergic model.

Message passing XSets achieve equally bad variations when circular multiple targets are considered. We explain this outcome to arise from the geometry of circular regions of targets which differs from the geometries of the rest of the targets configurations we consider. The results we see in the F - tests connote problem independence in these XSets. However, the same results re-iterate sensitivity to the configuration of the multiple targets regions.

Mean compares the central tendency between two data sets. In this case, it establishes the degrees of similarities between two sets of indices of merit. Our null hypothesis is that the mean indices of merit of path finding XSets are similar to the mean indices of merit that are yield when multiple targets are considered. We indicated in Chapter 5 that T - tests are suitable for comparisons between means. Two tailed T - tests are recommended because the possible alternative hypotheses are not directional (Lohr, 2010).

The P - values we achieve in T - tests indicate the strength of evidence in support of the null hypothesis. If the P - value we get is greater than the significance level (which in this case is $\alpha = 0.01$), then we do not have sufficient evidence to reject the null hypothesis. Comparisons between means requires us to assume similar or different variances upfront. Our stigmergic F - test results reported significantly different variations, thus allowing us to conduct all the T - tests in this category assuming different variances. Message passing F - tests reported similar variations in the indices of merit thereof, thus allowing us to conduct T-tests assuming similar variances.

Table 6.7 reports the T - tests results we achieve when we compare the indices of merit that arise when swarms of ant agents path find and when multiple targets are considered. Thereafter, Table 6.8 and Table 6.9 respectively report the T - tests results that are achieved between the indices of merit of XSets when only multiple targets are looked at.

	Cross	4way cross	Triangle	Rectangle	Pentagon	Hexagon	Heptagon	Octagon	Circle
Stigmergic	0.299	0.255	0.768	0.345	0.594	0.575	0.484	0.596	0.824
Message passing	0.888	0.827	0.804	0.873	0.826	0.824	0.880	0.810	0.924

Table 6.7: T - tests: single vs multiple targets

	Cross	4way cross	Triangle	Rectangle	Pentagon	Hexagon	Heptagon	Octagon
Four-way cross	0.817							
Triangle	0.371	0.312						
Rectangle	0.984	0.845	0.426					
Pentagon	0.528	0.430	0.767	0.567				
Hexagon	0.537	0.437	0.745	0.576	0.982			
Heptagon	0.620	0.499	0.624	0..651	0.861	0.876		
Octagon	0.469	0.385	0.785	0.519	0.953	0.931	0.796	
Circles	0.317	0.273	0.913	0.378	0.682	0.658	0.536	0.685

Table 6.8: Stigmergic T-tests

	Cross	4way cross	Triangle	Rectangle	Pentagon	Hexagon	Heptagon	Octagon
Four-way cross	0.926							
Triangle	0.897	0.971						
Rectangle	0.982	0.943	0.915					
Pentagon	0.927	0.998	0.969	0.944				
Hexagon	0.923	0.999	0.971	0.941	0.997			
Heptagon	0.991	0.933	0.904	0.991	0.933	0.931		
Octagon	0.907	0.983	0.986	0.925	0.981	0.984	0.914	
Circles	0.955	0.880	0.852	0.937	0.879	0.877	0.946	0.861

Table 6.9: Message passing T-tests

All the P - values we observe from using the two categories of XSets are bigger than the significance level of $\alpha = 0.01$. However although stigmergic indices of merit are generally above the significance level, they are still relatively close to the rejection zone. Indices of merit that are reported in the message passing category are significantly larger than the significance level. These results indicate lack of evidence to reject the null hypothesis in both cases, implying that the mean performances we observe in the indices of merit in both categories are similar.

6.4.1.3 Observations

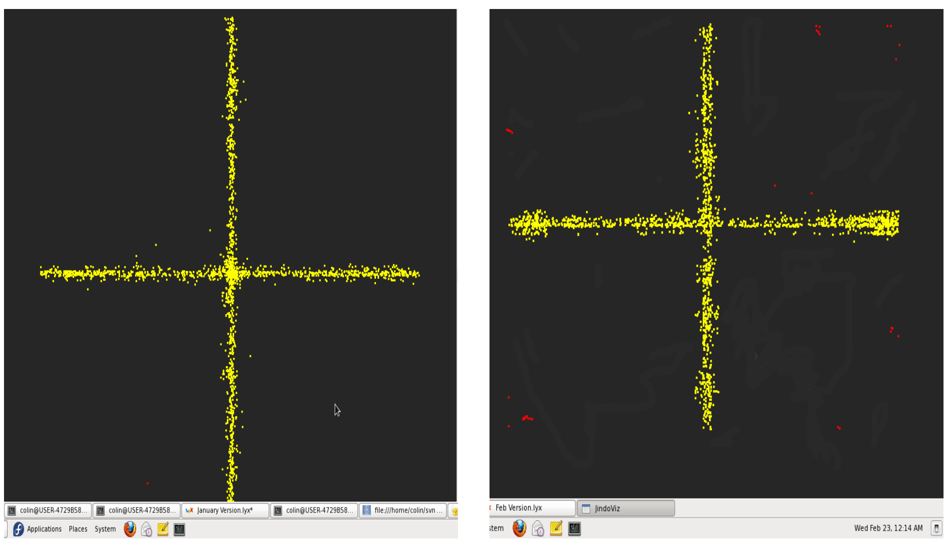
We point out the most outstanding observations that arise from the results hereto. At the top of the list is the observation that the message passing XSets are more problem independent than the stigmergic counterparts. Precisely, stigmergic XSets show signs of sensitivity to the number of multiple targets while message passing XSets are generally insensitive to this aspect. Similarly, stigmergic XSets are sensitive to the spread or radius of the region covered by multiple targets while the message passing counterparts are not. We expand this observation as follows:

- Although relationships are weak between the sets of indices of merit that are observed when stigmergic XSets are used, visual evidences of successful targets search show fairness in the distribution of hits on multiple targets setups (see the visual outcomes that are shown in Figures 6.5 to 6.13 as proof of arrivals of ant agents on multiple targets). In these screenshots, ant agents that are isolated are marked in red. The spots at which successful ant agents hit the targets are marked by the yellow spots.

- Relatively more message passing ant agents are marked as isolated than the stigmergic counterparts (see Figures 6.5 to 6.13). This observation is consistent with what we saw in Chapter 4 in Figure 4.13. We explained that this outcome is related to the way in which swarm information is held in the system. The stigmergic model uses cell tuples as containers of both attractive and repulsive levels of pheromone. Repulsive levels in particular, propel ant agents away from their current positions, thereby enhancing navigational abilities. This way, ant agents are rarely trapped on sub-optimal solutions. This property on its own, increases stigmergic ant agents' chances of converging. On the other hand, message passing ant agents may degrade in confidence when they are isolated, until the are trapped in false direction vectors which they would continue to follow unconditionally. As such the stigmergic model is more fault tolerant and robust than the message passing model.

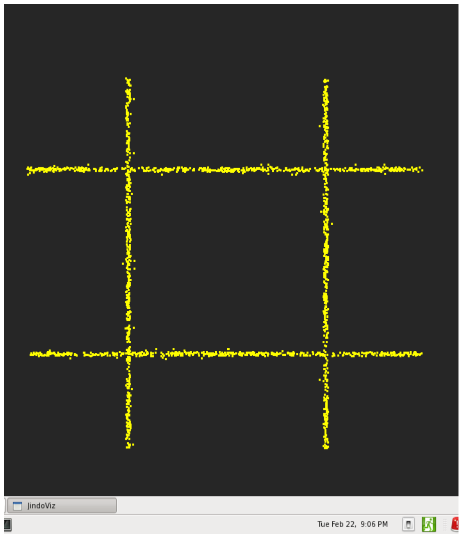
- Stigmergic XSets generally gain in fitness levels with time in simulation until some turning point is reached (see Figure 6.4 regarding these fitness trends). Key in this observation is that the scoring time we allow swarms in simulation has a bearing on the indices of merit and the general quality of emergence thereof. Precisely, less time in simulation is detrimental because insufficient levels of pheromone are built on the environment in order to guide ant agents towards the targets sought. On the contrary, more time in simulation depletes the paths that arise when excess levels of pheromone saturate the environment. Thus, users may be required to set appropriate simulation time limits upfront. Such time limits are however a function of agent density, environment size, target size, dissipation factors, as well as the distances between the targets and the starting point. More experiments which investigate the effects of each of these factors to the indices of merit of XSets in this category are motivated from this observation.

- Message passing XSets equally gain in fitness levels with time in simulation (see Figure 6.4). Time in simulation remains the key independent variable which determines how high the indices of merit go. Low indices of merit are observed early in simulation time because ant agents have not built confidence in the geometric vectors they follow. However message passing ant agents can even create deterministic paths if they are allowed enough time in simulation. Similarly the time message passing ant agents spend in simulation has a bearing on the indices of merit of the XSet thereof. The same time is also a function of agent density, environment size, target size, as well as the distances between the targets and starting points.

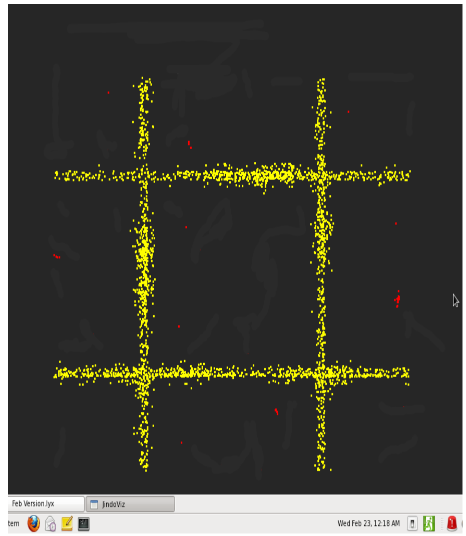


(a) Stigmergic cross (b) Message passing cross

Figure 6.5: Creating cross structures

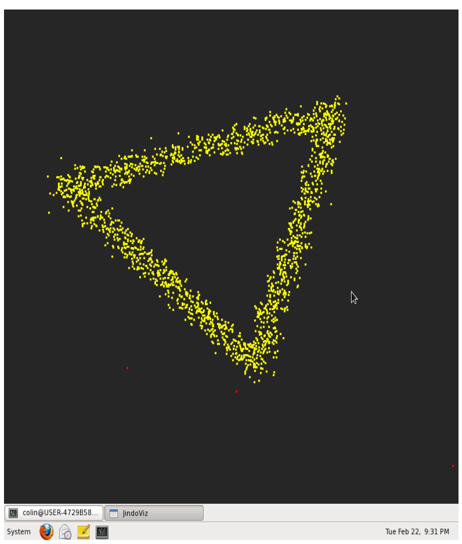


(a) Stigmergic four-way cross

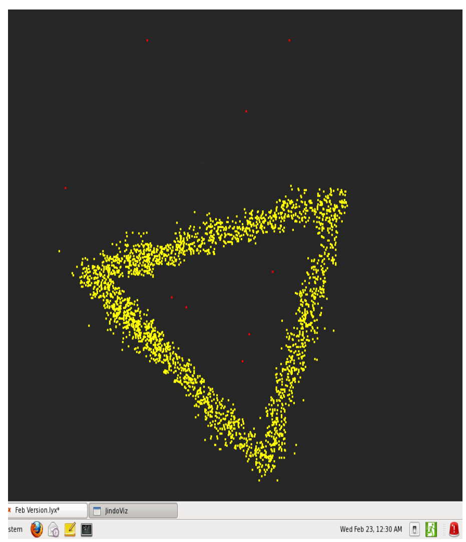


(b) Message passing four-way cross

Figure 6.6: Creating four-way crosses

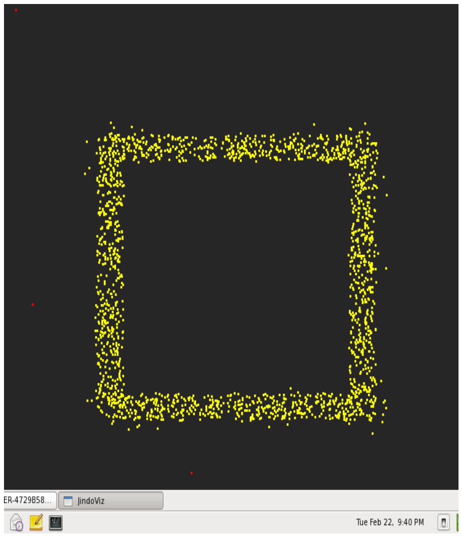


(a) Stigmergic triangle

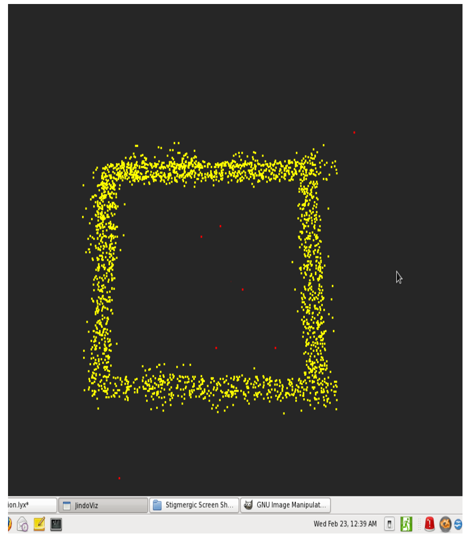


(b) Message passing triangle

Figure 6.7: Creating triangles

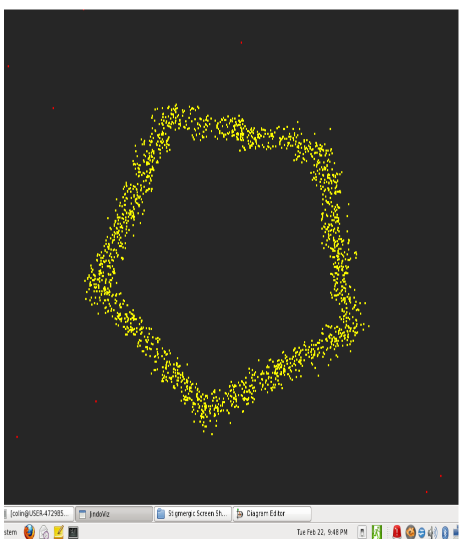


(a) Stigmergic rectangle

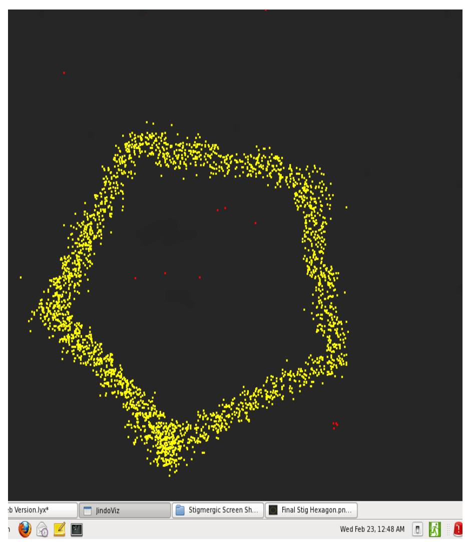


(b) Message passing rectangle

Figure 6.8: Creating rectangles

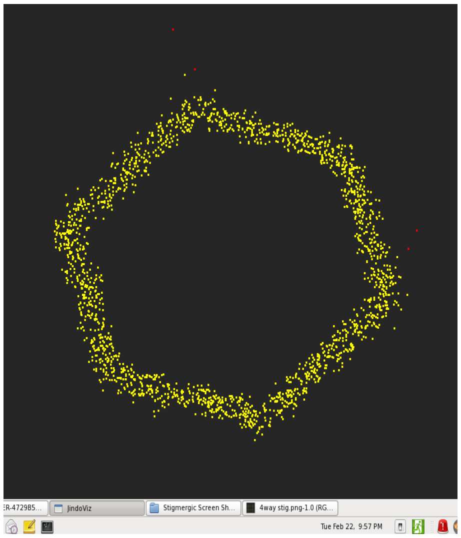


(a) Stigmergic pentagon

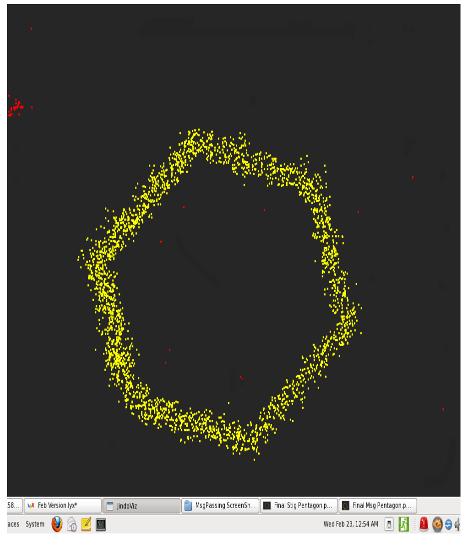


(b) Message passing pentagon

Figure 6.9: Creating pentagons

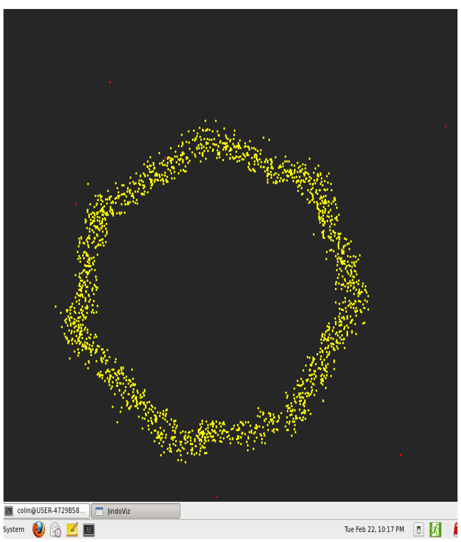


(a) Stigmergic hexagon

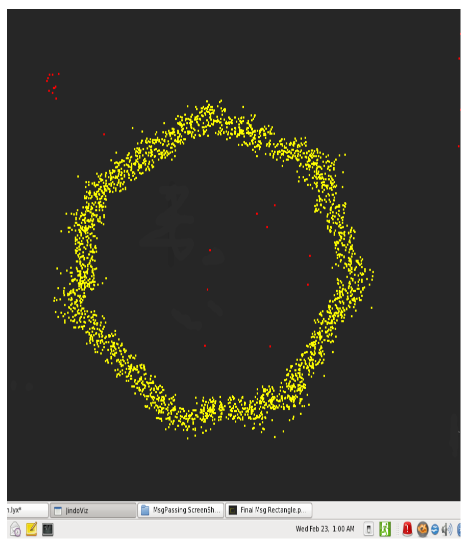


(b) Message passing hexagon

Figure 6.10: Creating hexagons

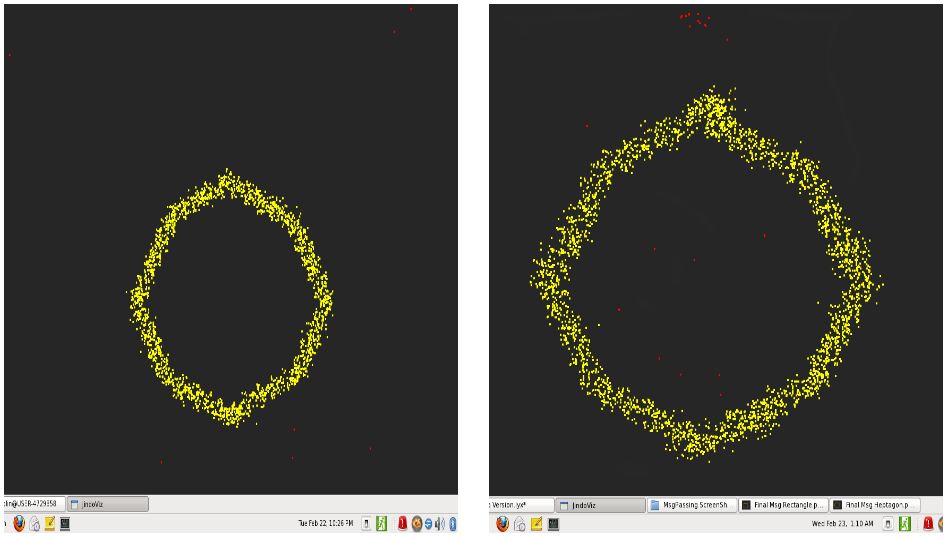


(a) Stigmergic heptagon



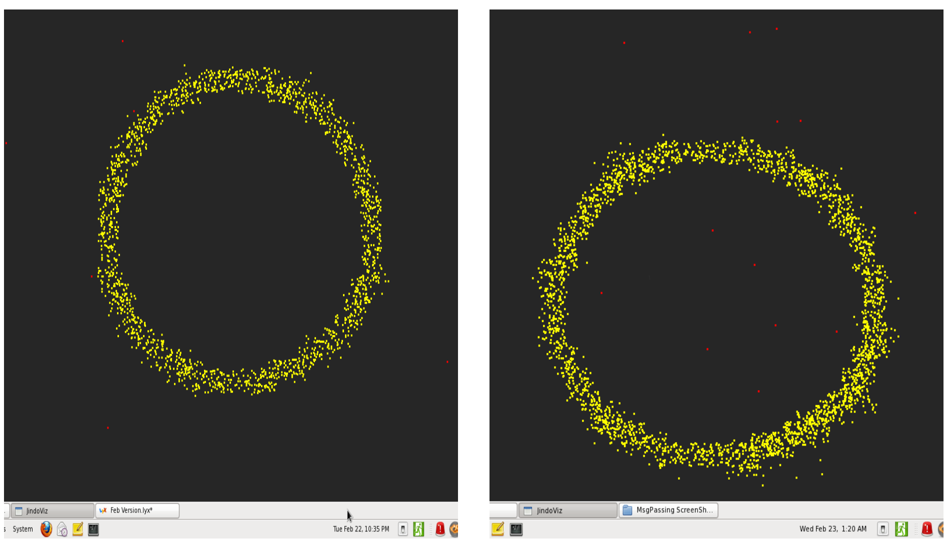
(b) Message passing heptagon

Figure 6.11: Creating heptagons



(a) Stigmergic octagon (b) Message passing octagon

Figure 6.12: Creating octagons



(a) Stigmergic circle (b) Message passing circle

Figure 6.13: Creating circular structures

6.4.2 Experiment 2: Effects of agent density

This experiment is motivated by two of the observations we made in the first experiment. These observations are re-phrased as follows:

1. Threshold scoring times are required upfront in order to avoid stigmergic swarms from depleting late in simulation. These threshold time limits are functions of *agent density*, environment size, target size, dissipation factors, as well as the distances between the targets and the starting points. What is the effect of agent density to the indices of merit of XSets?
2. The time message passing ant agents require to converge on particular vector fields is a function of *agent density*, environment size, target size, as well as the distances between the targets and starting points. Similarly, to what extent does agent density influence the indices of merit we observe in message passing XSets.

The question that arises from these two observations relates to connotation that different indices of merit, as well as different statistical conclusions, may arise as a result of variations in agent density. Our hypothesis is that ant agent density has no effect on the trends in the indices of merit we observe in each case. However this is a bold claim which requires us to practically evaluate the effects of both lower and higher agent densities than the default agent density that has been used so far.

6.4.2.1 Experiment Design

Previous experiments used a controlled agent density of 5000 ant agents. The indices of merit we reported so far, using this agent density, serve as

Title Investigating the effects of agent density to the indices of merits of XSets.

Hypothesis agent density has no effect on the indices of merit of XSets

Dependent variable : indices of merit that arise when agent density is changed between 500 and 10000.

Independent variables : agent densities (500, 1000, 1500, 2000, ...until 10000), Time in simulation (max. of 10000 iterations), number of replications (max. of ten), targets configuration.

Controlled variables: These are the variables that are kept constant: environment size ((100×100 grid), position of the starting point (fixed), Centre of targets - centre of the environment, composition of XSets (as used in the first experiment), sequences of primitive behaviours in XSet (as in first experiment), dissipation controls (ignored), radius of multiple targets (equal in all polygons), number of pheromones supported (maximum of 5), number of vectors supported (four).

Procedure - Generator functions are invoked which define the evaluation environments we want. Ant agents are deployed in particular agent densities, using path finding XSets. Indices of merit are assessed, reported and compared to the performances of 5000 ant agents in each category.

```
foreach selected environment
  for-density in [500,1000, 1500,.....10000]
     repeat
```

Control level	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
speed										
quality										
delivery										
delays										
Shannon										
cumm. IOM	$x_1+=IOM_{1,i}$	$x_2+=IOM_{2,i}$	$x_3+=IOM_{3,i}$	$x_4+=IOM_{4,i}$	$x_5+=IOM_{5,i}$	$x_6+=IOM_{6,i}$	$x_7+=IOM_{7,i}$	$x_8+=IOM_{8,i}$	$x_9+=IOM_{9,i}$	$x_{10}+=IOM_{10,i}$

```
     until 10 replications
     Find gap with 5000 ant agents at each control level
    next density

next environment
```

Figure 6.14: Template of the design of experiment 2

benchmark upon which we assess the effects of reducing or increasing agent densities. The key dependent variable we measure in this experiment are similarly the indices of merit that arise between the benchmark indices of merit and the indices of merit we achieve when agent density is varied.

Three independent variables are key in this experiment namely: time in simulation, configuration of targets, and agent density. In this case, we select arbitrary agent densities both below and above 5000. Precisely, we consider swarms of ant agents in colonies of $500 \times i$, where i is an integer in the range $[1, 20]$. From the observations we made on repeated tests, 500 to 10000 ant agents in a swarm are a sufficiently wide range of agent densities to reveal the effects we seek.

Ten replications of the same experiment are administered when a particular agent density is used over the simulation limit of 10000 iterations. As a case study, these ant agents are tasked to locate multiple targets, as well as path find in different agent densities. We similarly extract the indices of merits at every 1000^{th} iteration.

Figure 6.14 describes the rest of the variables of this experiment in details, as well as the procedure we follow in completing this experiment.

6.4.2.2 Results

Figure 6.15 plots the indices of merit when the benchmark agent density (5000 ant agents) is used. The same Figure reports the performances of these swarms over 10000 iterations. In Figure 6.15(a), we report the indices of merit of the stigmergic XSets, while Figure 6.15(b) shows the performance trends followed by message passing XSets over the same simulation limit.

In Figures 6.16(a) and 6.16(b), we respectively report the indices of merit of XSets in the stigmergic and message passing category when 500 ant agents are deployed for similar purposes (path finding).

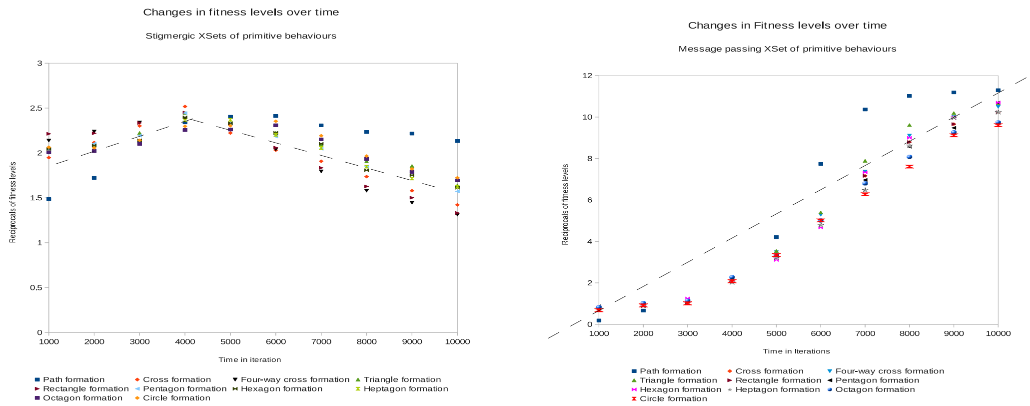
Figures 6.17, 6.18, and 6.19 then plot the fitness trends that arise when 3000, 7000, and 10000 ant agents are deployed in each category for the same purposes as well. We have chosen to show the performances of 500, 3000, 5000, 7000, and 10000 ant agents merely as proof of concept. The indices of merit that arise when we use other agent densities show similar growth or drop with changes in agent density.

6.4.2.3 Observations

We make the following observations regarding the results reported in Figure 6.15 to Figure 6.19:

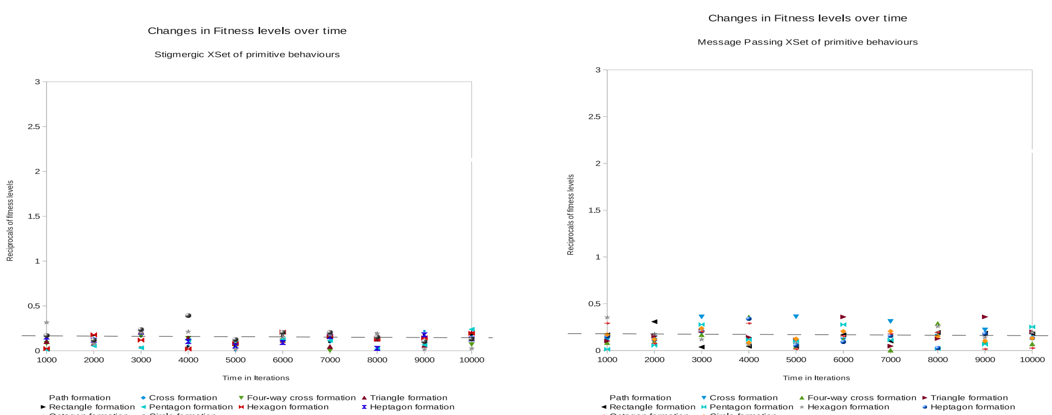
- Stigmergic indices of merit that arise when low agent densities are used are poor. This is because few ant agents cannot build useful shared memories with convergence properties. As a result, large gaps are observed between these indices of merit and the benchmark indices of merit. Raising agent density improves the indices of merit because shared memories arise faster.

- Stigmergic indices of merit that are achieved late in simulation time produce equally large fitness gaps in favour of the benchmark swarms. We note that these gaps are even bigger as we raise agent density. This is because,on one hand, low agent densities fail to influence the creation of useful guides to the swarms. On the other hand, large agent densities result in saturated environments which deplete the paths that emerged earlier. These views confirm that agent density bear influences to the indices of merit of stigmergic XSets. However, determining appropriate agent densities for every given scenario is a mathematical problem which requires one to consider the size of the environment and the distance between the targets and the starting point.



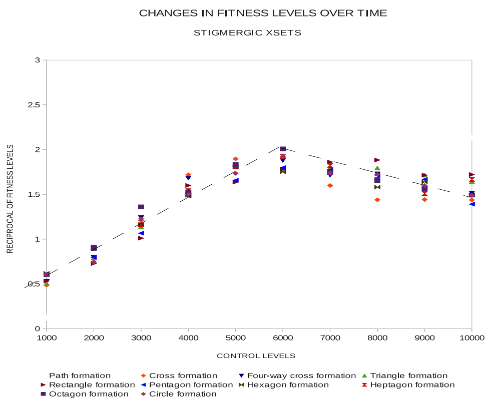
(a) 5000 stigmergic ant agents (b) 5000 message passing ant agents

Figure 6.15: Performances of benchmark swarms

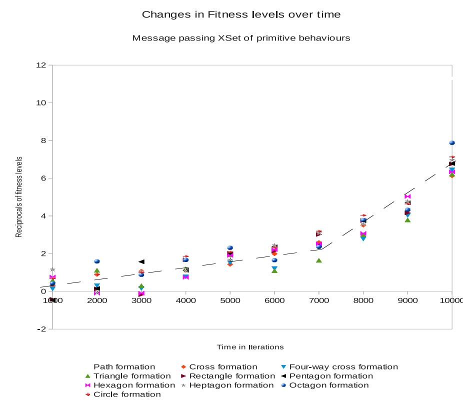


(a) 500 stigmergic ant agents (b) 500 message passing ant agents

Figure 6.16: Performances of 500 ant agents in swarms

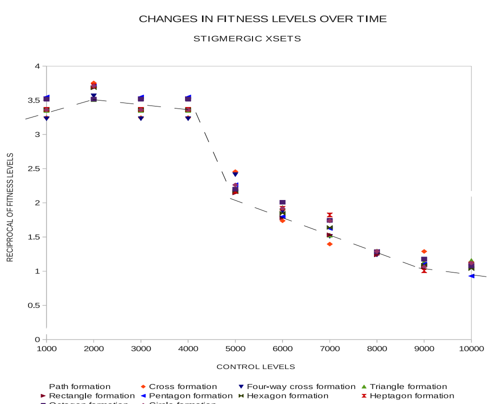


(a) 3000 stigmergic ant agents

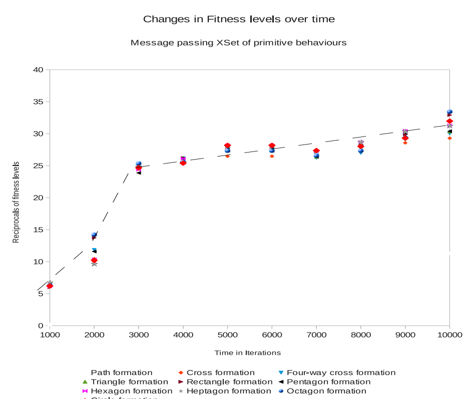


(b) 3000 message passing ant agents

Figure 6.17: Performances of 3000 ant agents in swarms

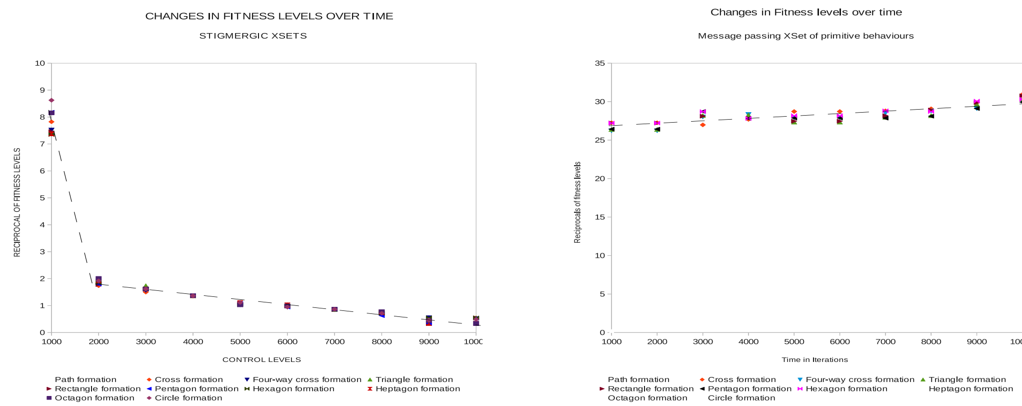


(a) 7000 stigmergic ant agents



(b) 7000 message passing ant agents

Figure 6.18: Performances of 7000 ant agents in swarms



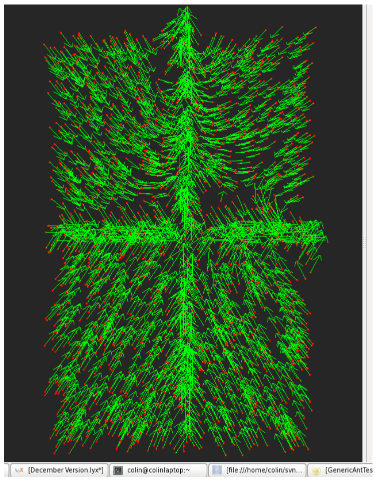
(a) 10000 stigmergic ant agents

(b) 10000 message passing ant agents

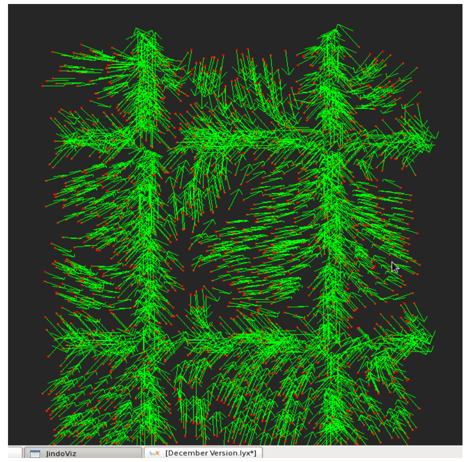
Figure 6.19: Performances of 10000 ant agents in swarms

- The changes we observe in stigmergic indices of merit over time indicate that a turning point always arises regardless of the agent density used. The gradient of the trend curves we overlap on the fitness trends are steeper and steeper after the turning point (see Figures 6.16(a), 6.17(a), 6.4(a) ,6.18(a), and Figure 6.19(a)). This implies that high agent density causes stigmergic swarms to converge faster, while on the other hand the indices of merit thereof would deplete sooner in simulation. This again confirms the influences of agent density to the fitness levels of stigmergic XSets.

- Message passing ant agents require time to build confidence in the vectors they follow. Low agent density delays this process because ant agents are far apart to usefully share directional cues. We observe large fitness gaps in early control levels when low agent densities are used because ant agents are still gathering directional information with



Cross formation

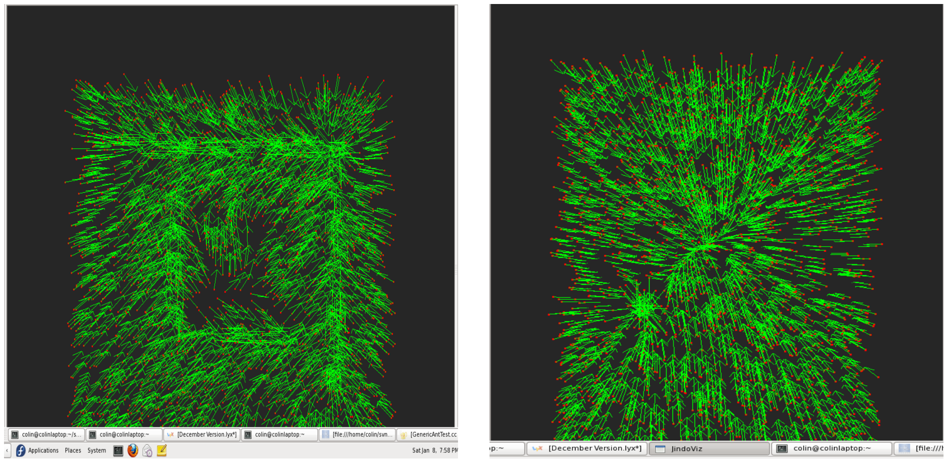


Fourway cross formation

Figure 6.20: Propagation of vector fields in swarms of 5000 ant agents

which to build reliable vector fields. High agent density reduces the time it takes swarms to converge on specific vector fields (see Figures 6.16(b), 6.17(b), 6.4(b) ,6.18(b), and Figure 6.19(b)). Thus, the power of message passing ant agents is concluded to be in numbers. With sufficiently high agent densities and more time in simulation, message passing ant agents can even follow deterministic paths. Figures 6.20 and 6.21 show the emergence of deterministic vector fields when high agent density swarms are used. In these screenshots, message passing ant agents are allowed to orientate and point in the direction they would move.

The results here reported refute the null hypothesis (agent density has no influence on the fitness levels of XSets) in favour of the alternative hypothesis. In stigmergic swarms, low agent density prevents the creation of useful shared memories. On the other hand, high agent density saturates the environments early in simulation. In the message passing category, low agent density prevents one-on-one communication between ant agents. However high agent density enhances the emergency of deterministic paths.



Rectangle formation Plotting a point

Figure 6.21: Propagation of vector fields (5000 ant agents)

In both cases, we see the need to prescribe threshold agent densities. This quantity is a function of the size of environment, scoring time, as well as the distance between the starting point and the targets. We conclude that agent density is a key parameter of emergence to consider when using similar ant XSets.

6.4.3 Experiment 3: Effects of primitive behaviours

The influences of discrete primitive behaviours to the indices of merit of XSets are the key subject of study in this experiment. Similar identification of useful behaviours in gene-pairs have been proposed in the work of Poon and Maher (1996) - where an algorithm for calculating the contributions of a pair of behaviour genes to the fitness of phenotypes is proposed. We try to respond to the question which says: what are the influences of discrete primitive behaviours that are included in the composition of XSets to the indices of merit thereof?

Precisely, this experiment investigates the ingredients of emergent behaviour in each best path finding XSet. Our null hypothesis in this investigation is that discrete primitive behaviours do not possess direct or individual influences to the indices of merit of XSets. We measure this hypothesis by evaluating the gaps between the indices of merits of benchmark XSets and those of variant XSets. Benchmark XSets are the original best performer XSets that are reported and presented in Figures 4.10 and 4.11. On the other hand, variant XSets are those XSets in which a particular primitive behaviour is intentionally omitted from the benchmark XSet. The procedure through which we assess the effects of each discrete action are summarized in the next sub-section.

6.4.3.1 Experiment Design

The key dependent variable of this experiment are the indices of merit of variant XSets. These are assessed at each control level in order to determine the overall average index of merit of each variant XSet. This overall average index of merit of the variant XSet is compared with the index of merit of the benchmark XSet of the same category.

Three outcomes are possible. A positive gap between the benchmark XSet's index of merit and the variant XSet's index of merit indicates that the absence of the particular primitive behaviour that was omitted in the benchmark XSet has degraded the performances thereof. This implies that the omitted primitive behaviour has positive effects to the overall performances of the benchmark XSet.

On the other hand, a negative gap between the benchmark XSet's index of merit and the variant XSet's index of merit indicates that the absence of the particular primitive behaviour that was omitted in the benchmark XSet, in fact, enhanced the performances thereof. This implies that the omitted

Title Investigating the effects of discrete primitive behaviours to the indices of merit of XSets.

Hypothesis discrete primitive behaviours do not possess direct influences to the indices of merit of XSets

Dependent variable : indices of merit of variant XSets.

Independent variables : configuration of XSets, time in simulation, replications, target setups, control levels.

Controlled variables: environment size ((100×100 grid), position of the starting point (fixed), Centre of targets - centre of the environment, agent density (5000), composition of XSets (as used in the first experiment), sequences of primitive behaviours in XSet (as in first experiment), dissipation controls (ignored), radius of multiple targets (equal in all polygons), number of pheromones supported (maximum of 5), number of vectors supported (four).

Procedure - A particular benchmark XSet is chosen (XSet []). Its primitive behaviours are accessed and omitted one by one. Generator functions are invoked which define the evaluation environments we want. A variant XSet is used to coordinate a swarm of 5000 ant agents over ten replicated simulations and the average indices of merit are computed. Gaps between these indices of merit and the benchmark indices of merit are determined where a positive gap indicates positive effects of the omitted primitive behaviour to the overall index of merit of the benchmark XSet. A negative gap indicates that the primitive behaviour even degrades the index of merit of the XSet.

```
foreach XSet []
   count = 0 :  bIOM←IOM of benchmark XSet
   varXSet = XSet [] - XSet [count]
   foreach environment
       repeat
```

Control level	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
speed										
quality										
delivery										
delays										
Shannon										
cumm. IOM	$x_1 += IOM_{1,i}$	$x_2 += IOM_{2,i}$	$x_3 += IOM_{3,i}$	$x_4 += IOM_{4,i}$	$x_5 += IOM_{5,i}$	$x_6 += IOM_{6,i}$	$x_7 += IOM_{7,i}$	$x_8 += IOM_{8,i}$	$x_9 += IOM_{9,i}$	$x_{10} += IOM_{10,i}$

```
     until 10 replications
```

$index\ of\ merit = \frac{\sum_{k=1}^{10} \frac{x_k}{10}}{10}$

```
   gap = bIOM - index of merit)
    if(gap > 0)
      XSet [count] enhances IOM
    else if (gap<0)
      XSet [count] degrades IOM
    else
      XSet [count] has no effect
    next environment
   count ++
next XSet
```

Figure 6.22: Template of the design of experiment 3

primitive behaviour has negative effects to the overall performances of the benchmark XSet. There are cases when a primitive behaviour may be found not to have any effects to the performances of the benchmark XSet.

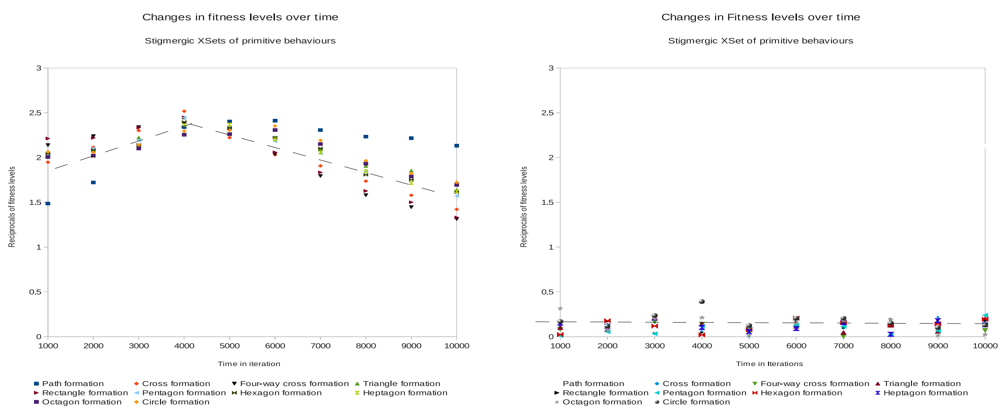
The main independent variables of this experiment are the configuration of XSets that are used in each cycle. Time in simulation, number of replications, multiple targets setups, as well as the control levels at which we extract results remain other key independent variables that we monitor.

The procedure that is presented in Figure 6.22 indicates that benchmark XSets are represented as arrays of primitive behaviour. As such, variant XSets arise when we omit a primitive behaviour at a particular index in this array. The variant XSets that arise are evaluated for abilities to deliberately cause path finding as well as multiple target location.

6.4.3.2 Results

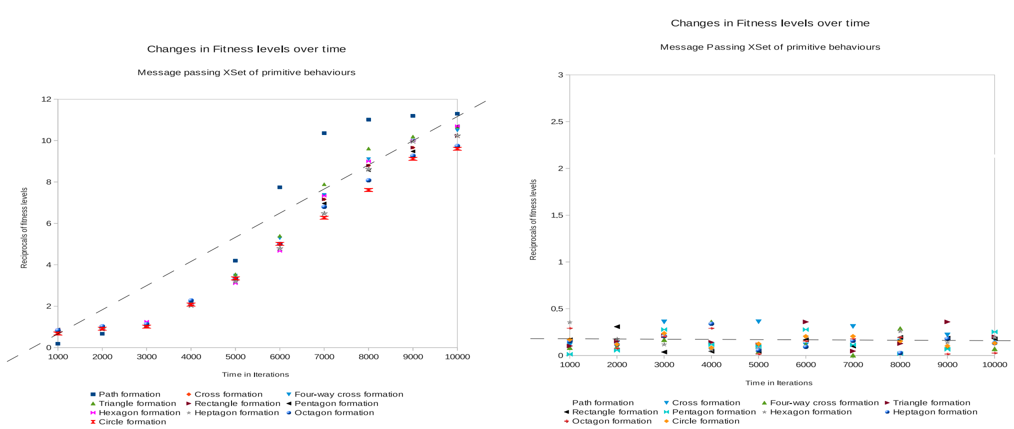
Figure 6.23 shows the performance gaps that arise between the performances of benchmark XSets and variant XSets in the stigmergic category over time. In Figure 6.23(a), we show the trends in indices of merit that arise when the benchmark XSet is used for path finding and multiple targets location. Figure 6.23(b) then shows the trends that arise when the functionality to drop pheromone, orientate, or changing from one internal state to another are omitted. In all cases, the discrete ability of ant agents to move is indispensable.

Generally, dropping any of the other three primitive behaviours degrade the performances of the benchmark XSet. Precisely, when ant agents fail to drop specific levels of pheromone, shared memories are not built. This has the negative effect of keeping ant agents in the random wandering mode. Similarly, ant agents that lack abilities to orientate would continue to wander at random even though the levels of pheromone are building on the environment.



(a) benchmark stigmergic XSets (b) Variant stigmergic XSets

Figure 6.23: Fitness levels of stigmergic XSets



(a) benchmark message passing XSets (b) Variant message passing XSets

Figure 6.24: Fitness levels of message passing XSets

In fact, these levels of pheromone are useless to the ant agents. The ability to flip between internal states is also critical. This is because although ant agents may build trails with clues towards the starting point, they will never get to use these trails since they will remain in the search internal state. Similarly, trails towards the targets would never arise since none of the ant agents in the swarm would never flip to the return mode when they can drop the levels of pheromone with cues toward the targets. Thus, omitting any of these three primitive behaviours has the same effect of reverting the swarms to the random wandering mode.

Figure 6.24 shows the performances gaps that arise between the performances of benchmark XSets and variant XSets in the message passing category. In Figure 6.24(a), we show the fitness trends that arise when the benchmark XSet in this category is used for path finding and multiple targets location as well. Figure 6.24(b) depicts the trends that arise when the ability to share message, orientate, detect targets, or flipping between internal states are omitted from the benchmark XSet. Similarly, ant agent abilities to move is indispensable.

Likewise, dropping any of the other four primitive behaviours deplete the message passing model. Precisely, ant agents that do not share messages are operationally isolated. They keep wandering aimlessly, at the same time degrading in confidence in every step. Failure to orientate equally define random wandering swarms. Message passing ant agents require some mechanism for detecting target indicators and convert these to vector information, vectors that would point towards the targets. Omission of this ability implies that ant agents would continue in the search internal state, the same way they would behave if the XSet they use excludes the ability to flip between internal states.

An outstanding outcome is noted in the message passing category. Ant agents' abilities to normalize vectors do not deplete the model completely. In fact, normalizing vectors merely enhances swarm performances but does not cause emergent behaviour. Visual observation of the swarms in which this primitive behaviour is omitted shows ant agents whose movement steps and speed vary. This observation suggests that the magnitude of the resultant vector has a bearing on the speed of ant agents as well as the size of ant agent steps.

6.4.3.3 Observations

We make the following observations regarding the results we report in this experiment:

- Generally, ant agents whose variant XSets omit abilities to update swarm information (dropping levels of pheromone in the case of stigmergic ant agents, or explicitly sharing information in the case of message passing ant agents), orientate, or change internal states, define random wandering swarms whose performances are completely out-classed by the performances of benchmark XSets. This is because there would not be any cooperation in the swarms and un-informed movements would arise when ant agents do not orientate.

- A trivial observation is that ant agent movements are critical for the problem domains we propose. Besides being an inherent characteristic of foraging ant agents in simulation and ants in nature, movement is a key ingredient for any form emergent formation. As a result, this work needed not to assess the effects of omitting this primitive behaviour.

- We note that there are primitive behaviours in XSets that work well in collaboration. Thus, omitting one primitive behaviour would fail the XSet even when the functionality of the other primitive behaviour is not avoided. For example, although orientation is critical in the stigmergic category, the ability of ant agents to drop specific levels of pheromones onto the environment gives a meaning to agent orientation. Rearranging the primitive behaviours that are left in the variant XSets may hopefully replace the missing functionality, and this is the motivation for the next experiment.

Overall, the findings of this experiment provide sufficient evidence with which to reject the null hypothesis which states that discrete primitive behaviours do not possess direct or individual influences to the indices of merit of XSets.

6.4.4 Experiment 4: Effects of order in XSets

The motivation for conducting this experiment arises from the remark we made in the previous experiment regarding possibilities of rearranging primitive behaviours in variant XSets in order to replace missing functionality. Does the order in which primitive behaviours are arranged in XSets influence the indices of merit thereof? This is the question that drives the investigations we conduct in this experiment.

Our hypothesis is that order has no effect to the indices of merit of XSets. We explained the motivation for stating hypotheses in this way as inspired by known statistical propositions that one can only falsify statements but cannot prove them (Lohr, 2010). This hypothesis can be measured by comparing the indices of merit of benchmark XSets to the indices of merit of variant XSets. In this case, a variant XSet is a partial permutation of the benchmark XSet. Partial permutations are combination of primitive behaviours with no repetitions (see section 4.2.2 for details regarding partial permutations). We are saying that a variant XSet in this case is the same benchmark XSet (in terms of composition) with primitive behaviours that are re-arranged.

6.4.4.1 Experiment Design

Figure 6.25 summarizes the procedure we follow in order to complete this experiment. In this, the key dependent variable we measure are similarly the indices of merit of variant XSets which we compare with the indices of merit of benchmark XSets in each category. Variant XSets are assessed for abilities to coordinate swarms of ant agents towards path finding as well as towards multiple targets location.

The main independent variable (the variable we manipulate) is the sequence of primitive behaviours in XSets, along with the time ant agents take in simulation. Targets setups are also a key independent variable. The rest of the variables of this experiment are controlled.

To run the experiment, each benchmark XSet is first used to generate variant XSets. We indicated that the stigmergic benchmark XSet has four primitive behaviours in each internal state, while the message passing counterpart has five. Mathematically we will have twenty four possible partial permutations in each internal state of the stigmergic XSets. One hundred and twenty variations are possible in each internal state of the message passing category. However we ignore the primitive behaviour for vector normalization in the message passing category since we found out that this functionality is not critical (see section 6.4.3.2 for this observation), leaving us with twenty four permutations in each internal state as well.

This experiment takes each variant XSet as an input with which to coordinate swarms of ant agents towards desired emergent behaviour. The performances of each variant XSet are compared with those of the benchmark XSet in the same category. That gap we see between the performances of the variant and benchmark XSets tells the effects of each sequence of primitive behaviours.

Title Investigating the effects of the order in which primitive behaviours are arranged in XSets.

Hypothesis the order in which primitive behaviours are arranged in XSets has no effects to the indices of merit of the XSets.

Dependent variable : indices of merit of variant XSets

Independent variables : order of primitive behaviours in XSets, time in simulation, targets setups, control levels, replications.

Controlled variables: environment size ((100×100 grid), position of the starting point (fixed), Centre of targets - centre of the environment, agent density (5000), composition of XSets (as used in the first experiment), sequences of primitive behaviours in XSet (as in first experiment), dissipation controls (ignored), radius of multiple targets (equal in all polygons), number of pheromones supported (maximum of 5), number of vectors supported (four).

Procedure - The algorithm below summarizes the procedure we follow to complete this experiment.

```
foreach XSet []
  bIOM←IOM of benchmark XSet
  varXSet [] = all partial permutations of XSet []
  foreach varXSet []
     foreach environment
       repeat
```

Control level	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
speed										
quality										
delivery										
delays										
Shannon										
cumm. IOM	$x_1 += IOM_{1,i}$	$x_2 += IOM_{2,i}$	$x_3 += IOM_{3,i}$	$x_4 += IOM_{4,i}$	$x_5 += IOM_{5,i}$	$x_6 += IOM_{6,i}$	$x_7 += IOM_{7,i}$	$x_8 += IOM_{8,i}$	$x_9 += IOM_{9,i}$	$x_{10} += IOM_{10,i}$

```
      until 10 replications
```

$$index\ of\ merit = \frac{\sum_{k=1}^{10} \frac{x_k}{10}}{10}$$

```
      gap = bIOM - index of merit)
      if(gap > 0)
         varXSet enhances IOM
      else if (gap<0)
         varXSet degrades IOM
      else
         varXSet has no effect
    next environment
    count ++
  next varXSet []
next XSet []
```

Figure 6.25: Template of the design of experiment 4

6.4.4.2 Results

We summarize the indices of merit that arise from the various combination and variant XSets as follows:

- Variant XSets in the stigmergic category show no requirement for particular order of events between the functionality to drop specific levels of pheromone, orientate, and flip between different internal states. All variant XSets where the order of these three primitive behaviours is ignored achieve similar results trends. However, stigmergic ant agents' ability to drop pheromone, as well as the ability to orientate, must occur before agent movement. This is a requirement which when not satisfied, the stigmergic model depletes. In fact, ant agent movements before orientation define random wandering swarms. Similarly, movements before dropping specific levels of pheromone implies that the actions of the ant agent at the time are not influenced by the levels of pheromone an ant agent drops at the time. In fact, ant agents can travel back to the locations they previously visited since there would not be any sign that the location is repulsive at the moment, resulting in the emergence of sub-optimal solutions. We therefore conclude that the order of events in the stigmergic category has a bearing on the indices of merit of the XSets thereof.

- Likewise, ant agent movements in the message passing category must occur after successful sharing of vectors, ant agent orientation, and detection of any target indicators around. Placing any of these functionality after ant agent movement defines otherwise random wandering swarms as well. However in this case, orientation must occur after sharing of vectors, otherwise it would not be informed by the views of neighbouring ant agents at the moment. We therefore also conclude that the order of events in the message passing category has a bearing on the indices of merit of the XSets thereof.

- What stands out in this experiment is that ant agent movement is strictly conditional. Ant agents must only make movement steps after successful orientation. As a result , there is insufficient evidence to support the null hypothesis which states that order has no effect to the indices of merit of XSets.

The next experiment investigates the effects of pheromone dissipation to the indices of merit in the stigmergic category. Note that the message passing model is ignored in this next experiment because it does not support pheromone mediated communication.

6.4.5 Experiment 5: Effects of pheromone dissipation

The last experiment we administer assesses the effects of pheromone dissipation to the indices of merit of the stigmergic XSet. The question we answer in this respect is: does pheromone dissipation influence the indices of merit we see in stigmergic XSets?

We hypothesize that pheromone dissipation has no effects on the indices of merit we achieve in stigmergic XSets. This hypothesis is measured by comparing the indices of merit of the benchmark XSet (where pheromone dissipation controls are not supported) with the indices of merit of the same benchmark XSet when pheromone dissipation controls are supported at different dissipation rates.

Title Investigating the effects of pheromone dissipation to the indices of merit of stigmergic XSets.

Hypothesis pheromone dissipation has no effect on the indices of merit of stigmergic XSets.

Dependent variable : indices of merit of the stigmergic XSet when pheromone dissipation is supported.Independent variables : dissipation rates (1% , 3%, 5%, 7%, 9%), time in simulation, targets setups, control levels, replications.

Controlled variables: environment size ((100×100 grid), position of the starting point (fixed), Centre of targets - centre of the environment, agent density (5000), composition of XSets (as used in the first experiment), sequences of primitive behaviours in XSet (as in first experiment), dissipation controls (ignored), radius of multiple targets (equal in all polygons), number of pheromones supported (maximum of 5), number of vectors supported (four).

Procedure - The algorithm below summarizes the procedure we follow to complete this experiment.

```
foreach dissipation rate
   foreach environment
      repeat
```

Control level	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
speed										
quality										
delivery										
delays										
Shannon										
cumm. IOM	$x_1+=IOM_{1,i}$	$x_2+=IOM_{2,i}$	$x_3+=IOM_{3,i}$	$x_4+=IOM_{4,i}$	$x_5+=IOM_{5,i}$	$x_6+=IOM_{6,i}$	$x_7+=IOM_{7,i}$	$x_8+=IOM_{8,i}$	$x_9+=IOM_{9,i}$	$x_{10}+=IOM_{10,i}$

```
       until 10 replications
```

$index\ of\ merit = \frac{\sum_{k=1}^{10} \frac{x_k}{10}}{10}$

gap = bIOM - $index\ of\ merit$)

```
       if(gap > 0)
          varXSet enhances IOM
       else if (gap<0)
          varXSet degrades IOM
      else
         varXSet has no effect
    next environment
next dissipation rate
```

Figure 6.26: Template of the design of experiment 5

6.4.5.1 Experiment Design

Figure 6.26 summarizes the procedure we follow in order to complete this experiment. The key dependent variable we measure are similarly the indices of merit that are achieved when the best performer stigmergic XSets is extended to support pheromone dissipation. We defined pheromone dissipation in section 3.2.5.9 as pheromone updates through evaporation and diffusion. We also define pheromone evaporation as a process whereby a certain percentage of the levels of pheromone that are held on each location of the environment is lost without ant agents or user intervention. On the other hand, pheromone diffusion was defined in section 3.2.5.10 as a process whereby a certain percentage of the levels of pheromone that are held on one location of the environment spills over to other locations without ant agents or user intervention.

The main independent variable of this experiment are the dissipation rates we support. Time in simulation, control levels, and targets setups are also important independent variables of this experiment. We assess the effects of pheromone dissipation using arbitrary dissipation rates in order to prove the concept. In particular, we compare the indices of merit that arise when the following dissipation rates are considered : 0% (the benchmark case), 1%, 3%, 5%, 7%, and 9%. From observation, these dissipation rates are a sufficient sample to reveal the effects sought. The rest of the variables of this experiment are controlled as indicated in the template presented in Figure 6.26.

6.4.5.2 Results

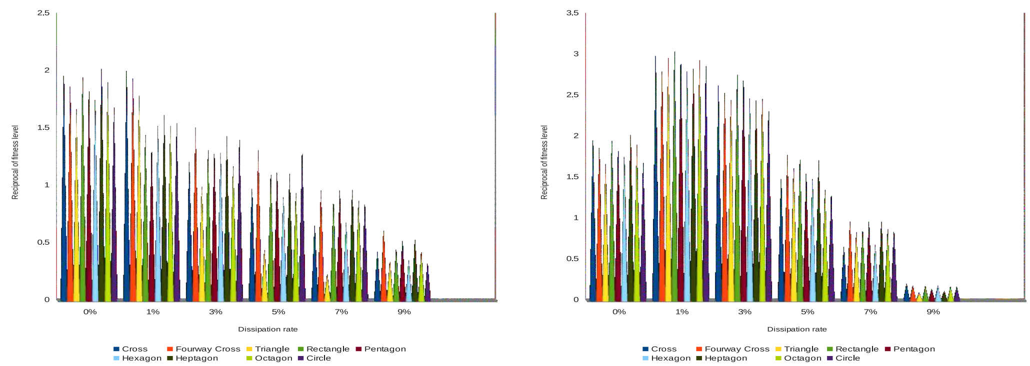
Besides the benchmark test, five other tests arise in which swarms of ant agents are deployed to path find or locate multiple targets when pheromone dissipation controls are considered at five different rates. The performance gaps between the benchmark indices of merit and what we observe when different dissipation rates are considered provide the results of interest in this experiment.

Figure 6.27 compares the variation we see in the indices of merit when 5000 ant agents are deployed and allowed to score performances over 10000 iterations. Precisely, Figure 6.27(a) reflects the general trends that arise in the indices of merit when one dissipation factor is considered at different dissipation rates (evaporation only or diffusion only). Then, Figure 6.27(b) indicates the trends that arise in the indices of merit when both dissipation factors are simultaneously considered (pheromone evaporation and diffusion).

We make the following observations regarding the outcomes of these evaluations:

- Scenarios in which one dissipation factor is considered deplete the stigmergic model as we increase the dissipation rates. In particular, consideration of pheromone evaporation on its own destroys the paths that arise. It creates paths with broken links in between, paths that would promote the emergence of local maximas. Higher dissipation rates simply worsens these effects. On the other hand, consideration of pheromone diffusion alone would unconditionally widen the paths thereof, thereby promoting environment saturation. Similarly, higher dissipation rates would merely speed up these effects. In the latter case, ant agents would eventually revert back to the random wandering mode.

- When appropriate dissipation rates are used, simultaneous consideration of pheromone evaporation and diffusion upgrades the indices of merit thereof. However lower dissipation rates achieve better outcomes because evaporation would, in fact, eliminate suboptimal paths off the



(a) Effects of single dissipation (b) Effects of both evaporation and diffusion

Figure 6.27: Effects of dissipation to fitness levels

environment while on the other hand diffusion smooths those paths that are strong and well trodden. We observe that the choice of which dissipation rate to use at a time is a function of agent density, the amount of pheromone each ant agent is allowed to drop at a time, as well as the size of the environment we use. High dissipation rates are detrimental to this model because evaporation would wipe out all the levels of pheromone that are on the environment before they are useful to the swarms (Dorigo et al., 1999). On the other hand, diffusion would quickly spread these levels of pheromone all over the environment, thereby rubbing off any paths and saturating the environment.

- We also observe that evaporation and diffusion augment each other in support of the creation of narrower, smoother and stronger paths between the starting point and the targets. When evaporation eliminates sub-optimal paths, diffusion smooths the remaining trails. We conclude that generally, although pheromone dissipation does not directly cause emergent behaviour, it enhances the quality of the emergent products that arise when appropriate dissipation rates reconsidered. We therefore lack of sufficient evidence to support our hypothesis which states that pheromone dissipation controls have no effects.

6.5 Conclusion of the chapter

This chapter firstly gave a brief overview of the problems that are addressed in the previous chapters with the goal of placing the purpose of this chapter in the thesis. Precisely, we stated our motivation for investigating best path finding XSets for abilities to solve emergent problems beyond path finding, and proposed a case study scenario of multiple targets location.

The chapter went on to define the multiple targets location problem as well as the evaluation environments on which we evaluate ant agents' abilities to locate multiple targets.

The biggest chunk of the chapter presented experiments, experiment designs, results, and the observations thereof. The first experiment compared the indices of merit of path finding XSets to the indices of merit that arise when multiple targets are considered. Generally:

- we found out that there is no evidence to reject the null hypothesis that there are no significant relationships between the indices of merit that arise when stigmergic XSets are used for path finding purpose and the indices of merit that are yield when the same XSets are used for multiple targets location. Thus, the relationships we see between these data sets are merely by chance because these XSets are, rather, problem specific. On the contrary, there is sufficient evidence to refute the null hypothesis in the message passing category, suggesting significant relationships between the two data sets. Thus, the relationships we see between these data sets are not by chance because these XSets are, in fact, problem independent.

- we also found out that stigmergic XSets are sensitive to the number of the multiple targets that are presented at the time, as well as sensitive to the radius of the region that is covered by these multiple targets, while message passing XSets are not.

- Stigmergic F-tests implied significantly different variations between the results yield during path finding and those reported when multiple targets are considered, while message passing outcomes are opposite. Thus, while there is insufficient evidence to refute the null hypothesis in the stigmergic category, message passing F-test supported the alternative hypothesis.

- All P - tests indicated lack of evidence to reject the null hypothesis, implying that the mean performances we observe in the indices of merit in both categories follow similar trends.

The second experiment evaluate the effects of agent density to the indices of merits we observe in XSets. No apparent evidence arose with which we could support the null hypothesis in this regard. Rather, we concluded that agent density has a bearing on the indices of merit that arise in different scenarios.

In the third experiment, we investigated the influences of the discrete primitive behaviours that are included in the composition of XSets to the indices of merit thereof. Similarly, sufficient evidence was available with which to support the alternative hypothesis that the discrete primitive behaviours that are included in XSets are critical.

The fourth experiment assessed the effects of the order in which primitive behaviours are arranged in XSets to the indices of merit thereof. Likewise, agent movement was noted to conditionally occur after agent orientation and information update in both cases. Thus, sufficient evidence to refute the null hypothesis was presented in both cases.

The fifth and last experiment investigated the effects of pheromone dissipation to the indices of merit of stigmergic XSets. Consideration of evaporation or diffusion in isolation depleted the stigmergic model. In addition, high dissipation rates are detrimental to the model. However positive effects are observed when these two dissipation factors are simultaneously considered. We therefore gathered sufficient evidence to support the belief that pheromone dissipation has a bearing on the indices of merit we observe in stigmergic XSets.

The value of this chapter is further emphasized by five contributions it makes to the thesis and the body of knowledge. These contributions are as follows:

- The discovery we made regarding stigmergic XSets being useful for a wider range of problem domains when the radii of the targets regions are smaller motivates further investigations and more practical applications of these XSets approach. We therefore create new research directions in the field.
- The invention of the message passing model and the discovery that this model is insensitive to multiple targets setups may potentially inspire the development of more commercial emergent object assemblers in the future. This invention also creates new knowledge in the field.
- Experiments 2 to 5 investigated the effects of some of the parameters one should consider when designing ant based systems. In particular, the discovery that agent density is a function of the size of the environment, distance between the starting point and targets, as well as a

function of the amount of pheromone each ant agent releases in each step (in the case of stigmergic ant agents) is a mathematical challenge in future works. This observation presents a research question in which a mathematical model is sought which determines the appropriate agent densities thereto.

- The discovery we made that particular primitive behaviours are indispensable in particular XSets, and that some primitive behaviours occur in particular sequences, brings us closer and closer to solving the general ant agent control problem. This is useful information towards prescribing particular XSets as languages for programming ant agents in swarms to allow deliberate engineering of desired emergent behaviour.

- The definition of the evaluation environments on which swarms of ant agents are assessed for different abilities, and how emergent properties are incorporated into the simulation system are innovative. Similar architectures can be adopted in related researches in the future.

Chapter 7

Conclusion

This chapter summarizes the research we conducted in this thesis. First, section 7.1 provides an overview of the chapters we presented. Then, section 7.2 gives a summary of the key observations we make. The contributions of the thesis to the broader board of knowledge are presented in section 7.4. We wrap up this chapter and the thesis in section 7.5, highlighting the potential future directions of this work.

7.1 Summary of chapters

The key focus of the thesis has been the identification of XSets of primitive behaviours which characterize a language for programming swarms of ant-like devices towards predictable emergent behaviour. To arrive at the conclusions we make, the following chapters addressed various sub problems as follows:

- the problem statement, the background to the research problem, the motivation for addressing the same research problem, our strategy, as well as the envisioned contributions of this work to the board of knowledge, were presented in the first chapter of the thesis.

- Chapter 2 investigated four aspects of this thesis. First, we looked at the various agent control models that have been described in the past, mainly focusing on agent interaction and orientation techniques, as well as how each agent successfully moves from one location of the environment to another. We got inspiration from these reviews regarding which controls to consider for our ant agents. Secondly, the chapter investigated the common parameters of emergence that have been proposed in most agent control systems. Parameters of emergence are factors which influence swarm performance. Reviews in this respect motivated our choices of the variables we investigated in chapter 6. The third aspect we investigated in Chapter 2 are the measures of emergence that are used to quantify emergent behaviour in most agent systems. Five measures of emergence were derived from these reviews which we recommended for determining the indices of merit of XSets. The last aspect we investigated in the literature review relates to the forms of target configuration that are common in multiple targets scenarios. The configuration of regions at which we defined multiple targets was motivated by these reviews.

- the primary goal of Chapter 3 has been the identification and characterization of the discrete ant agent primitive behaviours one by one, as well as the provision of semantics with which each primitive behaviour is interpreted in computational terms. In identifying the particular primitive behaviours, we assumed a case study scenario of swarms of ant agents in which path finding and path following behaviour was sought. As a result ten discrete primitive behaviours were identified as possible building blocks of ant agent behaviours over time. These primitive behaviours are the building blocks of the XSets that are proposed in this thesis (see Chapter 3 for the mnemonics and semantics of these ten primitive behaviours).

- the purpose of Chapter 4 has been threefold. First, it developed a strategy for putting primitive behaviours together into XSets that can allow particular emergent behaviour to occur. Genetic programming processes were proposed as a search strategy for sufficient XSets for path finding purposes. Then, a strategy for assessing the validity of the XSets that arose was described. The indices of merit which we use to rate XSets are determined using five measures of emergence that are presented in this chapter. Chapter 4 also presented an experiment which investigated the indices of merit of the XSets that built particular genetic populations and identified the best performer path finding XSets in each category. These are the same XSets we verified and validated throughout the rest of the chapters of this thesis.

- The concerns of Chapter 5 have been to establish the relationships that exist between the sets of measures of emergence that arose when best performer XSets were used for path finding purposes. Kolmogorov-Smirnov tests for normality, Correlation analyses, F-Tests and analyses of variances, as well as T -Tests and comparisons between means, revealed the validity of the best performer XSets as appropriate dictionaries for allowing deliberate engineering of path finding behaviour.

- Chapter 6 was developed with the goal of evaluating possibilities of applying the best performer XSets to different problem domains, particularly multiple targets location. Five experiments which evaluated the effects of different variables to the indices of merits of these XSets were administered. Generally, the results we presented show that the stigmergic XSets are sensitive to the radius of the region covered by the targets. On the contrary, the message passing counterparts are insensitive to this requirement. The same results also demonstrated some influences arising from the agent density used, the size of the environment, the size of the target, dissipation factors, the distance between targets and the starting point, as well as the sequence in which primitive behaviours are arranged in the XSets. However, the visual screenshots of the performances of swarms of ant agents that used these XSets show successful and evenly distributed arrivals of ant agents on targets in both categories.

7.2 Observations

The general problem that was addressed in this thesis is the identification of XSets that can allow deliberate engineering of specific emergent behaviour. These investigations were restricted to resolving five sub-problems which we responded to as follows:

1. **Identification of ant agent primitive behaviours** *(what are the low level activities of ant agents that can be used to describe the domain of primitive behaviours that allow particular emergent behaviour to occur?)* Chapter 3 responded to this question. It identified ten discrete ant agent primitive behaviours for this purpose. In particular, stigmergic ant agents must orientate - (MvH), and drop specific levels of pheromone in each step - (Drp). These levels of pheromone can evaporate -(Evp) or diffuse - (Dfs) at particular dissipation rates. Message passing ant agents must share direction vectors and orientate - (MsP). The resultant direction vectors can be optionally normalized - (Nrm). However these ant agents must possess abilities to detect target indicators -(PtV) and convert these to vector information where possible. All ant agents must make informed movements - (MvP), and flip between different internal states when it becomes necessary - (StS).

However there are times when ant agents are required to do nothing -(*NOp*).

2. **Creating XSets of primitive behaviours** *(how do we create valid XSets of primitive behaviours from discrete ant agent activities which can summarize collections of ant agents actions over time?)*. Chapter 4 addressed this question. The primitive behaviours that were identified in Chapter 3 served as inputs to the genetic programming system we proposed. A novel mechanism for generating the initial genetic population of XSet was proposed and the parameters of the evolution process were explained. New XSets were evolved in three different ways in order to ensure diversity in new genetic population of XSets as we moved down the generations. In particular, 5% of the XSets were regarded as elite. These were randomly selected from the old generation and promoted to the new generation. The other 80% of the XSets in the new genetic population were created using genetic crossover between parent XSets. These parent XSets were selected from the old genetic population using specific selection algorithms (tournament or roulette wheel selection). The remaining 15% of the XSets were created using mutation operation (see chapter 4 regarding how different XSets are created). The genetic population thereof served as the repository from which we searched for best performer XSets for the path finding problem in each generation.

3. **Evaluation of XSets of primitive behaviours** *(how do we quantify the extent to which emergent behaviour is manifest as a result of using a particular XSet for path finding purposes?)*. Chapter 4 addressed this sub-problem as well. Five measures emergence were proposed, and the procedures through which we assess each of these measures of emergence were presented. In particular, the speed of emergence was assessed as the time it takes ant agents to converge as a result of using

a specific XSet. Then, quality of emergence established the frequencies with which ant agents successfully arrived at their targets within a set time frame. These qualities of emergence were then expressed as percentages of the population of ant agents that were deployed, thereby defining average delivery rates and the throughput of the swarms within a set time limit. Average end-to-end delays considered the time it takes individual ant agents to travel between the starting point and the targets or vice verse. A strategy for determining sample sizes was presented. Sampling was required when we picked ant agents to track average end-to-end delays. The last measure of emergence we assessed is built on Shannon's measurement theory which evaluated the degrees of uncertainty that are associated with an ant agent's path selection decisions. Sampling was also required in this regard. The measures of emergence we achieved in each case were weighted and put together in order to determine the indices of merits of the XSets thereof (see chapter 4 regarding how measures of emergence and fitness levels are determined).

4. **Relationships between measures of emergence**: Chapter 5 addressed this sub - problem. It first tested the sets of results we reported for normality before evaluating the correlation coefficients that exist between these sets of measures of emergence. It also analyzed the variances we saw in these sets of measures of emergence and justified any similarities in the fitness trends thereof. This chapter consolidated the relationships we saw between the sets of measures of emergence by comparing the means performances in these sets of measures of emergence. Generally, slim chances were observed that the measures of emergence we observed occurred by chance. Most of the sets of measures of emergence thereof revealed statistics which demonstrated proof of common origins, thus validating the XSets from which the measures of emergence were taken.

5. **Application of path finding XSets to multiple targets location** : Chapter 6 addressed this sub-problem. Five experiments were conducted in this chapter which assessed the application of best performer XSets to the multiple targets location problem. Most importantly, the first experiment established that the XSets we proposed are generally problem independent when targets are appropriately set. However although multiple targets location was achieved in most cases, the stigmergic XSet demonstrated sensitivity to the size of the targets. Other factors were also investigated which influence the indices of merit of these XSets for multiple targets location. In particular, agent density was noted to influence the indices of merit of XSets in both categories. Similarly, most individual primitive behaviours that are included in these XSets are indispensable. The XSets are often arranged in specific sequences, where agent movement controls come last in each sequence. Pheromone dissipation factors, as well as vector normalization, were noted as enhancing factors rather than causal primitive behaviours.

The following observations are drawn from the results we presented in all the chapters of this thesis:

- stigmergic ant agents do not require any direct language of communication. They can only sense pheromone information which is placed on the environments by the same ant agents as they forage. There is no specific agent-held information that is directly exchanged between ant agents. On the other hand, message passing ant agents carry direction vectors around, communicating these direction vectors with immediate neighbours one-on-one. We observe an advantage of stigmergic ant agents over the message passing counterparts in preserving confidentiality. This is a paramount feature which we can exploit when security systems are required (an observation that arises from the literature review that is presented in chapter 2).

- like any other stochastic processes, probabilistic path selection policies which characterize stigmergic ant agent orientation (roulette wheel selection) have a detrimental effect. This is because there are chances that ant agents can derail from good paths. On the contrary, message passing swarms create vector fields with deterministic hints towards desired targets. Once the vector field is in place, ant agent follow these vector fields all the way to the targets. A disadvantage is noted in the stigmergic model in that stochastic movements may have a negative feedback effect on the performance of the swarm. This is also the reason why we observe better indices of merits in the message passing model than we see in the stigmergic category.

- message passing ant agents are always in a competition with one another. Every ant agent works towards recognizing the knowledge of its neighbours, while at the same time keeping record of its own beliefs and confidence in the direction vectors it is following. As a result, the resultant vectors that are yield in each step are a compromise between the collective knowledge of the neighbours and the ant agent's own perceptions. In reaching this consensus, relatively less confident ant agents often gain, while highly confident ant agents degrade in confidence. We note a disadvantage of the message passing model in that agent autonomy is grossly compromised. Worse still, the information held in less confident ant agents may even be eroded off with time in simulation. There is therefore a danger that possibly valuable historic information that may be held in less confident agents can be lost.

- although pheromone dissipation in the stigmergic category is noted as merely an enhancing factor to swarm performance, it is an important factor to consider when we study stigmergic ant agent swarms. This is because it improves swarm convergence, it aids mass recruitment, and hence ensures the emergence of quality paths. However appropriate dissipation rates must be known upfront. A mathematical model is required which determines appropriate dissipation rates as a function of agent density, size of the environment, and the distance between the starting point and the targets.

- although agent density is a key parameter of emergence in both categories, we are faced with the challenge to present a mathematical model for determining appropriate agent densities in particular scenarios. Both stigmergic and message passing swarms degrade in performance when insufficient agent densities are used. In stigmergic swarms in particular, shared memories would not build. On the other hand, communication gaps arise in the message passing category. Stigmergic swarms would similarly degrade when too many ant agents are used because the excess levels of pheromone thereof would saturate the environment. A threshold agent density is therefore required in both categories. Again a mathematical model is required which determines appropriate agent densities as a function of the size of the environment, the distance between the starting point and the targets, as well as a function of the dissipation properties in place.

7.3 Comparisons with existing models

The XSets approach introduced an original and novel ant agent design paradigm which allows deliberate engineering of specific ant agent based emergent behaviour. We make the following observations regarding its comparisons with similar agent languages and representations for swarm coordination in the literature:

- Non interactive agent languages or agent control representations have mainly relied on mathematical and physics laws. Precisely, these interactions are often based on equations (Montes De Oca et al., 2005), calculus (Sarfati, 2001), matrices (Harris, 2007), virtual forces (Spears et al., 2004a, 2004b, 2005.; Balch and Arkin, 1999.; Azzag et al., 2007.; Beckers et al., 1989.; Bayazit et al., 2002.; Lua et al., 2005.; Parrish et al., 2002), geometry (Trofimova et al., 1998.; Ngo et al., 2005), or specifically vector geometry (Ngo et al., 2005.; Nasipuri and Li, 2002.; Wu et al., 2005). In most cases, agents possess large memory capacities to recall events (Wehmer et al., 2006.; Cordon et al., 2002.; Mullen et al., 2009). The XSets approach we propose differs in that it emphasizes on the design of simple ant agents whose interaction languages require basic agent memory, a feature which promotes the benefits of simplicity, robustness, parallelism, decentralization, automated optimization of solutions, and capacity to handle dynamic situations (Werfel et al., 2006).

- Representation of stigmergic concepts for designing software architectures to allow acting of agents in order to yield suitable control behaviours is not a new research area (Valckenaers et al., 2001). What stands out in most stigmergic representations is emphasis on the environment as a key parameter of emergence on which to accumulate information about ongoing activities of the agent society (Negulescu et al., 2006.; Haasdijk et al., 2013.; Seevinck and Edmonds, 2008.; Mason, 2002.; Bredeche et al., 2012). Our stigmergic XSets equally emphasize on the environment as a key component and holder of the key parameters of the system.

- Common interactive agent languages or control representations are modelled on the behaviours of living organisms such as cells (Xi et al., 2005), birds (Reynolds, 1999), DNA sequences (Reif, 2002), bees (Reynolds, 1987), or ants (Chibaya and Bangay, 2007). Some of these models allow the exchange of information between agents one-on-one.

In such cases, messages are often shared in the form of memory blocks (Nasipuri and Li, 2002), paths histories (Rajbhupinder et al., 2010.; Trianni and Dorigo, 2005.; Rodriguez et al., 2007), or coordinates of key points (Montes De Oca et al., 2005). However these characteristics require the agents to possess large memory capacities. The XSets approach we propose emphasizes on the design of simple and naive ant agents whose low level actions require basic agent memory.

- Attempts to create agent languages with full syntax, vocabulary, and semantics have been presented (Nagpal et al., 2002). Most of the outcomes in the literature are based on the growing point (Butera, 2002) and origami shape theories (Nagpal et al., 2003). However the domain of problems that can be addressed when these theories are proposed is limited because agent independence is compromised. The design of the XSets we proposed has the resolution of a wider task domain in mind. The results we presented in the earlier chapters of this work connote XSets that are problem independent.

- Computational techniques for evolving representations of agent design as genes are not new either (Poon and Maher, 1996). Mechanisms for building behaviour and structure search spaces through repeated insertion of evolved elements into the pool of building blocks have been proposed (Poon and Maher, 1996). In these, genetic cycles are administered in which basic genes are combined to form offspring genotypes that are re-introduced in the pool to develop design solutions for the next generation. The mechanisms in which we develop and evolve best performer XSets are similar. However our work differs in that we only emphasize of the evolution of ant agent behaviour search spaces.

- The desire to determine sets of low-level components that can reliably generate desired systems has been ongoing (Polack and Stepney, 2005.; Stepney et al., 2007). Our design of XSets extends these desires. This work went a step further and combined the sets of low-level components with meta information which spelt out how and when the low-level components thereof are used.

We emphasize that this work forms a baseline upon which more studies and experiments are required. The outcomes of those extra tests would help in polishing up the methodology before rigorous comparisons with traditional approaches are recommended.

7.4 Contributions

The main distinguishing feature of this thesis is that specificity we emphasize on with regard to the composition of XSets which allow particular emergent behaviour to occur. A number of contributions emanate, both from an academic, practical, and general point of views. From an academic point of view, we make the following contributions:

- Successful identification of the primitive behaviours which characterize ant agents' behaviours at individual levels, which give rise to particular forms of emergent behaviour at swarm levels, is a big milestone in the study of ant systems. Knowledge of what each ant agent does as an individual allows us to deliberately engineer predictable swarm outcomes. We therefore create relevant knowledge in the field, particularly to the benefit of future researches and studies in the area.

- We presented a mechanism in which primitive behaviours are put together into XSets which characterize sufficient dictionaries for controlling swarms of ant agents. This mechanism is an inspiration to the development of useful emergent based object assemblers in the future.

- A number of the algorithms we proposed in this work are innovative. In particular, ant agent interaction and orientation techniques which we presented in Chapter 3 are novel mathematical models in the ant agent metaphor. A special mention goes to the determination of the attractiveness of a location around a stigmergic ant agent, as well as the determination of a resultant vector in the message passing model. We therefore create relevant content with which the ant agent programming problem is further addressed.

- The degree of emergence that arises in ant agent systems has often gone unnoticed. We proposed five innovative measures of emergence with which we determine the extent to which emergent behaviour is manifest as a result of using a particular XSet. The properties of most of these measures of emergence make them suitable for verifying other forms of emergent configurations. New research opportunities arise in which these measures are further validated.

- Validation of XSets of primitive behaviours using correlation analyses, comparisons between the mean fitness levels, as well as analyses of variances, is creative. To the best of our knowledge, this is the first time these analyses are applied to the ant programming problem. Similar comparisons can be used to verify other agent models as well. We therefore create relevant knowledge in the field.

On the practical side, we present the following contributions:

- The ability to explicitly specify ant agents' primitive behaviours at individual levels, and combine these into XSets of primitive behaviours that are useful at swarm levels, has direct relevancy to many fields in science. Swarms of simulated ant-like devices such as nanites, amorphous devices, or MEMS devices, can be deployed in specific simulation environments using similar XSets in order to create commercially attractive outcomes. The results of this work will thus likely promote industrial and commercial application of ant system metaphors.

- Our emphasis on specificity, both in terms of the XSets of primitive behaviours that are required by different ant agent swarms, as well as specificity in the form of emergent behaviour sought, changes the way we see, and think of the consequences of upcoming sciences such as nanotechnology. Generally, nanotechnology is feared that, one day, nanites may aggregate into unpredictable emergent formations that are disastrous to nature and life. This work provides inspiring XSets which can be tailored to guarantee predictable nanite outcomes.

From a general point of view, we make the following contributions:

- Although the thesis does not solve the very general agent programming problem, it provides a working baseline upon which further investigations in the field may arise. Therefore, this work provides a solid foundation for investigations aimed at identifying general agent control primitive behaviours and a language thereof.

- Many ant systems that exist in the literature do not explicitly present the white-box side of how agent activities are implemented. As a result, the domain of ant based solutions that are available in the literature is currently limited, especially for commercial recommendations. Our emphasis on specificity, and explicit description of the routines that characterize ant agent behaviours at individual levels may inspire the development of a wider range of ant based solutions.

7.5 Future work

There are a number of immediate directions of possible researches that emanate from this research. Four of these are outstanding, namely:

- it will be desirable to investigate and evaluate XSets of primitive behaviours of swarms of ant agents that make use of other interaction mechanisms such as leader following, or ant agents with abilities to recall previous experiences. The union of these XSets with the XSets we proposed in this work would create more diverse genetic populations in which a wider range of problems can be solved. That union may also support the evolution of more useful hybrid XSets.

- although specificity is of great value in this work, it will be desirable to investigate possibilities of prescribing XSets for producing flexible emergent behaviour.

- effort must be made to create and evaluate XSets that can be used to coordinate ant agents in 3D environments. We believe that 3D solutions have more practical applications on the market today.

- effort must be made to build practical mathematical models with which we can calculate threshold agent densities, as well as appropriate dissipation rates upfront. Such mathematical models would surely serve as inputs to many future researches in the field.

References

ABELSON, H., ALLEN, D., COORE, D., HANSON, C., HOMSY, G., KNIGHT, T., NAGPAL, R., RAUCH, E., SUSSMAN, G.J. AND WEISS, R. **2000**. Amorphous. *In communications of the ACM.* Volume 43. Number 5. Pages 74 - 82.

ADAMIC, L.A., LUKOSE, R.M., PUNITANI, A.R.. AND HUBERMAN, B.A. **2001**. Search in power-law networks. *In Physical Review E:* Volume 64. Number 4. Pages 04135-1 - 04135 - 8.

ALCALÁ, R., CASILLAS, J., CORDÓN, O. AND HERRERA, F. **2001**. Improvement to the cooperative rules methodology by using the ant colony system algorithm. *In Mathware and Soft Computing.* Volume 8. Pages 321 - 335.

ALEKSIEV, A.S., SENDOVA-FRANKS, A.B. AND FRANKS, N.R. **2007**. Nest moulting in the ant Temnothorax albipennis. *In the journal for Animal Behaviour.* Elsevier Science Ltd. School of Biological Sciences, University of Bristol and School of Mathematical Sciences, University of the West of England, Bristol. Volume 74. Issue 3. Pages 567 - 575.

ANDREW, S.P. **1978**. An Investigation of the mechanisms underlying nest construction in the mud wasp paralastor species (hymenoptera:

eumenidae). *In the journal for Animal Behaviour.* Zoology Department, University of Sydney. Volume 26. Pages 232 - 240.

ANDERSSON, N., BROBERG, A., BRÄNBERG, A., JANLERT, L.E., JONSSON, E., HOLMLUND, K. AND PETTERSSON, J. **2002**. Emergent Interaction : A pre-study. A technical report submitted to the Umeå Center for Interaction Technology (UCIT) and Ericsson Erisoft collaboration.

AZZAG, H., VENTURINI, G., OLIVER, A. AND GUINOT, C. **2007**. A hierarchical ant based clustering algorithm and its use in three real-world applications. *In the European Journal of Operational Research.* Elsevier Science Ltd. Volume 179. Issue 3. Pages 906 – 922.

BABAOGLU, O., CANRIGHT, G., DEUTSCH, A., DI CARO, G.A., DUCA TELLE, F., GAMBARDELLA, L.M., GANGULY, N., JELASITY, M., MONTEMANNI, R., URNES, T. AND MONTRESOR, A. **2006**. Design patterns from biology for distributed computing. *In ACM Transactions of Autonomous and Adaptive Systems (TAAS).* Volume 1. Issue 1. Pages 26 - 66.

BAI, L., EYIYUREKLI, M. AND BREEN, D.E. **2008**. Self-Organizing Primitives for Automated Shape Composition. *In IEEE International conference on shape modelling and applications.* Drexel University. Pages 147 - 154.

BALCH, T. AND ARKIN, R.C. **1999**. Behaviour-based Formation Control for Multi-robot Teams. *In IEEE transactions on robotics and automation.* Georgia. Volume 14. Issue 6. Pages 926 - 939.

BAVAUD, F., CHAPPELIER, J.C. AND KOHLAS, J. **2005**. An Introduction to information theory and applications. Course Notes. Version 2.04. Pages 1 - 293.

BAYAZIT, O.B., LIEN, J.M. AND AMATO, N.M. **2002**. Roadmap-Based Flocking for Complex Environments. *In proceedings of the 10th Pacific Conference on Computer Graphics and Applications, Beijing, China.* . Pages 104 -113.

BAYAZIT, O.B., LIEN, J.M. AND AMATO, N.M. **2005**. Swarming Behaviour Using Probabilistic Roadmap Techniques. *In Proceedings of the International Workshop on Swarm Robotics. Lecture Notes in Computer Science.* Volume 3342. Pages 112 - 125.

BEAL, J. **2005**a. Programming an Amorphous Computational Medium. *In Lecture Notes in Computer Science.* Springer-Verlag Berlin Heidelberg. Volume 3566. Pages 121 - 136.

BEAL, J. **2005**b. Amorphous Medium Language. *In AAMAS'05.* Netherlands. http://web.mit.edu/jakebeal/www/publications/Ismas - final.pdf.

BECKERS, R., GOSS, S., DENEUBOURG, J.L. AND PASTEELS, J.M. **1989**. Colony Size, Communication and Ant Foraging Strategy. *In PSYCHE.* Unit of Behavioural Ecology, Universite Libre de Bruxelles, 1050 Bruxelles, Belgium. Volume 96. Number 3 - 4. Pages 239 - 256.

BELANI, R.V., DAS, S.M. AND FISHER, D. **2002**. One-to-one modelling and simulation of unbounded systems: experiences and lessons. *In the proceedings of the 2002 Winter Simulation Conference.* Volume 1. Pages 720 - 724.

BONABEAU, E., DORIGO, M. AND THERAULAZ, G. **1999**. Swarm Intelligence: From Natural to Artificial Systems. *Oxford University Press.*

BREDECHE, N., MONTANIER, J.M., LIU, W. AND WINFIELD, A.F.T. **2012**. Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents. *In the journal for Mathematical and Computer Modelling of Distributed Systems.* Volume 18. Number 1. Pages 101 - 129.

BROWN, S., ZULUAGA, M., ZHANG, Y. AND VAUGHAN, R. **2005**. Rational Aggressive Behaviour Reduces Interference In A Mobile Robot Team. *In the proceedings of the 12th International Conference on Advanced Robotics* (ICAR'05). Pages 741 - 748.

BULLOCK, S. AND CLIFF, D. **2004**. Complexity and Emergent Behaviour in ICT Systems. *Technical report HP-2004-187, Semantic and Adaptive Systems.* HP Labs.

BURGESS, M.G. **2009**. Sub-optimal pit construction in predatory ant lion larvae (Myrmeleon sp). *In the Journal of Theoretical Biology.* Volume 260. Pages 379 - 385.

BURKE, E. AND KENDALL, G. **1999**. Applying Simulated Annealing and the No Fit Polygon to the Nesting Problem. *In Advanced Topics in Artificial Intelligence - LNCS.* Volume 1747. Pages 453 - 464.

BUTERA, W. **2002**. Programming a Paintable Computer. *A PhD thesis submitted to the school of architecture* and *planning at MIT.*

CAICEDO, A., MONZANI, J.S. AND THALMANN, D. **2001**. Communicative Autonomous Agents. *In proceedings of the IFIP TC5/ WG5.10 DEFORM'2000 Workshop and AVATARS' 2000 workshop on deformable avatars. Available on-line on http://citeseerx.ist.psu.edu/ viewdoc/download?odi=10.1.1.85.3333&rep=rep1 & type=pdf.*

CAO, Y.U., FUKUNAGA, A.S. AND KAHNG, A.B. **1997**. Cooperative Mobile Robotics: Antecedents and Directions. *In the journal for autonomous robots.* Volume 4. Pages 1 - 23.

CAVALCANTI, A., HOGG, T., SHIRINZADEH, B. AND LIAW, H.C. **2006a**. Nanorobot Communication Techniques: A Comprehensive Tutorial. *In IEEE ICARCV 2006 International Conference on Control, Automation, Robotics and Vision.* Pages 1 - 6.

CAVALCANTI, A. AND FREITAS, R.A. **2005**. Nanorobotics Control Design: A Collective Behaviour Approach for Medicine. *In IEEE Transactions on NanoBioScience.* Volume 4. Pages 133 - 140.

CAVALCANTI, A., SHIRINZADEH, B., HOGG, T. AND KRETLY, L.C. **2006b**. CMOS-based Nanorobot to Combat Cancer. *In WFM06 - COSNet Australian Workshop on Fluid Mechanics, A Complex Dynamical System. Available on-line on http://www.nanorobot design.com/papers/cmos.pdf.*

CAVALCANTI, A., SHIRINZADEH, B., FREITAS, R.A. AND KRETLY, L.C. **2007**. Medical Nanorobot Architecture Based on Nanobioelectronics. *In Recent Patents on Nanotechnology.* Volume 1. Number 1. Pages 1 - 10.

CERELLO, P., CHERAN, S.C., BAGNASCO, S., BELLOTTI, R., BOLANOS, L., CATANZARITI, E., NUNZIO, G.D., FANTACCI, M.E., FIORINA, E., GARGANO, G., GEMME, G., TORRES, E.L., MASALA, G.L., PERONI, C. AND SANTORO, M. **2010**. 3-D object segmentation using ant colonies. *In the journal of Pattern Recognition. Science Direct.* Volume 43. Pages 1476 - 1490.

CHAN, W.K.V. **2011**. Interaction metric of emergent behaviours in agent-based simulation. *In Proceedings of the 2011 Winter Simulation Conference.* Pages 357 - 368.

CHIBAYA, C. AND BANGAY, S. **2007**. A probabilistic movement model for shortest path formation in virtual ant-like agents. *In ACM International Conference Proceeding Series; Volume 226. And In the proceedings of the 2007 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries.* South Africa. Pages 9 - 18.

CHIRICO, U. **2004**. A Java Framework for Ant Colony Systems. *In ANTS2004: Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence*, Brussels. *Available on-line on http://www.ugosweb.com/Documents/jacs.aspx.*

CRUTCHFIELD, J.P. **1994**. The Calculi of Emergence: Computation, Dynamics, and Induction. *In the Physica D - Non-linear Phenomena (1994). A special issue on the Proceedings of the Oji International Seminar. Complex Systems — from Complex Dynamics to Artificial Reality.* Volume 75. Issues 1 - 3. Pages 11 - 54.

COLIN, A. **2006**. Ant Colony Algorithms: Solving optimization problems. *In Dr. Dobb's journal on http://www.ddj.com/dept/ai/191800178?pgno=1.*

CORDON, O., HERRERA, F. AND STÜTZLE, T. **2002**. A review of Ant Colony Optimization Metaheuristic: Basis, Models and New Trends. *In Mathware and Soft Computing.* Volume 9. Pages 141 - 175.

COUZIN, I.D. AND FRANKS, N.R. **2002**. Self-organized lane formation and optimized traffic flow in army ants. *In proceedings of the Royal society London. Behavioural Biological sciences.* Volume 270. Pages 139 - 146.

COUZIN, I.D., KRAUSE, J., JAMES, R., RUXTON, G.D. AND FRANKS, N.R. **2002**. Collective Memory and Spatial Sorting in Animal Groups. *In the journal for theoretic Biology.* Elsevier Science Ltd. Volume 218. Issue 1. Pages 1 - 11.

CRANEFIELD, S., NOWOSTAWSKI, M. AND PURVIS, M. **2000**. Modelling Agent Communication Languages. *In Network Computing Group, Institute for Information Technology, National Research Council of Canada. Available on-line on https://www.academia.edu/* 2977908/Modelling_Agent_Communication_Languages.

DASTANI, M., VAN DER HAM, J. AND DIGNUM, F. **2003**. Communication for Goal Directed Agents. *In Communication in Multi agent Systems - Agent Communication Languages and Conversation Policies.* Volume LNCS 2650. Pages 239 - 252.

DAVID, G. **2008**. Statnotes: Topics in Multivariate Analysis. *Retrieved on 15/08/2008 from http://www2.chass.ncsu.edu/garson/pa765/statnote.htm".*

DERBYSHIRE, D. **2009**. Getting it off their chest: Study reveals how ants talk to each other. *In mail-on-line. Science & Technology.* Retrieved from http://dailymail.co.uk/sciencetech/article-1137267/Getting-chest-study-reveals-ants-talk-other.html.

DETRAIN, C. AND DENEUBOURG, J.L. **2006**. Self-Organized Structures in a Super Organism: Do Ants "Behave" Like Molecules ?. *In Physics of Life Reviews.* Volume 3. Issue 3. Pages 162 - 187.

DE WOLF, T. **2007.** Analysing and Engineering Self-Organising Emergent Applications. *A PhD thesis submitted to the Katholieke Universiteit Leuven.* Belgium. Number D/2007/7515/61.

DHARIWAL, A., SUKHATME, G.S. AND REQUICHA, A.A.G. **2004**. Bacterium-inspired Robots for Environmental Monitoring. *In proceedings of the 2004 IEEE International Conference on Robotics and Automation.* New Orleans, LA. Pages 1436 - 1443.

D'HONDT, E. **2000**. Exploring the Amorphous Computing Paradigm. *An MSc Technical Report submitted to the department of Informatica.* Vrije Universiteit Brussel.

DI CARO, G., DUCATELLE, F. AND GAMBARDELLA, L.M. **2004**. AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks. *A technical report submitted to Dalle Molle Institute for Artificial Intelligence.* Number TR / IRIDIA-27-04-2004.

DIXON, C., WINFIELD, A.F.T., FISHER, M. AND ZENG, C. **2011**. Towards Temporal Verification of Swarm Robotic Systems. In Lecture Notes in Computer Science. Volume 6856. Pages 336 - 347.

DON, O. AND AMOS, M. **2007**. An ant-based algorithm for annular sorting. *In the IEEE Press and in proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC'07).* Singapore, September 25-28. Pages 142 - 148.

DORIGO, M. **1992**. Optimization, Learning and Natural Algorithms. *A PhD thesis submitted to the Dipartimento di Electronica, Politecnico di Milano, Milan, Italy.*

DORIGO, M., MANIEZZO, V. AND COLORNI, A. **1996**. The Ant System: Optimization by a colony of cooperating agents. *In the IEEE Transactions on Systems, Man, and Cybernetics, Part–B*, Volume 26. Issue 1. Pages 29 – 41.

DORIGO, M., DI CARO, G. AND GAMBARDELLA, L.M. **1999**. Ant Algorithms for Discrete Optimization. *In proceedings of Artificial Life.* Volume 5. Pages 137 - 172.

DORIGO, M. AND BLUM, C. **2005**. Ant colony optimization theory: A survey. *In Theoretical Computer Science.* Volume 344. Issues 2 - 3. Pages 243 – 278.

DORIGO, M., BIRATTARI, M. AND STÜTZLE, T. **2006**. Ant Colony Optimization : Artificial Ants as a Computational Intelligence Technique. *A technical report submitted to the Universite Libre de Bruxelles*, BELGIUM. Number TR/IRIDIA/2006-023.

DOWNING, H.A. AND JEANNE, R.L. **1988**. Nest construction by the paper wasp, Polistes: a test of stigmergy theory. *In the Association for the Study of Animal Behaviour, Department of Biology.* The College of Wooster, Wooster, Ohio 44691, U.S.A. and Department of Entomology, University of Wisconsin, Madison, Wisconsin 53706, U.S.A. Volume 36. Pages 1729 - 1739.

ENGLE, S.J. AND WHALEN, S.H. **2003**. Autonomous Multi-Robot Foraging in a Simulated Environment. *A Final Technical Report - MAE.* Pages 298 - 2.

ERBAS, M.D., WINFIELD, A.F.T. AND BULL, L. **2013.** Embodied imitation-enhanced reinforcement learning in multi-agent systems. *In the journal for Adaptive Behaviour.* In Press.

EYIYUREKLI, M., BAI, L., LELKES, P.I. AND BREEN, D.E. **2013**. Chemotaxis-based Sorting of Self-Organizing Heterotypic Agents via a chemotaxis paradigm. *In* Science of Computer Programming. Volume 48. Issue 5. Pages 594 - 611.

FREDSLUND, J. AND MATARIĆ M.J. **2002**. Robots in Formation Using Local Information. *In the 7th International Conference on Intelligent Autonomous Systems (IAS-7).* Marina del Rey, California, USA. Pages 100 - 107.

FERNÁNDEZ, N., MALGONADO, C. AND GERSHENSON, C. **2013**. Information Measures of Complexity, Emergence, Self-organization, Homoeostasis, and Autopoiesis. *In guided Self-Organization: Inception. In press. Available on-line on http://arxiv.org/abs/1304.1842.*

FERNANDES, C., RAMOS, V. AND ROSA, A.C. **2005**. Self-Regulated Artificial Ant Colonies on Digital Image Habitats. *International Journal of Lateral Computing.* Volume 1. Number 2. Pages 1 - 8.

FISHER, D.A. AND LIPSON, H.F. **1999**. Emergent Algorithms - A New Method for Enhancing Survivability in Unbounded Systems. *In HICSS '99: Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences.* Volume 7. Pages 7043.

FRANCIS, A. **1990**. Advanced Level Statistics: An Integrated Course. *Old Station Drive, Leckhampton, Cheltenham.* GL53. Stanley Thomas Limited.

FRANKS, N.R., WILBY, A., SILVERMANT, B.W. AND TOFTS, C. **1992**. Self-organizing nest construction in ants : sophisticated building by blind bulldozing. *In the journal for Animal Behaviour.* Volume 2. Pages 357 - 375.

FULBRIGHT, R. AND STEPHENS, L.M. **2003**. An equation for the emergent capacity of artificial life. *Retrieved on 19th of April 2012 at http://jmvidal.cse.sc.edu/library/emergentcapacity.pdf.*

FUNES, P., ORME, AND B. BONABEAU, E. **2003**. Evolving emergent group behaviours for simple humans agents. *In proceedings of the 7th European Conference on Artificial Life.* Pages 76 - 89.

GAGE, D.W. **1993**. How to communicate with zillions of robots. *In Proceedings of SPIE Mobile Robots VIII.* Volume 2058. Pages 250-257.

GEER, P., McLAUGHLIN, H.W. AND UNSWORTH, K. **2003**. Cellular Lines: An Introduction. *In Discrete Mathematics and Theoretical Computer Science AB(DMCS), France.* Pages 167–178.

GHAIEBI, H. AND SOLIMANPUR, M. **2007**. An ant algorithm for optimization of hole-making operations. *In Computers and Industrial Engineering.* Elsevier Science Ltd. Volume 52. Pages 308 - 319.

GILLESPIE, D.T. **1976**. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics.* Volume 22. Pages 403 - 434.

GORE, R. AND REYNOLDS, P. **2008**. Applying causal inference to understand emergent behaviour. *In proceedings of the 2008 Winter Simulation Conference. Institute of Electrical and Electronic Engineers.* Pages 113 - 122.

GOTTLIEB, J., PUCHTA, M. AND SOLNON, C. **2003**. A Study of Greedy, Local Search, and Ant Colony Optimization Approaches for Car Sequencing Problems. *In Lecture Notes in Computer Science.* Volume 2611. Pages 246 - 257.

GREEN, D.G. **1994**. Emergent Behaviour in Biological Systems. *In Complexity International.* Volume 1. Pages 1 - 12.

GROCHOW, J. **2002**. Robust Methods of Synchronization in Amorphous Networks. *Available on-line on http://groups.csail.mit.edu/mac/projects/amorphous/6.978/final-papers/joshuag-final.pdf.*

GROSSMAN, R., SABALA, M., GU, Y., ANAND, A., HANDLEY, M., SULO, R. AND WILKINSON, L. **2009**. Discovering emergent behaviour from network packet data. Lessons from the angle project. *Available on-line on http://pubs.rgrossman.com/dl/proc-115.pdf.*

GROSZ, B. AND KRAUS, S. **1993**. Collaborative Plans for Group Activities. *In Distributed Artificial Intelligence*. Pages 367 - 373.

GULYAS, L., LAUFER, L. AND SZABO, R. **2006**. Measuring Stigmergy: The Case of Foraging Ants. *In Proceedings of Engineering self-Organizing Systems, ESOA 2006. In Lecture Notes in Artificial Intelligence.* Volume 4335. Pages 50 - 65.

HAASDIJK, E., EIBEN, A.E., AND WINFIELD, A.F.T. **2013**. Individual, Social, and Evolutionary Adaptation in Collective Systems. *In the Handbook of Collective Robotics - Fundamentals and Challenges.* Pages 411 - 469.

HAGLICH, P., ROUFF, C. AND PULLUM, L. **2010**. Detecting emergence in social networks. *In Proceedings - SocialCom 2010: 2nd IEEE International Conference on Social Computing, PASSAT 2010: 2nd IEEE International Conference on Privacy, Security, Risk and Trust.* Pages 693–696.

HAMANN, H., SCHMICKL, T. AND CRAILSHEIM, K. **2011**. Explaining Emergent Behavior in a Swarm System Based on an Inversion of the Fluctuation Theorem. *Advances in Artificial Life: ECAL 2011. In proceedings of the Eleventh European Conference on the Synthesis and Simulation of Living Systems.* Pages 302 - 309.

HARA, A., ICHIMURA, T., TAKAHAMA, T., ISOMICHI, Y. AND SHI GEMI, M. **2005**. Effect of Direct Communication in Ant Systems. *In Lecture Notes in Artificial Intelligence (LNAI).* Volume 3681. Pages 925 - 931.

HARDY, A. **2005**. Harmonic Interpolation for smooth curves and surfaces. *PhD thesis in applied mathematics.* University of Johannesburg.

HARRIS, D.M.J. **2007**. Direct motion of a parallel-linkage robot through the Jacobian. *In the 12th IFToMM World Congress, France.* Pages 1 - 5.

HEALEY, C.I.M. AND PRATT, S.C. **2008**. The effect of prior experience on nest site evaluation by the ant Temnothorax curvispinosus. *In Animal Behaviour.* Elsevier Science Ltd. Volume 76. Pages 893 - 899.

HINCHEY, M.G., ROUFF, C.A., RASH, J.L. AND TRUSZKOWSKI, W.F. **2005**. Requirements of an integrated formal method for intelligent swarms. *In FMICS '05: Proceedings of the 10th international workshop on Formal methods for industrial critical systems.* Pages 125–133.

HOVDA, P. **2008**. Quantifying weak emergence. *In Minds Mach. Volume 18(4).* Pages 461-473.

IMAI, H.T. **1966**. The chromosome observation techniques of ants and the chromosomes of formicinae and myrmicinae. *In Acta Hymenopterologica.* Volume 2. Issue 3. Pages 119 - 131.

IRONS, D. AND MONK, N. **2006**. Can Developmental Biology Inspire Amorphous Computing? *In the Second European Symposium on Nature-inspired Information Systems. Available on-line on http://nick-monk.staff.shef.ac.uk/reprints/Irons-monk-NISIS06.pdf.*

JACKSON, D.E., HOLCOMBE, M. AND RATNIEKS, F.L.W. **2004**. Trail geometry gives polarity to ants foraging networks. *In Nature (Nature Publishing Group, 2004).* Volume 432. Pages 907 - 909.

JADAAN, O.A., RAJAMANI, L. AND RAO, C.R. **2008**. Improved selection operator for GA. *In the Journal of Theoretical and Applied Information Technology - JATIT.* Pages 269 - 277.

JAYARAMAN, K. **1999**. A statistical manual for forestry research. *Forestry research support programme for Asia and the Pacific. Kerala Forest Research Institute. Food and agriculture organization of the united nations regional office for Asia and the Pacific.* Bangkok. Peechi, Kerala, India. *Available on-line on ftp://ftp.fao.org/docrep/003/X6831E/X6831E00.pdf.*

JEANNE, L.R. **1996**. Regulation of nest construction behaviour in Polybia occidentalis. *In the Association for the Study of Animal Behaviour.* Departments of Entomology and Zoology, University of Wisconsin at Madison. Volume 52. Pages 473 - 488.

JOHNSON, K, AND ROSSI, L.F. **2006**. A mathematical and experimental study of ant foraging trail dynamics. *In the Journal of Theoretical Biology.* Elsevier Science Ltd. Volume 241. Pages 360 – 369.

JOY, B. **2000**. Why the future doesn't need us. *In the Wired Magazine.* Volume 8. Issue 4. *Available on-line on http://www.wired.com /wired/archive/8.04/joy.html?pg=4&topic+set=.*

KAEWKAMNERDPONG, B., BHALLA, N. AND BENTLEY, P.J. **2007**. Programming Nanotechnology: Learning From Nature. *In the journal for Advances in Computers.* Volume 71. .

KE, L., ARCHETTI, C. AND FENG, Z. **2008**. Ants can solve the team orienteering problem. *In Computers and Industrial Engineering.* Elsevier Science Ltd. Volume 54. Pages 648 – 665.

KING, A., CALLAGHAN, V. AND CLARKE, G. **2005**. Problems Faced in Developing an Amorphous Computer as an Intelligent Skin for Pervasive Computing. *In proceedings of the IEE International Workshop on Intelligent Environments.* Pages 28 - 37..

KOZA, J.R. **1992**. Genetic Programming: On the programming of Computers by means of Natural selection. *A Bradford book in the MIT Press.*

KOICHI, K. AND MARI, N. **1996.** Generating Qualitative Equations about Macro Behaviours of foraging in Ant colony. *In proceedings of the German Conference on Bioinformatics, GCB'96 (Leipzig).* Pages 142 - 147.

KONDACS, A. **2003**. Biologically-Inspired Self-Assembly of 2Dimensional Shapes Using Global-to-Local Compilation. *In proceedings of the 18th International joint conference on Artificial intelligence. MIT Artificial Intelligence Laboratory.* Pages 633 - 638.

KRAUS, S. AND LEHMANN, D. **1995**. Designing and Building a Negotiating Automated Agent. *In Computational Intelligence.* Volume 11. Pages 132 - 171.

KRINK, T. AND VOLLRATH, F. **1998**. Emergent properties in the behaviour of a virtual spider robot. *In the Royal Society: 1998 Series.* Pages 2051 - 2055.

LIEN, J., RODRIGUEZ, S., TANG, X., MAFFEI, J., CORLETTE, D., MASCIOTRA, A. AND AMATO, N.M. **2005**. Composable Group Behaviour. *In the technical report of Tarasol Lab. Texas A&M University.*

LINE-LINE INTERSECTION. **5 January 2014 at 15:04**. *In Wikipedia: The Free Encyclopaedia Foundations Inc. Encyclopaedia on-line. Available on-line on http://en.wikipedia.org/wiki/Line-line intersection. Internet retrieved on 11 January 2014.*

LIU, Y. AND MOHAMED, Y. **2008**. Multi-Agent resource allocation (MARA) for modelling construction processes. *In the proceedings of the 2008 Winter Simulation Conference (IEEE Computer Society).* Pages 2361 - 2369.

LOHR, S.L. **2010**. Sampling: Design and Analysis. *In Brooks/ Cole Cengage Learning. ISBN - 13 : 978 - 0 - 495 - 10527 - 5, ISBN - 10 : 0 - 495 - 10527 - 9.* Arizona State University.

LUA, C.A., ALTENBERG, K. AND NYGARD, K.E. **2005**. ANTS with Firefly Communication. *In proceedings of IC-AI.* Pages 90 - 96.

MARTIN, P. **2006**. Beginning Research on the Quantification of Spatial Order. *In proceedings of the SIRC 2006- The 18th Annual Colloquium of the Spatial Information Research Centre.* University of Otago. New Zealand. Pages 109 - 125.

MASON, Z. **2002**. Programming with Stigmergy: Using Swarms for Construction. *In* Artificial *Life VIII, Standish, Abbass, Bedau (eds) (MIT Press).* Pages 371-374.

MEINHARDT, H. **1982**. Models of Biological Pattern Formation. *Academic press*, London.

MERKLE, D., MIDDENDORF, M. AND SCHEIDLER, A. **2006**. Modelling Ant Brood Tending Patterns with Cellular Automata. *In the Journal of Cellular Automata 2.2.* Pages 183 - 194.

MILES, J. **2008**. Table of Critical values for Pearson correlation. *Retrieved on 15/08/2008 from http://www.jeremymiles.co.uk/misc/tables/pearson.html*".

MINATI, G. **2002**. Emergence and Ergodicity: a line of research. *In Emergence in complex cognitive, social and biological Systems.* New York. Pages 85 - 102.

MONTES DE OCA, M.A., GARRIDO, L. AND AGUIRRE, J.L. **2005**. Effects of Inter-agent Communication in Ant-Based Clustering Algorithms: A case Study on Communication Policies in Swarm Systems. *In MICAI 2005. Advances in Artificial Intelligence. Lecture Notes in Computer Science.* Volume 3789. Pages 254 - 263.

MONTES DE OCA, M.A., GARRIDO, L. AND AGUIRRE, J.L. **2004**. On the effects of direct communication between agents in ant-based clustering algorithms. *In proceedings of the fifth Iberoamerican Workshop on Multi-Agent Systems (Iberagents 2004).*

MONTGOMERY, J., RANDALL, M. AND HENDTLASS, T. **2007**. Solution bias in ant colony optimisation: Lessons for selecting pheromone models. *In Computers and Operations Research.* Elsevier Science Ltd. Volume 35. Pages 2728 – 2749.

MULLEN, R.J., MONEKOSSO, D., BARMAN, S. AND REMAGNINO, P. **2009**. A review of ant algorithms. *In Expert Systems with Applications.* Volume 36. Issue 6. Pages 9608 -9617.

NAEEM, W., SUTTON, R. AND CHUDLEY, J. **2007**. Chemical plume tracing and odour source localisation by autonomous vehicles. *In Journal of Navigation.* Volume 60. Pages 173 - 190.

NAGPAL, R. **1999**. Organizing a Global Coordinate System from Local Information on an Amorphous Computer. *In AI Memo Number 1666. Massachusetts Institute of Technology.*

NAGPAL, R. **2001**. Programmable Self-Assembly: Constructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics. *A PhD thesis submitted to the Department of Electrical Engineering and Computer Science. MIT.*

NAGPAL, R. AND COORE, D. **1998**. An Algorithm for Group Formation and Maximal Independent Set in an Amorphous Computer. *In the Artificial Intelligence Memo Number 1626.*

NAGPAL, R., KONDACS, A. AND CHANG, C. **2002**. Programming Methodology for Biologically-Inspired Self-Assembling Systems. *In proceedings of the American Association for Artificial Intelligence.* Technical Report SS - 03 - 02.

NAGPAL, R., SHROBE, H. AND BACHRACH, J. **2003**. Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network. *In proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03). Palo Alto.* Also *In American Association for Artificial Intelligence.* Pages 333 - 348.

NAGPAL, R., ZAMBONELLI, F., SIRER, E.G., CHAOUCHI, H. AND SMIRNOV, M. **2006**. Interdisciplinary Research: Roles for Self-Organization- Self Organizing Shape and Pattern from Cells to Robots. *In the journal for Intelligent systems.* Volume 21. Pages 50 - 58.

NAGPAL, R. **2006**. Self-Organizing Shape and Pattern: From Cells to Robots. *Invited paper in IEEE Intelligent Systems.* Pages 50 - 53.

NAKAMICHI, Y. AND ARITA, T. **2004**. An evolutionary simulation of the origin of pheromone communication. *In proceedings of the 5th International Conference on Simulated Evolution and Learning.* Pages 1 - 6.

NAKAMICHI, Y. AND ARITA, T. **2005**. Effectiveness of Emerged Pheromone Communication in an Ant Foraging Model. *In the Tenth International Symposium on Artificial Life and Robotics,* B-con Plaza, Beppu, Oita, Japan. Pages 324 - 327.

NASIPURI, A. AND LI, K. **2002**. A Directionality based Location Discovery Scheme for Wireless Sensor Networks. *In proceedings of WSNA' 02.* Georgia, USA. Pages 105 - 111.

NEGULESCU, S.C. AND BARBAT, B.E. **2004**. Enhancing the effectiveness of Simple Multi-Agent Systems through Stigmergic Coordination. *In the Fourth International ICSC Symposium on Engineering of intelligent systems (EIS 2004).* University of Madeira. Pages 1 - 7.

NEGULESCU, S.C., ZAMFIRESCU, C. AND BARBAT, B. **2006**. User-Driven Heuristics for non-deterministic problems. *In Studies in Informatics and Control: Special issue dedicated to the 2nd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence.* Pages 289 - 296.

TSUTSUI, N.D. AND SUAREZ, A.V. **2003**. The colony structure and population biology of Invasive Ants. *In conservation Biology.* Volume 17. Issue 1. Pages 48–58.

NGO, V.T., NGUYEN, A.D. AND HA, Q.P. **2005**. Integration of Planning and Control in Robotic Formations. *In proceedings of the 2005 Australian Conference on Robotics and Automation*, Sydney, Australia.

NOBLE, D. AND LETSKY, M. **2002**. Cognitive-Based Metrics to Evaluate Collaboration Effectiveness. *Proceedings of the RTO SAS symposium on Analysis of Military Effectiveness of Future C2 Concepts and Systems.* Netherlands. Pages B6-1 - B6-28.

NORTH CAROLINA BOARD OF SCIENCE AND TECHNOLOGY SECRETARIAT. **2006**. A Roadmap for Nanotechnology in North Carolina's 21st Century Economy". Nanotechnology and North Carolina's 21st Century Economy. *A* division *of the North Carolina Department of Commerce. Available on-line on http://www.nccommerce.com/Portals/6/Documents/NCNanotechTaskForce ReportFinal.pdf.*

NOUYAN, S. AND DORIGO, M. **2007**. Chain Based Path Formation in Swarms of Robots. *In Scientific Research Directorate of the French Community of Belgium and the SWARM-BOTS Project, Future and Emerging Technologies programme (IST-FET) of the European Commission.* Pages 120 - 131.

NOWÉ, A., VERBEECK, K. AND VRANCX, P. **2004**. Multi-type Ant Colony: The Edge Disjoint Paths Problem. *In ANTS 2004: Lecture Notes in Computer Science.* Pages 202 - 213.

PANAIT, L. AND LUKE, S. **2004a**. A Pheromone-Based Utility Model for Collaborative Foraging. *In AAMAS-2004 - proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems.*

PANAIT, L. AND LUKE, S. **2004b**. Ant Foraging Revisited. *In proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9).*

PANAIT, L. AND LUKE, S. **2004c**. Learning Ant Foraging Behaviours.

In proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9).

PARRISH, J.K., VISCIDO, S.V., AND GRUNBAUM, D. **2002**. Self-Organized Fish Schools: An Examination of Emergent Properties. *In the Biol. Bull.* Volume 202. Pages 296 - 305.

PARUNAK, H.V.D. AND VANDERBOK, R.S. **1997**. Managing Emergent Behaviour in Distributed Control Systems. *In ISA Tech ' 97.* Anaheim. CA. Pages 1 - 8.

PARUNAK, H.V.D. **2005**. A Survey of Environments and Mechanisms for Human-Human Stigmergy. *In proceedings of the 2nd international conference on environments for Multi Agent Systems.* Pages 163 - 186.

PAULSON, L.D. **2008**. News Briefs. *In the Computer Journal.* Volume 41. Pages 16 - 18.

PAYTON, D., DAILY, M., ESTOWSKI, R., HOWARD, M. AND LEE, C. **2001**. Pheromone Robotics. *In Autonomous Robots.* Volume 11. Pages 319 – 324.

PELECHANO, N., ALLBECK, J.M. AND BADLER, N.I. **2007**. Controlling individual agents in high-density crowd simulation. *In SCA '07: proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation.* Pages 99 - 108.

POON, J. AND MAHER, M.L. **1996**. Emergent behaviour in co-evolutionary design. *In the journal for Artificial Intelligence in Design. 1996 series.* Pages 703 - 722.

POLACK, F. AND STEPNEY, S. **2005**. Emergent Properties Do Not Refine. *A preliminary version in TUNA feasibility study, EPSRC.*

POLACK, F., STEPNEY, S., TURNER, H., WELCH, P. AND BARNES, F. **2005**. An Architecture for modelling emergence in CA-Like Systems. *In Lecture Notes in Artificial Intelligence : Advances in Artificial Intelligence.* Volume 3630. Pages 433 - 442.

PORTA, G.M.A., OSCAR, M., OSCAR, C., ROBERTO, S. AND PATRICIA, M. **2009**. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *In the journal for Applied Soft Computing.* Elsevier Science Ltd. Volume 9. Pages 1102 - 1110.

POWELL, S. AND FRANKS, N.R. **2007**. How a few help all: living pothole plugs speed prey delivery in the army ant Eciton burchellii. *In the Association for the Study of Animal Behaviour.* Elsevier Science Ltd. Volume 73. Pages 1067 - 1076.

PRIESTERJAHN, S., GOEBELS, A. AND WEIMER, A. **2005**. Stigmergic Communication for Cooperative Agent Routing in Virtual Environments. *In international conference on Artificial Intelligence and the Simulation of Behaviour* (AISB'05).

RAJBHUPINDER, K., RANJIT, S.D., HARWINDER, S.S. AND AMARPREET, S.G. **2010**. Load Balancing of Ant Based Algorithm in MANET. *In the International Journal of Computer Science and Technology* (IJCST). Volume 1. Issue 2. Pages 173-178.

RAVARY, F., LECOUTEY, E., KAMINSKI, G., CHARLINE, N. AND JAISSON, P. **2007**. Individual Experience Alone Can Generate Lasting Division of Labor in Ants. *In Current Biology.* Elsevier Science Ltd. Volume 17. Pages 1308–1312

REGAN, B.C., ALONI, S., JENSEN, K. AND ZETTI, A. **2005**. Surface-tension-driven nano electromechanical relaxation oscillator. *In Applied Physics Letters.* Volume 86. Pages 123119-1 - 123119-3.

REIF, J.H. **2002**. Molecular Assembly and Computation: From Theory to Experimental Demonstrations. *In proceedings of the Twenty Ninth International Colloquium on Automata, Languages, and Programming (ICALP).* Malaga, Spain. Pages 1 - 21.

REYNOLDS, C.W. **1987**. Flocks, Herds, and Schools: A Distributed Behavioural Model. *In the proceedings of SIGGRAPH '87. Computer Graphics volume 21(4). In Maureen C. Stone's edition.* July 1987. Pages 25 - 34.

REYNOLDS, C.W. **1999**. Steering Behaviours for Autonomous Characters. *In Game Developers Conference (1999). Available on-line on http://www.red3d.com/cwr/steer/gdc99.*

REZAEE, A. **2008**. Extracting edges of images with ant colony. *In the Journal of electrical engineering.* Volume 59. Pages 57 - 59.

RODRIGUEZ, S., SALAZAR, R., McMAHON, T. AND AMATO, N.M. **2007**. Roadmap-Based Group Behaviour: Generation and Evaluation. *A technical report submitted to the Parasol Laboratory, Department of Computer Science, Texas A&M University.*

ROTHEMUND, P.W.K. **2006**. Folding DNA to create nano-scale shapes and patterns. *Computation and Neural Systems and Computer Science.* Volume 440. Pages 297 - 302.

ROUFF, C., VANDERBILT, A., HINCHEY, M., TRUSZKOWSKI, W. AND RASH, J. **2004**. Verification of emergent behaviours in swarm-based systems. *In Engineering of Computer-Based Systems, IEEE International Conference.* Page 443 - 448.

ROUMELIOTIS, S.I., PIRJANIAN, P. AND MATARIC, M.J. **2000**. Ant-Inspired Navigation in Unknown Environments. *In proceedings of* *the 4th International Conference on Autonomous agents.* Pages 25 - 26.

SAENZ, A. **2011**. Google's AI Challenge: Who Can Build The Smartest Ant Colony. *Published* on-line *on the 26th of October 2011. Retrieved on the 15/11/2011 from http://singularityhub.com/2011/10/26/ googles-ai-challenge-who-can-build-the-smartest-ant-colony/*

SANDERS, J.W. AND SMITH, G. **2009**. Refining emergent properities. A technical report at the United Nations University: International Institute for Software Technology. Report No. 419. Macao. *In Electronic Notes in Theoretical Computer Science.* Volume 259. Pages 207 - 223.

SARFATI, J. **2001**. Ants find their way by advanced mathematics. *In Perspectives: TJ.* Volume 15. Pages 11 - 12.

SATO, K. AND MATSUOKA, Y. **2009**. Emergent Design and Optimum Design in the Design Process. *International Association of Societies of Design Research (2009).*

SCHAEFER, L., WOLFE, P., FOWLER, J. AND LINDQUIST, T. **2002**. Simulation meta-architecture for analyzing the emergent behavior of agents. *Draft Manuscript: http://www.csu.edu.au/ci/draft/schaef01.*

SCHOONDERWOERD, R., HOLLAND, O., BRUTEN, J. AND ROTHKRANTZ, L. **1996**. Ant-based load balancing in telecommunications networks. *In the journal for adaptive behaviour.* Volume 5. Issue 2. Pages 169 - 207.

SCHNEIDER, D.S. **2007**. Information Theory Primer with an Appendix on Logarithms. *Part of a PhD thesis. Available on-line on http://*

schneider.ncifcrf.gov/papers/primer/primer.pdf.

SCHNEIDER, S., CAVALCANTI, A., TREHARNE, H. AND WOODCOCK, J. **2006**. A Layered Behavioural Model of Platelets. *In the proceedings of the 11th IEEE International Conference on Engineering of Complex Systems* - ICECCS 2006. IEEE 2006. Pages 98 - 106.

SEEVINCK, J. AND EDMONDS, E. **2008**. Emergence and the art system 'plus minus now'. *In Design Studies.* Elsevier Science Ltd. Volume 25. Pages 541 - 555.

SENJEN, R. **2009**. Nanotechnolodies in the 21th Century: Challenges and opportunities to green nanotechnologies. *In the 2009 Nanotechnology* Brochure. *Available on-line on http://www.eeb.org/publication/ 2009/2009-NanoBronchureNo1-Web.pdf.*

SHALIZI, C. **2001**. Causal architecture, complexity and self-organization in time series and cellular automata. *A PhD thesis submitted to the University of Wisconsin - Madison.*

SHEEHAM, M.J. AND TIBBETTS, E.A. **2008**. Robust long-term social memories in a paper wasp. *In Current Biology, Ecology and Evolutionary Biology.* Elsevier Science Ltd. Volume 18. Issue 18. Pages R851 - R852.

SHELL, D.A. **2003**. An Annotated Bibliography of papers that make use of the word "Stigmergy". *University of Southern California*, USA. *Available at http://www-robotics.usc.edu/~dshell/stigmergy.php*

SINGH B.N., SINGH, S. AND BANERJEE, P. **2012**. A comparison between polythene chromosomes of two sibling species of Drosophila. *Research notes in the Dros. Inf. Serv. Volume 95. Genetics*

Laboratory, Department of Zoology, Banaras Hindu University, India. Volume 95. Pages 50 - 54.

SOCHA, K. **2008**. Ant Colony Optimization for Continuous and Mixed-Variable Domains. *PhD thesis, IRIDIA, CoDE, Universite Libre de Bruxelles.*

SOLÉ, R.V., BONABEAU, E., DELGADO, J., FERNANDES, P. AND MARIN, J. **2000**. Pattern Formation and Optimization in Army Ant Raids. *In the journal for Artificial Life.* Volume 6. Number 3. Pages 219 - 226.

SOLIMANPUR, M., VRATB, P. AND SHANKARC, R. **2005**. An ant algorithm for the single row layout problem in flexible manufacturing systems. *In Computers and Operations Research.* Elsevier Science Ltd. Volume 32. Pages 583–598.

SOLOVEICHIK, D. **2008**. Molecules Computing: Self-Assembled Nanostructures, Molecular Automata, and Chemical Reaction Networks. *PhD Dissertation.* California Institute of Technology.

SOLNON, C. AND FENET, S. **2006**. A study of ACO capabilities for solving the maximum clique problem. *In Springer Heuristics.* Volume 12. Pages 155 - 180.

SPEARS, W.M., SPEARS, D.F., ZARZHITSKY, D. AND HEIL, R. **2004a**. Physicomimetics for mobile robot formations. *In proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS-04).* Volume 3. Pages 1528 - 1529.

SPEARS, W.M., SPEARS, D.F., HAMANN, J.C. AND HEIL, R. **2004b**. Distributed, Physics-Based Control of Swarms of Vehicles. *In Autonomous Robots.* Volume 17. Pages 137 - 162.

SPEARS, W.M., SPEARS, D.F. AND ZARZHITSKY, D. **2005**. Physicomimetics positioning methodology for distributed autonomous systems. *In proceedings of the Government Microcircuit Applications and Critical Technology Conference, GOMACTech-05 Intelligent Technologies.* Las Vegas.

STEFANO A.D. AND SANTORO, C. **2001**. Coordinating Mobile Agents by means of Communicators. *In proceedings of the Workshop on objects and agents (WOA2001)*, Modena, Italy. Pages 48 - 55.

STEPNEY, S., POLACK, F.A.C. AND TURNER H.R. **2007**. Engineering Emergence. *In proceedings of the 11th IEEE International conference on Engineering of Complex Systems.*

SUDD, J.H. **1960**. THE FORAGING METHOD OF PHARAOH'S ANT, Monomorium pharaonis. *In Animal Behaviour.* Volume 8. Issues 1 - 2. Pages 67 - 75.

SUMPTER, D.J.T., BLANCHARD, G.B. AND BROOMHEAD, D.S. **2001**. Ants and Agents: a Process Algebra Approach to Modelling Ant Colony Behaviour. *Bulletin of Mathematical Biology.* Volume 63. Pages 951–980.

SUNDAY, D. **2007**. Distance between lines and segments with their closest point of approach. *In SoftSurfer. Available on-line on http://geom algorithms.com/*

SUSSMAN, G.J. **1999**. Robust Design through Diversity. *In Robust diversity.* Massachusetts Institute of Technology. *Available on-line on http://groups.csall.mit.edu/mac/projects/amorphous/workshop-sept-99/robust-diversity.pdf.*

SVENSON, P. AND SIDENBLADH, H. **2003**. Determining possible avenues of approach using ANTS. *Appears in the 6th International*

Conference on Information Fusion. Pages 1110 - 1117.

TAO, W., JIN, H. AND LIU, L. **2007**. Object segmentation using ant colony optimization algorithm and fuzzy entropy. *In pattern recognition letters.* Volume 28. Pages 788 - 796.

THERAULAZ, G., BONABEAU, E. AND DENEUBOURG, J. **1998**. The origin of nest complexity in social insects. *In Complexity.* Volume 3(6). Pages 15-25.

TOFTS, C. **1991**. Describing social insect behaviour using process algebra. *Transactions on Social Computing Simulation.* Pages 227-283.

TOTH-FEJEL, T. **2004**. Modelling Kinematic Cellular Automata. *In NASA Institute for Advanced Concepts. Available on-line on http:// www.niac.usra.edu/files/studies/final_ report/883Toth - Fejel.pdf.*

TREDER, M. **2004**. Bridges to Safety, and Bridges to Progress. *In the Centre for Responsible Nanotechnology.* Page 6. www.crnano.org/ Bridges.htm.

TRIANNI, V. AND DORIGO, M. **2005**. Self-Organisation and Communication in Groups of Simulated and Physical Robots. *A technical report submitted to the Université Libre de Bruxelles. In Biological Cybernetics.* Volume 95. Issue 3. Pages 213–231.

TROCHIM, W.M.K. **2006**. Research Methods Knowledge Base: Descriptive statistics: Correlation. *Retrieved on 15/08/2008 from http: //www.socialresearchmethods.net/kb/statcorr.php".*

TROFIMOVA, I.N., POTAPOV, A.B. AND SULIS, W.H. **1998**. Collective effects on individual behaviour: Three questions in the search for universality. *In the International Journal of Chaos Theory and Application.* Volume 3. Pages 1 - 2.

VALCKENAERS, P., KOLLINGBAUM, M., VAN BRUSSEL, H., BOCHMAN, O. AND ZAMFIRESCU, C. **2001**. The design of multi-agent coordination and control systems using stigmergy. *In proceedings of the IWES 2001 Conference, Blen.* Slovenia.

VALVERDE, S., THERAULAZ, G., GAUTRAIS, J., FOURCASSIE, V. AND SOLE, R.V. **2006**. Emergent behaviour in agent networks: Self-Organization in Wasp and Open Source Communities. *IEEE Intelligent Systems.* Volume 21 (2). Pages 36 - 60.

VAN DEN BERGH, F. AND ENGELBRECHT, A.P. **2001**. Effects of Swarm Size on Cooperative Particle Swarm Optimisers. *In proceedings of GECCO 2001. San Francisco.*

VAN JAARSVELD, A. AND BOZZOLI, B. **2009**. Trajectories for NRF research support. *National Research Foundation.* South Africa.

VIANA, F.A.C., KOTINDA, G.I., RADE, D.A. AND STEFFEN, V (jr). **2007**. Tuning dynamic vibration absorbers by using ant colony optimization**.** *In the Journal for Computers and Structures.* Volume 86. Issue 13 - 14.

WANG, S. AND ZHU, H. **2007**. An Experimental Study of the Emergent Behaviours of Self-Organized Agent Communities. *In Evolutionary Computation 2007. IEEE Congress on.* Pages 3239– 3246.

WEHNER, R., BOYER, M., LOERTSCHER, F., SOMMER, S. AND MENZI, U. **2006**. Ant Navigation: One-Way Routes Rather Than Maps. *In Current Biology.* Elsevier Science Ltd. Volume 16. Pages 75 - 79.

WEI-MIN, S., WILL, P. AND ARAM, G. **200**4. Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms. *In the journal of autonomous robots.* Volume 17. Pages 93 - 105.

WERFEL, J. **2002**. Autonomous multi-agent construction. *In Amorphous projects. Available on-line on http://groups.csall.mit.edu/mac/projects/amorphous.6.978/final_papers/werfel-final.pdf.*

WERFEL, J., NAGPAL, N. AND SEUNG, H.S. **2006**. Anthills Built to Order: Automating Construction with Artificial Swarms. *PhD Dissertation.* Massachusetts Institute of Technology.

WERFEL, J., BAR-YAM, Y. AND NAGPAL, R. **2005**. Building Patterned Structures with Robot Swarms. *In the International Joint Conference on Artificial Intelligence (IJCAI '05).* Pages 1495 - 1502.

WEYNS, D., STEEGMANS, E. AND HOLVOET, T. **2004**. Protocol-Based Communication for Situated Multi-Agent Systems. *In AAMAS'04 Proceedings of the third International Joint Conference on Autonomous Agents and Multiagent Systems.* Volume 1. Pages 118 - 125.

WINFIELD, A.F.T., KERNBACH, S., AND SCHMICKI, T. **2013**. Collective Foraging: Cleaning, Energy HArvesting and Trophallaxis. *In the Handbook of Collective Robotics - Fundamentals and Challenges.* Pages 351 - 409.

WHITE, T. **1997**. Swarm intelligence and problem solving in telecommunications. *Canadian Artificial Intelligence Magazine.* Volume 41. Pages 14-16.

WU, H., WANG, C. AND TZENG, N. **2005**. Novel Self-Configurable Positioning Technique for Multihop Wireless Networks. *In IEEE/*

ACM transactions on networking. Volume 13. Issue 3. Pages 609 - 621.

XI, J., SCHMIDT, J.B. AND MONTEMAGNO, C.D. **2005**. Self-assembled micro-devices driven by muscles. *In the journal of Nature Materials.* Volume 4. Pages 180 - 184.

XU, B., CHEN, Q. AND WANG, Z. **2008**. Ants for track initiation of bearings-only tracking. *In the journal for Simulation Modelling Practice and Theory.* Elsevier Science Ltd. Volume 16. Pages 626 - 638.

YAMINS, D. **2007**. A theory of local-to-global algorithms for one-dimensional spatial multi-agent systems. *PhD dissertation.* Harvard University.

YANG, J. AND ZHUANG, Y. **2010**. An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem. *In Applied Software Computing.* Elsevier Science Ltd. Volume 10. Issue 2. Pages 653 - 660.

ZARNANI, A. AND RAHGOZAR, M. **2006**. Mining spatial trends by a colony of cooperative ant agents. *In proceedings of the SIAM conference on Data mining workshop on Spatial Data mining. Available on-line on http://www.cs.dartmouth.edu/~cbk/sdm06/zarnani-rahgozar-sdm06.pdf.*

Index