A networked multi-agent combat model : emergence explained

Author:
Yang, Ang

Publication Date:
2007

DOI:
https://doi.org/10.26190/unsworks/18098

A Networked Multi-Agent Combat Model: Emergence Explained

Ang Yang
M.InfoSci. (Massey University, New Zealand)
M.EnvGeo. (Nanjing University, China)
B.Sci. (Ocean University of China, China)

A thesis submitted in fulfillment of the requirements
for the degree of **Doctor of Philosophy** at the
School of Information Technology & Electrical Engineering
University of New South Wales
Australian Defence Force Academy

Abstract

Simulation has been used to model combat for a long time. Recently, it has been accepted that combat is a complex adaptive system (CAS). Multi-agent systems (MAS) are also considered as a powerful modelling and development environment to simulate combat. Agent-based distillations (ABD) - proposed by the US Marine Corp - are a type of MAS used mainly by the military for exploring large scenario spaces. ABDs that facilitated the analysis and understanding of combat include: ISAAC, EINSTein, MANA, CROCADILE and BactoWars. With new concepts such as networked forces, previous ABDs can implicitly simulate a networked force. However, the architectures of these systems limit the potential advantages gained from the use of networks.

In this thesis, a novel network centric multi-agent architecture (NCMAA) is proposed, based purely on network theory and CAS. In NCMAA, each relationship and interaction is modelled as a network, with the entities or agents as the nodes. NCMAA offers the following advantages:

1. An explicit model of interactions/relationships: it facilitates the analysis of the role of interactions/relationships in simulations;

2. A mechanism to capture the interaction or influence between networks;

3. A formal real-time reasoning framework at the network level in ABDs: it interprets the emergent behaviours online.

For a long time, it has been believed that it is hard in CAS to reason about emerging phenomena. In this thesis, I show that despite being almost impossible to reason about the behaviour of the system by looking at the components alone because of high nonlinearity, it is possible to reason about emerging phenomena by looking at the network level. This is undertaken through analysing network dynamics, where I provide an English-like reasoning log to explain the simulation.

Two implementations of a new land-combat system called the Warfare Intelligent System for Dynamic Optimization of Missions (WISDOM) are presented. WISDOM-I is built based on the same principles as those in existing ABDs while WISDOM-II is built based on NCMAA. The unique features of WISDOM-II include:

1. A real-time network analysis toolbox: it captures patterns while interaction is evolving during the simulation;

2. Flexible C3 (command, control and communication) models;

3. Integration of tactics with strategies: the tactical decisions are guided by the strategic planning;
4. A model of recovery: it allows users to study the role of recovery capability and resources;
5. Real-time visualization of all possible information: it allows users to intervene during the simulation to steer it differently in human-in-the-loop simulations.

A comparison between the fitness landscapes of WISDOM-I and II reveals similarities and differences, which emphasise the importance and role of the networked architecture and the addition of strategic planning.

Lastly but not least, WISDOM-II is used in an experiment with two setups, with and without strategic planning in different urban terrains. When the strategic planning was removed, conclusions were similar to traditional ABDs but were very different when the system ran with strategic planning. As such, I show that results obtained from traditional ABDs - where rational group planning is not considered - can be misleading.

Finally, the thesis tests and demonstrates the role of communication in urban terrains. As future warfighting concepts tend to focus on asymmetric warfare in urban environments, it was vital to test the role of networked forces in these environments. I demonstrate that there is a phase transition in a number of situations where highly dense urban terrains may lead to similar outcomes as open terrains, while medium to light dense urban terrains have different dynamics.

Acknowledgement

First and foremost I would like to thank my principal supervisor Associate Professor Hussein A. Abbass for his continuous encouragement, patience and guidance throughout the course of my research. He provided extensive help far beyond what is expected of a supervisor.

I am also grateful to my co-supervisors Dr. Ruhul Sarker from UNSW@ADFA and Dr. Neville J. Curtis from DSTO, Australia for their advice, support and enthusiasm.

I wish to thank all academic and administrative staff in computer science, especially Dr. Michael Barlow, Dr. Ed Lewis, Dr. Weiping Zhu and Mr. Mike Ford for their valuable suggestions.

I especially thank Jennifer Badham, who provided both technical and linguistic help. My thanks also go to all of my fellow at the Artificial Life and Adaptive Robotics Laboratory (ALAR) for their assistance and company, particularly Lam Bui, Mehrdad Khodai-Joopari, Minh Ha Nguyen, Helen Dam and Vinh Bui.

I want to thank Dr. Yin Shan for his useful discussion and constructive advice throughout my research even after he left UNSW@ADFA.

The people to whom I am most indebted, are my parents, Wensheng Yang and Guohua Xia, and my wife, Xuemei Sun. I can never overstate their importance to me, for this achievement would never have been possible without their love, support and sacrifice. My wife has accompanied me for more than 10 years. She has continuously been supporting me with her endless love, encouragement and patience.

Finally, I would like to thank my gorgeous baby son, Terry Yang. Your arrival brought a lot of happiness to me at the end of this journey.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Signed: ....

Date: ....

Publications

Peer-reviewed publications arising from this thesis are listed below:

Submitted:

- [Yang et al. (2006a)] Yang, A., N. J. Curtis, H. A. Abbass, and R. Sarker (2006). NCMAA: a Network centric multi-agent architecture for modelling complex adaptive systems. (Submitted).

Journal Publications:

- [Yang et al. (2006b)] Yang, A., H. A. Abbass, and R. Sarker (2006). Characterizing warfare in red teaming. IEEE Transactions on Systems, Man, Cybernetics, Part B: Cybernetics, 36(2), 268-285.

Springer – Lecture Notes Series:

- [Yang et al. (2006c)] Yang, A., H. A. Abbass, and R. Sarker (2006). Land combat scenario planning: A multiobjective approach. In The Sixth International Conference on Simulated Evolution And Learning (SEAL'06), LNCS, Hefei, China. (Accepted)
- [Yang et al. (2005a)] Yang, A., H. A. Abbass, and R. Sarker (2005). WISDOM-II: A network centric model for warfare. In Ninth International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES 2005), LNCS 3683, Melbourne, Australia.
- [Yang et al. (2004a)] Yang, A., H. A. Abbass, and R. Sarker (2004). Landscape dynamics in multi-agent simulation combat systems. In Proceedings of 17th Joint Australian Conference on Artificial Intelligence, LNAI 3339, Cairns, Australia. Springer-Verlag.

Other Refereed Publications:

- [Yang et al. (2005b)] Yang, A., H. A. Abbass, and R. Sarker (2005). Risk assessment of capability requirements using WISDOM-II. In A. Bender (Ed.), The International Society for Optical Engineering (SPIE), International Symposium of Microelectronics, MEMS, and Nanotechnology: Complex Systems Conference, Proceeding of SPIE Vol 6039, Brisbane, Australia.

- [Yang et al. (2005c)] Yang, A., H. A. Abbass, M. Barlow, R. Sarker, and N. J. Curtis (2005). Evolving capability requirements in WISDOM-II. In H. A. Abbass, T. Bossamier, and J. Wiles (Eds.), Advances in Artificial Life, Proceeding of The Second Australian Conference on Artificial Life (ACAL05), Sydney, Australia, pp. 335-348. World Scientific Publisher.
- [Yang et al. (2006d)] Yang, A., N. J. Curtis, H. J. Abbass, R. Sarker, and M. Barlow (2006). Wisdom-II: A network centric model for warfare. In P. Perez and D. Batten (Eds.), Complex Science for a Complex World. ANU E Press, Australia.
- [Yang et al. (2005d)] Yang, A., H. A. Abbass, and R. Sarker (2005). How hard is it to red team? In H. A. Abbass and D. Essam (Eds.), Applications of Information Systems to Homeland Security and Defense. Idea Group Inc.

Un-refereed Publications:

- [Yang et al. (2005e)] Yang, A., H. A. Abbass, and R. Sarker (2005). Evolving agents for network centric warfare. In GECCO '05: Proceedings of the 2005 workshops on Genetic and evolutionary computation: Second Workshop on Military and Security Applications of Evolutionary Computation, Washington D.C., NY, USA, pp. 193-195. ACM Press.
- [Yang et al. (2004b)] Yang, A. (2004). Understanding network centric warfare. ASOR BULLETIN 23(4), 2-6.

Contents

List of Figures

List of Tables

List of Acronyms

ABD	Agent Based Distillation
ABS	Agent Based Simulation
AGG	Aggressive scenario
AHP	Analytic Hierarchy Process
BAL	Balanced scenario
BDI	Belief-Desire-Intention agent architecture
C2	Command and Control
C3	Command, Control and Communication
CAS	Complex Adaptive System
CASTFOREM	Combined Arms and Support Task FORce Evaluation Model
CGF	Computer Generated Forces
COP	Common Operating Picture
COW	Coward scenario
CROCADILE	Conceptual Research Oriented Combat Agent Distillation Implemented in the Littoral Environment
DAG	Directed Acyclic Graph
DEF	Defensive scenario
DIAS	Dynamic Information Architecture System
EC	Evolutionary Computation
EINSTein	Enhanced ISAAC Neural Simulation Toolkit
ELAN	Extended LANchester model
EMO	Evolutionary Multi-objective Optimization
ES	Evolutionary Strategy
FDC	Fitness Distance Correlation
GA	Genetic Algorithm
GOL	Goal oriented scenario
HERM	Higher-Order Entity Relationship Modelling
ISAAC	Irreducible Semi-Autonomous Adaptive Combat
LE	Lanchester Equation
LER	Loss Exchange Ratio
MANA	Map Aware Non-uniform Automata
MAS	Multi-Agent System
MAVF	Multi-Attribute Value Function
MHPCC	Maui High Performance Computing Center
ModSAF	Modular Semi-Automated Forces
MOUT	Military Operations in the Urban Terrain
NCMAA	Network Centric Multi-Agent Architecture
NCW	Network Centric Warfare
NLER	Normalized LER

OODA	Observe-Orient-Decide-Act
OR	Operation Research
PRS	Procedural Reasoning System
RA	Robust Analysis
RCA	Root Cause Analysis
RePast	Recursive Porous Agent Simulation Toolkit
VAG	Very aggressive scenario
SA	Situation Awareness
SAF	Semi-Automated Forces
SEM	Structural Equation Model
SSM	Soft Systems Methodology
WISDOM	Warfare Intelligent System for Dynamic Optimization of Missions
WISDOM-I	the version I of WISDOM
WISDOM-II	the version II of WISDOM

Chapter 1

Introduction

1.1 Background

Decision makers are usually faced with a large number of threats and ultimately try to make decisions in such a way that vulnerabilities are mitigated before the implementation and execution of these decisions. Vulnerabilities are holes in a security system, tactic, operation or plan. Red teaming (Mateski 2004; Sandoz 2001; DOD 2003) is a connotation for playing the devil; trying to penetrate the mind of the enemy or competitor to imitate their behaviours; understanding risk in the eyes of the opponent and mitigating vulnerabilities before it is too late.

Defence organizations (DOD 2003) have identified red teaming as a valuable activity to mitigate risk and challenge plans and tactics. It is common to say "we need to avoid risks". Red teaming is a risk assessment activity which answers questions such as: what are these risks and what are their natures? how do these risks come to exist in the first instance? who can create them to be able to understand and avoid them? how can we defend ourselves against these risks? what are their consequences? etc.

Exploiting vulnerabilities to mitigate risk can be done by human-based red teaming, where a force is divided into two teams; one simulating the enemy (red team) while the other simulating friends (blue team). Defence uses this approach in its operation exercises. However, this approach is extremely expensive (Yang et al. 2004) and does not enable defence analysts to explore all aspects of a situation. Computer simulations of multi-agent systems (MAS), among others, are used for software-based red teaming. These simulations explore abstract higher level scenarios of different vulnerabilities in a plan or operation. Once the weaknesses in the system are identified and a risk analysis is conducted, human-based red teaming can be used in a more focused way to increase the fidelity of the analysis.

Traditionally, defence analysts adopted what are known as Lanchester Equations to model and theorize about combat attrition (Ilachinski 1997; Ilachinski 2000). Lanchester Equations were introduced by F. W. Lanchester in 1916 (Lanchester 1916) as a set of linear dynamic equations that treat attrition as a continuous function over time. The set of equations is intuitive and easy to apply. However, such models, which are based on mathematical equations and detailed physical description of combat, can only provide an ideal model of military operations that is too abstract and far from realistic. The shortcomings of Lanchester equations have been listed and analyzed in the literature (Ilachinski 2000; Barlow and Easton 2002; Ilachinski 1999; Lauren 2000). The main drawbacks of Lanchester equations are: they are unable to deal with the dynamics of nonlinear interaction between the combating sides; they are incapable of accommodating spatial variations of forces; the nonlinearity of warfare entails that small changes in certain critical conditions can profoundly change the outcomes; in Lanchester equations, there is no link between movement and attrition; the participants have to interact with hostile or neutral forces and respond to their actions, where the environment changes its state and causes new responses from both sides; and Lanchester equations cannot integrate human factors into combat, such as emotions, aggressiveness, fear, anger, team cohesion and trust. This makes it difficult to anticipate the behaviours of individuals by using Lanchester equations. Moreover, the nature of terrain is usually neglected and it is not possible to model the suppressive effects of weapons.

With the advent of complex systems theory and its applications in warfare studies (Barlow and Easton 2002; Ilachinski 2000; Ilachinski 1999; Lauren 2000), researchers have realized that models based on complex systems theory, particularly the set of agent based simulation tools, may address the above shortcomings of traditional equation based models of warfare. A combat can be modelled as a complex adaptive system (CAS), which adapts, evolves and co-evolves with its environment (Schmitt 1997; Lauren 2000). A complex system can be thought of, generally speaking, as a dynamical system composed of many nonlinearly interacting parts and its overall behaviours stems from some basic set of underlying principles. Agent based simulations are based on the idea that the global behaviour of a CAS derives from the low-level interactions among its constituent agents. By modelling an individual constituent of a CAS as an agent, one may simulate a real world system by an artificial world populated by interacting processes. It is particularly effective to represent real world systems which are composed of a number of nonlinear interacting parts that have a large space of complex decisions and/or behaviours to choose from (Ilachinski 1997). Thus, these new promising methodologies, agent based simulations, provide an opportunity to analyze combat by focusing on the behaviours of and interactions between the participating entities instead of the performance of specific weapons or sensors.

1.2 Motivation, hypothesis and objective

In October 1995, Project Albert was launched in the United States Marine Corps Combat Development Command in Quantico. They attempted to use modern information technologies to explore "what if" questions in military operations. Project Albert tried to capture the emergent behaviour in synthetic environments by looking at the system as a whole rather than decomposing the system into parts. Project Albert directly led to the emergence of two agent based distillation systems (ABDs) for warfare: ISAAC (Irreducible Semi-Autonomous Adaptive Combat) (Ilachinski 1997; Ilachinski 2000) and EINSTein (Enhanced ISAAC Neural Simulation Toolkit)

(Ilachinski 2000; Ilachinski 2004). They created a new era for warfare analysis. They were the first two systems which modelled warfare as a CAS. Almost all later ABDs for warfare were inspired by these two systems. The meta-technique called data farming was also first introduced and used to explore the problem space with huge amounts of data generated from thousands or millions of distillations.

With the success of Project Albert, several ABDs for warfare have been developed and employed in military analysis. MANA (Map Aware Non-uniform Automata) (Lauren 2000; Lauren and Stephen 2002b; Galligan and Lauren 2003; Galligan 2004) was developed by the Defence Technology Agency, New Zealand. MANA first introduced the concept of way-points, internal situational awareness (SA) map and event-driven personality changes. These new features largely improved the adaptability of the agents to a changing battlefield. The version released at the end of 2004 concentrated on network centric communication, including different parameters of a communication network, such as reliability, accuracy, capacity and latency. BactoWars (White 2004) from the Defence Science and Technology Organisation (DSTO), Australia, focused on problem representation and attempted to provide a simple framework which allows analysts to model real world problems more adaptively and flexibly. CROCADILE (Comprehensive, Research Oriented, Combat Agent Distillation Implemented in the Littoral Environment) (Barlow and Easton 2002) from the University of New South Wales (UNSW) at the Australian Defence Force Academy (ADFA) was the first system to use a 3D continuous environment with a higher fidelity than that of ISAAC, EINSTein and MANA.

These systems have facilitated the analysis and understanding of combat. For example, MANA has been used to explore factors for success in conflict (Boswell et al. 2003). They offer an opportunity to analyse the behaviours that we would intuitively expect in the battlefield. Through the use of these systems, defence analysts are able to gain understanding of the overall shape of a battle and those factors which are playing key roles in determining the outcome of a battle. However, the drawbacks are also obvious:

- It is hard to capture the underlying structural interaction between agents. Although existing ABDs embed the structural interaction between agents, there is no explicit model for such interactions. Therefore, it is hard for defence analysts to identify the roles of these interactions during the simulation, which is a crucial point of a CAS.

- There is no connection between tactics and strategies. Existing ABDs are developed mainly based on the reactive agent architecture (Wooldridge and Jennings 1995; Nwana 1996; Sycara 1998; Wooldridge 1999) which focuses on tactics. There is almost no interaction between tactics and strategies being modelled by existing ABDs.

- There is no reasoning on emerging behaviours during the simulation. Due to the high degree of nonlinear interaction between agents, it is impossible to reason at the agent level, which makes it hard to understand the emerging behaviours exhibited by the simulation.

- It is hard to validate and verify the system. System behaviours emerge from simple low level rules in any CAS. In current ABDs, agents are programmed without an underlying theoretically sound software architecture. Therefore, it is very difficult to validate and verify them.

- It can be a computationally expensive exercise in some systems. This can be because of a bad design, unnecessary fidelity, or fancy tools without proper modelling.

- Current ABDs are based on conventional military tactics and tend not to be approached from an overarching systems view. Concepts such as network centric warfare (NCW) (Alberts et al. 1999; Alberts and Garstka 2001; Wilson 2004), with its inherent complexity and interdependency, present challenges to identifying correct inputs at the entity level. Thus, techniques addressing higher level manipulations must be employed. The concept of NCW advocates that information, knowledge and understanding can be efficiently shared by force elements if sensor, engagement, and the command and control system are effectively and securely connected through a series of networks. Such enhanced situation awareness will allow force elements to access the right information at the right time. It is anticipated that it would produce information superiority in battle field operations. In turn, information superiority dramatically increases the power of combat by speeding up the efficiency of command and making better decisions.

These drawbacks limit the ability of existing ABDs to study real world problems with high nonlinear interactions, e.g. NCW. The hypotheses of this thesis are:

> ***Hypothesis 1****: Results obtained from ABDs where there is no strategic planning or no coordination among agents can be misleading and cannot be generalized.*
>
> ***Hypothesis 2****: Network analysis and interaction are suitable for reasoning at the group level (explaining emergence) in complex systems.*

In this thesis, the following objectives are achieved to address above drawbacks.

1. Establish an understanding of combat as a CAS.

2. Identify theoretical causes of the drawbacks in existing ABDs for combat, and propose an innovative framework or agent architecture to overcome them.

3. Develop a theoretically sound ABD with strategic planning and characterize the solution space in military operations.

4. Conduct scenario based military analysis on military operations in the urban terrain (MOUT) to test our hypotheses.

1.3 Organisation of the thesis

The thesis is composed of nine chapters. The first chapter presents a brief background of the research topic, followed by the motivation and objectives of this thesis. Then the outline of the thesis is described. The chapter concludes with the major contributions of this thesis.

In chapter 2, a comprehensive literature review is undertaken in modelling and simulation of combat. The review consists of three aspects: traditional models and simulations, current agent based models and simulations, and emergence. This achieves part of objective 1 and 2.

In chapter 3, WISDOM-I is proposed. The features and characteristics of this system are described in detail. This achieves part of objective 1 and 2.

In chapter 4, a fitness landscape analysis is conducted based on WISDOM-I. Such an analysis is used to identify the degree of difficulty to search for optimal solutions in military operations. This achieves part of objective 1 and 2.

In chapter 5, the essential characteristics of a MAS for CAS is first investigated. Then NCMAA is proposed and its implementation is discussed by combining network theory and CAS. The system development cycle is presented at the end of the chapter. This achieves part of objective 2.

In chapter 6, WISDOM-I is re-designed and re-developed as WISDOM-II. A number of unique features are proposed and discussed in detail. At the end, a simple scenario analysis is conducted to exemplify the usage of WISDOM-II. This achieves part of objective 3.

In chapter 7, the fitness landscape is analysed to characterize the solution space of certain military operations based on WISDOM-II. The results are compared with those based on WISDOM-I from chapter 4 to show the difference between the two versions. This achieves part of objective 3.

In chapter 8, a series of military analyses is conducted on MOUT. A set of scenarios and urban terrains are built. Then the effect of force size, firepower and communication on the outcomes of combat are investigated and analysed. A number of interesting findings are discussed. This achieves objective 4.

Chapter 9 is the last chapter of the thesis. It concludes the thesis with a discussion of possible directions for future research.

1.4 Major contributions

The major contributions of the thesis fall into three classes:

1.4.1 MAS for CAS

A novel multi-agent architecture "NCMAA" is proposed in the thesis (Chapter 5). NCMAA is based purely on social network analysis and CAS principles. The system is designed on the concept of networks, where each operational entity in the system is either a network or a part of a network. The engine of the simulation is also designed around the concept of networks. While many MASs can be considered as operating on the concept of networks, designing and implementing the system on the concept of network is a more powerful approach because it provides the foundations for a new type of reasoning in these systems.

Existing network analysis focuses on the dynamics within one single network. However the interaction between networks has little been explored. This is a drawback in network analysis which we try to address in this thesis. The interaction between networks is the key to understand the system's behaviour.

The most important advantages of NCMAA are concluded as follows:

- It easily analyses the interaction between agents. MAS consists of a large number of interacting agents. Existing architectures, e.g. BDI or reactive agents pay more attention on how to model an individual agent. Some efforts have been made to model agents at the group level, such as Agent-Group-Role structure (Ferber and Gutknecht 1998) and YAMAN (Savall et al. 2001). However, there is no explicit structural presentation for interactions among agents. The new architecture is built on network theory. Each type of interaction or relationship among agents forms a network. Conducting network analysis will allow analysts to gain insight of these interactions and therefore gain better understanding of the whole system.

- It provides a chance to capture the interaction or influence between interactions. Based on a real-time network centric reasoning for the reciprocal interaction of networks, policy analysts, decision makers or any other users may capture the influence of one relationship on another relationship easily. Combining this with the previous point, NCMAA may help analysts to capture, analyse and understand the dynamics or patterns within a CAS.

- It establishes for the first time a formal framework for reasoning at the group level in ABDs. There is no reasoning in the reactive agent architecture while the reasoning is conducted at the individual level which leads to quite low scalability of the system in the cognitive agent architecture. No evidence prevents reasoning at the group level. A reasoning engine which conducts reasoning at the group (network) level is proposed. Such a reasoning engine conducts real time reasoning to allow analysts to understand the results during the simulation.

1.4.2 Military analysis

A new ABD for combat "WISDOM-II" is proposed in the thesis, which is the first simulation system purely inspired by both network theory and CAS (Chapter 6). With NCMAA, WISDOM-II facilitates the analysis and understanding of warfare.

WISDOM-II is the first ABD for combat with built-in network analysis tools. Based on network theory, WISDOM-II conducts network analysis automatically for each type of interaction within the system. Therefore during the simulation, it is easy to analyse the dynamics and capture the patterns of such interactions as they are evolving.

WISDOM-II is the first ABD for combat that is able to provide group-level real-time reasoning during the simulation. Based on a pre-defined causal model, reasoning is done through a sequential bi-variate time-series auto-correlation with path analysis and root cause analysis. Based on this real-time reasoning, emerging behaviours can then be interpreted in natural language and presented to defence analysts. Even without any knowledge of information technology, the defence analyst may still understand what is going on during the simulation.

WISDOM-II is the first ABD for combat which connects tactics and strategies. Any military operation involves different decision making mechanisms at different levels. The interaction between decision making mechanisms at different levels directly influences the outcome of military operations. WISDOM-II provides a platform for defence analysts to identify the role of decision making mechanisms in military operations.

WISDOM-II is the first ABD for combat with an explicit model of recovery in military operations. The recovery capability plays a crucial role in military operations. However it has not been addressed by existing ABDs. A model of artificial hospital is proposed and implemented in WISDOM-II. With this model, defence analysts can study the interaction between resource requirements and recovery capabilities.

WISDOM-II is the first ABD for combat which can employ heterogenous agents at the squad level. Although it increases the complexity of the system, it is more close to reality and the dynamics can be analysed easily through the above advantages of the new architecture. WISDOM-II supports various levels of command and control (C2) networks. It allows defence analysts to investigate the effect of C2 structure on the performance of a force with the aid of NCMAA.

Communication plays a crucial role in modern warfare. A comprehensive model of communication network is proposed and developed in WISDOM-II. The model includes a set of parameters, such as range, loss probability, latency, noise, etc. With such a model of communication network, various communication types can be modelled and simulated in WISDOM-II, such as point to point communication and broadcast communication, and direct communication and indirect communication. Initializing the communication network with different types of networks, e.g. small world network or scale free network, allows defence analysts to study the role of communication network topology in combat.

Global urbanization causes the focus of military operations to shift from open terrain to urban terrain. The features of urban terrain largely influence various aspects of military operations. However, the challenges in MOUT have little been explored in the literature. In the thesis, a set of stylised urban terrains is created, on which a series of scenarios are simulated by using WISDOM-II. The results are then analysed, based on which a number of findings are summarized (Chapter 8).

1.4.3 Evolutionary computation

Evolutionary computation (EC) techniques have widely been used in searching for optimal solutions for a problem. However, their value has not been fully recognized in the military domain. In the thesis, EC techniques are adopted through a fitness landscape analysis (Chapter 4 and 7).

Genetic algorithm (GA) and the fitness landscape have been adopted in ISAAC (Ilachinski 1997) and EINSTein (Ilachinski 1999) to search for optimal solutions for a predefined scenario. However, the characteristics of the solution space in combat simulations have not been studied. It is essential to understand the underlying nature of the solution space to gain insight of the problem difficulties. The fitness landscape analysis conducted in the thesis presents the characteristics of solution space associated with warfare simulations by both WISDOM-I and WISDOM-II. It helps analysts to gain better understanding of the nature of warfare, the difficulty of these problems, and gain insights into possible performances of different solutions.

Chapter 2

Multi-Agent Systems and Combat

2.1 Introduction

Thousands of years ago, human beings had already started to study warfare. The most famous book is called *The Art of War* (Griffith 1963) written by Sun Tzu, the Chinese strategist, in about 500 B.C. With economic and social developments, war has continuously been evolving. Since it is undesirable to provoke a real war to test a strategy, red teaming (Mateski 2004; Sandoz 2001; DOD 2003) has widely been adopted in military analysis. Almost all countries conduct military exercises to test force capabilities, new concepts and technologies. However, high fidelity human based red teaming is extremely expensive in terms of money and time. Recently low fidelity computer-based red teaming has been recognized as a very valuable tool in military analysis (Ilachinski 2004).

This chapter begins with a brief overview of red teaming in combat. Then conventional combat models and simulation systems are reviewed followed by a detailed survey of agent based combat systems. Finally, the concept of emergence is reviewed. Figure 2.1 presents a structural overview of modelling and simulation of combat and Table 2.1 is a summary of the literature review in this chapter.

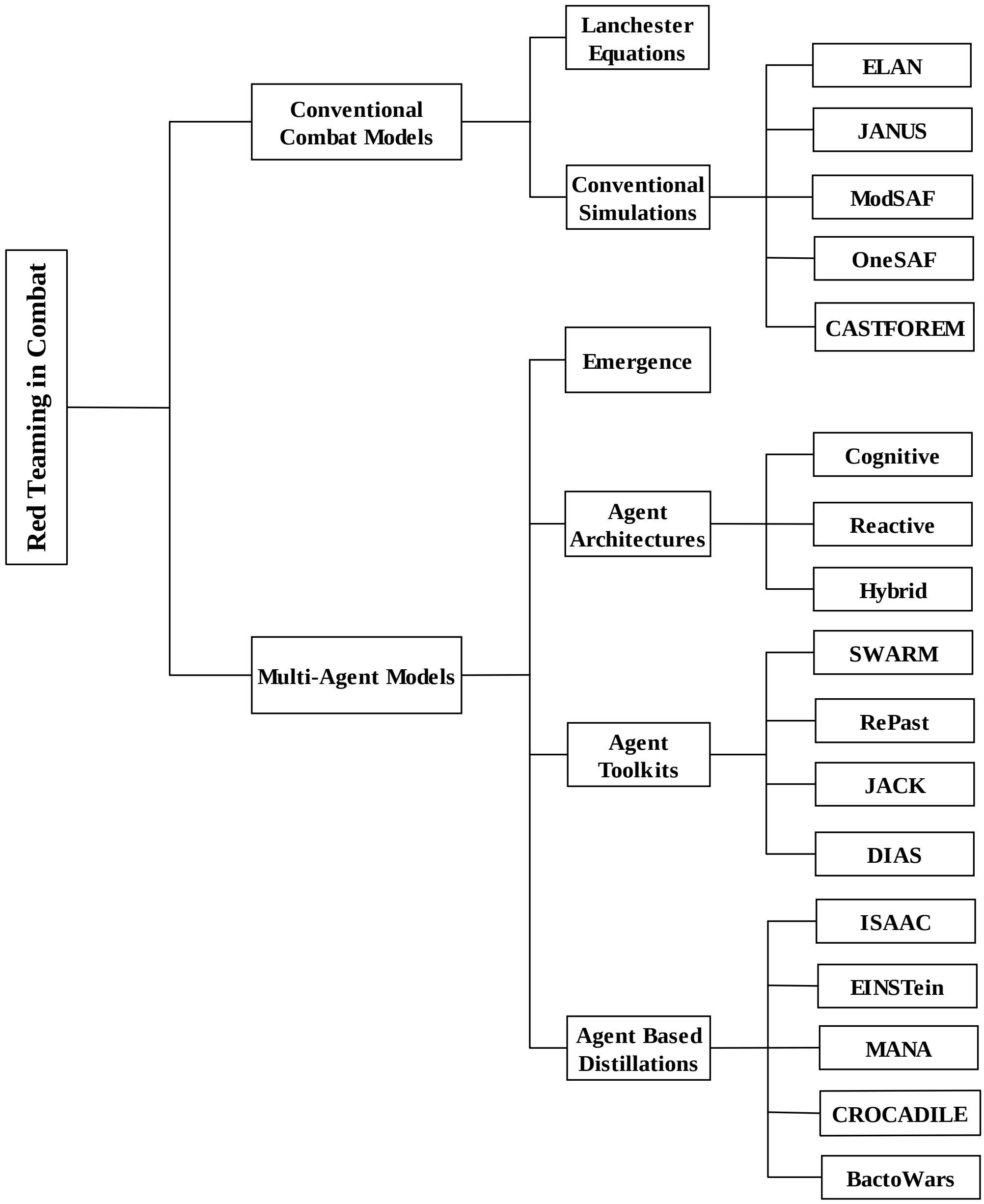


Figure 2.1: Overview of modelling and simulation of combat

2.2 Red teaming in combat

Red teams and red teaming processes have long been used as tools by various organizations such as government, defence and commercial organizations, to study a problem, a system, a plan, the way of thinking or a concept by anticipating adversary behaviours. Their purpose is to mitigate an organization's risk and increase its opportunities. Usually there are two sides in the red teaming process: "BLUE"

Table 2.1: Summary of literature review

Topic	**Year**	**Authors**
Red Teaming	2001	Sandoz
in Combat	2003	DoD
	2004	Mateski
Lanchester	1916	Lanchester
Equations	1964	Dolansky
	1984	Battilega and Grauge
	1987	Athans
		Lepingwell
	1995	Hartley
	1998	Fricker
	2001	Aragon
		Chen and Chu
	2002	Sheldon
	2003	Lucas and Turkes
Conventional	1994	Ceranowicz
Combat	1995	Bennington
Models		Caldwell and Wood
	1997	Larkin
	1998	Sawyers
	2000	Maxwell
	2001	Stone and McIntyre
		Wittman and Harrison
	2002	Fields
	2003	Simlote
	2004	Bates and McIntyre
	2005	MSRR
	2006	CACI International Inc (a)
		CACI International Inc (b)
		CACI International Inc (c)
Multi-Agent	1986	Brooks
Based	1987	Chapman
Models		Georgeff and Lansky
		Reynolds
	1990	Brooks
	1991	Kaelbling
		Maes
		Brooks (a)
		Brooks (b)
	1995	Rao and Georgeff
		Wooldridge and Jennings

Topic	Year	Authors
	1996	Nwana
	1997	Bonasso, Firby, Gat, Kortenkamp, Miller, and Slack
		d'Inverno, Kinny, Luck, and Wooldridge
	1998	Sycara
		Huhns and Singh
	1999	Ferber
		Wooldridge
	2000	Bryson
	2002	Dastani and Torre
		Nolfi
Multi-Agent Toolkits	1995	DIAS
	1996	Minar, Burkhart, Langton, and Askenazi
	1998	Campbell and Hummel
	1999	Wilensky and Stroup
	2002	Gilbert and Bankes
		Inchiosa and Parker
		Kota
		Serenko and Detlor
	2003	Collier, Howe and North
	2004	Luke, Cioffi-Revilla, Panait, and Sullivan
		Tisue and Wilensky
		Tobias and Hofmann
	2005	Agent Oriented Software Pty. Ltd. (JACK)
	2006	North, Collier and Vos
Multi-Agent Combat Models	1992	Beyerchen
	1997	Ilachinski
		Schmitt
	1998	Holland
	1999	Beckerman
		Goldstein
		Ilachinski
	2000	Brown
		Ilachinski
		Lauren
	2001	Grieger and Gill
		Lauren (a)
		Lauren (b)
		Lauren (c)
		Parunak and Brueckner
	2002	Barlow and Easton
		Cioppa
		Epstein

Topic	Year	Authors
		Grieger
		Lauren and Stephen
		Lauren
		Parunak, Brueckner and Sauter
		Odell
	2003	Bar-Yam
		Barlow
		Galligan and Lauren
		Galligan, Anderson, and Lauren
		Ilachinski
	2004	Bar-Yam
		Galligan
		Ilachinski
		Parunak and Brueckner
		White
	2005	De Wolf and Holvoet
		Wheeler
	2006	Perry

and "RED". The common role is that the "BLUE" side attempts to find the risk through the eyes of an adversary or competitor, the "RED" side. Mateski proposed a broad definition of red teaming (Mateski 2004).

> "Red teaming involves any activity - implicit or explicit - in which one actor ("BLUE") attempts to understand, challenge, or test a friendly system, plan, or perspective through the eyes of an adversary or competitor ("RED")"

Red teaming is a connotation for playing the devil; trying to penetrate the mind of the enemy or competitor and simulate their behaviours; understanding risk in the eyes of the opponent and mitigating vulnerabilities before it is too late. Red teams, in this context, are specially selected groups designed to anticipate and simulate the decision-making and behaviours of potential adversaries. Red teaming has already been recognized as an especially important tool by defence organizations (DOD 2003). It can deepen and widen understanding of options and behaviours of adaptive adversaries which help us find risk and vulnerabilities in existing strategies, postures, plans, programs, and concepts, challenge the people which adhere to old theory and doctrine that previously led to success, and train warfighters to be veteran. Defence organizations realized that red teaming can also be applied into all levels of military activities (Sandoz 2001; DOD 2003):

- Strategic level: challenge strategy assumptions and visions;
- Operational level: challenge force intelligence, organization and operation plan;
- Tactical level: challenge military training and system vulnerabilities.

Perceptions influence decision-making at all strategic, operational and tactical levels. Each side seeks to capture the other's intent and courses of action, and to maintain an advantage that will lead to the fulfilment of its own strategy and objectives. Due to the difference between societies, cultures and values, war has an inherently asymmetric nature proved through military history. Red teaming provides the means to simulate possible opponents by creating an environment where both sides try to adapt to their opponent's action. In such an environment the military can test, improve and substantialize new concepts, e.g. NCW, by refining strategic, operational and tactical issues, e.g. doctrine, organization, training, material, leader development, personnel and facilities issues (Sandoz 2001).

There are two types of red teaming: human-based red teaming and software-based red teaming. In human-based red teaming, a real force is divided into two teams; one simulating the enemy (red team) while the other simulating friends (blue team). Defence uses this approach in their operations' exercises. However, this approach is extremely expensive and does not enable analysts to explore all aspects of a problem. Computer simulations of multi-agent systems are used for software-based red teaming. These simulations explore abstract higher level scenarios of different vulnerabilities in a plan or operation. Once the weaknesses in the system are identified and a risk analysis is conducted, human-based red teaming can be used in a more focused way to increase the fidelity of the analysis.

In order to conduct red teaming, a number of combat models and simulation systems have been developed. Some of these models and simulations are reviewed later in this chapter.

2.3 Conventional combat models

2.3.1 Lanchester equations

Traditionally, defence analysts adopted what is known as Lanchester Equations (LEs) to model and theorize about combat attrition (Dolansky 1964; Lepingwell 1987; Chen and Chu 2001; Sheldon 2002; Lucas and Turkes 2003). LEs were introduced by F. W. Lanchester in 1916 (Lanchester 1916) as a set of linear differential equations that treat attrition as a continuous function over time and assume the attrition is proportional to the force size in combat.

Suppose that there are two forces, the blue and the red. $B_s(t)$ and $R_s(t)$ are the force size of the blue and red respectively, and α_b and α_r are the attrition coefficient for the blue and red force respectively. The attrition coefficient is the amount of attrition of one side caused by one unit of the other side. The LE linear law is defined by Equation 2.1, which is used to model "indirect fire" between the two forces. Both the blue attrition and the red attrition are proportional to the product of the blue force size and the red force size.

$$\begin{cases} \frac{dB_s}{dt} & = & -\alpha_b B_s(t) R_s(t) \\ \frac{dR_s}{dt} & = & -\alpha_r B_s(t) R_s(t) \end{cases} \tag{2.1}$$

The "direct fire" model of attrition is given by Equation 2.2, which is called the LE square law. In this case, the blue attrition is proportional to the red force size while the red attrition is proportional to the blue force size.

$$\begin{cases} \frac{dB_s}{dt} &= -\alpha_r R_s(t) \\ \frac{dR_s}{dt} &= -\alpha_b B_s(t) \end{cases} \tag{2.2}$$

Equation 2.3 is the LE mix law, which is used to model the situation that the blue is killed through "direct fire" by the red team and the red is killed through "indirect fire" by the blue team.

$$\begin{cases} \frac{dB_s}{dt} &= -\alpha_b R_s(t) \\ \frac{dR_s}{dt} &= -\alpha_r B_s(t) R_s(t) \end{cases} \tag{2.3}$$

The LE logarithmic law is defined by Equation 2.4, which is used to model non-combat losses (Sheldon 2002). The attrition of each force is proportional to its own force size. It implies that the damage is caused by its own force.

$$\begin{cases} \frac{dB_s}{dt} &= -\alpha_b B_s(t) \\ \frac{dR_s}{dt} &= -\alpha_r R_s(t) \end{cases} \tag{2.4}$$

LEs are very intuitive and therefore easy to apply. However they can only provide an ideal model of military operations that is too abstract and far from realistic as discussed in the previous chapter. Research on historical battles also shows that LEs cannot accurately match historical attrition data (Battilega and Grauge 1984; Hartley 1995; Fricker 1998; Lucas and Turkes 2003). However, because of their simplicity, LEs are still widely used in military analysis.

2.3.2 Conventional combat simulations

2.3.2.1 ModSAF

ModSAF (Modular Semi-Automated Forces) (Ceranowicz 1994) is an interactive, high resolution simulation system at entity level. It can simulate combined arms tactical operations up to the battalion level. It provides SAF (Semi-Automated Forces) simulation capabilities in an open architecture with a set of software modules and Computer Generated Forces (CGF) applications. These modules and applications are used to model the battlefield including physical, behavioral and environmental elements. ModSAF is a fully distributed system and allows unit simulation running over many computers while allowing one operator to control the entire unit. This open architecture allows it to scale up to the simulation of larger units while increasing simulation realism. ModSAF consists of three components: the SAFstation, the SAFsim, and the SAF-logger. The SAFstation (SAF workstation) provides the graphical user interface for the user. The SAFsim (SAF simulator) simulates all the entities, units, and environmental processes. Seven systems can be modelled in ModSAF: air defence, intelligence, mobility and survivability, combat service support, command and control, and fire support system. The SAF-logger records necessary information during the course of the simulation. However, all human behaviours are predefined and hard-wired in the source code in ModSAF. Therefore it is hard to use it as an exploratory analytical combat engine in military analysis.

2.3.2.2 JANUS

Janus (Caldwell and Wood 1995) is an event-driven, multi-sided, ground and air-ground simulation with conventional and chemical weapon systems. The computer-generated forces (CGF) are simultaneously directed and controlled by a set of players for each side who only have limited knowledge of enemy units. It is played on a computer-generated digitized terrain map. The terrain features include: elevation (i.e. contour lines), roads, rivers, vegetation, buildings, etc. All these features are differentiated visually by different colours. The weapon system is modelled through a series of detailed properties and the outcome of engagement is totally based on predefined probability distributions. The operational functionality of various kinds of systems in manoeuvre and artillery operations, such as direct fire, crew-served, ground weapon system, artillery system and so on, can be easily simulated in Janus. The interactions of these subsystems and the impact of the environment, e.g. weather, time of day and terrain features, on acquisition and engagement are modelled at a high level of fidelity.

2.3.2.3 ELAN

ELAN (Extended LANchester model) (Sawyers 1998) is a simple land combat simulation for joint operations, developed by the U.S. Army Training and Doctrine Command Analysis Center. It is a combat model at a medium level of resolution and focuses on terrain and tactics. The maximum size of tactical terrain is two kilometers. It is very fast and has a set of online analysis tools which allow users to easily analyze the simulation outputs. ELAN can output detailed information about killers and victims, both from unit on unit and weapon system on weapon system perspectives.

2.3.2.4 OneSAF

OneSAF (Wittman and Harrison 2001; Fields 2002) is a real-time distributed interactive simulation developed by the U.S Army Simulation, Training and Instrumentation Command. It can be used to model operations, systems and processes from the individual combatant, such as soldiers, tanks, and helicopters, through aggregate units to the Brigade level. The simulation can be run in a fully automated mode or under the control of human participants. The composibility of the simulated entities, units, behaviours and the synthetic environment makes it very flexible to develop various scenarios to explore the problem space. This flexibility also allows the analysts to test and verify new concepts by creating new equipment and behaviour combinations.

2.3.2.5 CASTFOREM

The Combined Arms and Support Task Force Evaluation Model (CASTFOREM) (MSRR 2005) is designed for evaluation of weapon systems and tactics in Brigade and smaller combined arms conflicts. It is a two-sided, closed-form, event-sequenced, high resolution simulation system for conventional and directed energy weapon systems with resolution to the item system level. Closed-form means it is not an interactive simulation, like Janus. It does not require human participation. In CASTFOREM, all events are stochastically modelled, e.g. probability of detection, probability of hit, and probability of kill while line of sight is deterministic. It is primarily used to model intense battle at Brigade or Battalion level up to one and one half hours in length. It can be used to model a range of operations including ammunition resupply, close combat, combat service support, logistics, fire support, intelligence and electronic warfare, etc. However, since it is a high resolution simulation system, designing a single scenario requires a very long time. Therefore, it is not suitable for exploratory analysis (Ilachinski 2004).

2.3.2.6 Other simulation models

In addition to the above models, there also exist many other simulation systems, such as JWARS (CACI 2006b; Maxwell 2000; Stone and McIntyre 2001; Simlote 2003), JMASS (CACI 2006a; Larkin 1997), and JSIMS (CACI 2006c; Bennington 1995). All these simulation systems are very useful and widely employed for various purposes, such as training and defence operation analysis. However, the shortcomings are also well documented. As discussed by Ilachinski (Ilachinski 2004), and Bates and McIntyre (Bates and McIntyre 2004), most of these systems are very complex, high resolution, usually hard-codes features in source code or hardware, are hard to interface with, have limited data collection and analysis facilities, unable to model information operations, require real-time run times, etc. High fidelity and real-time mode make these simulation systems very difficult to be used as exploratory analysis tools for defence (Ilachinski 2004). Exploratory analysis requires the development of smaller, low fidelity, abstract combat simulation systems which may help defence analysts to develop and verify concepts and principles, and answer "what-if" questions in military operations.

2.4 Agent architectures and simulation toolkits

Agent-based simulation is a very popular exploratory tool for the study of CAS. The key idea of CAS is that the global behaviours of a real system emerge from simple, low-level interaction among the system constructive components. Therefore analyzing the interaction among agents, the dynamics and emergent behaviours through simulations may help analysts to gain insight into the real system which is modelled as a MAS.

The fundamental building block of most CAS models is the adaptive, autonomous, intelligent agent. The way to design and build agents is called agent architecture, which specifies how to define the constructive component modules of the agent, the interactions between these modules, the way to build knowledge, the actions taken by the agent and the control mechanism (Maes 1991; Kaelbling 1991; Ferber 1999). When building a MAS for a real system, the system developer has to answer such questions as (Ferber 1999): how can the various parts of an agent be integrated so that it may behave as what the designer expects?, does the agent have a symbolic or logic representation of its environment?, what kind of decision mechanisms does the agent use to respond to its environment?, etc. There are two schools at the opposite ends of agent architectures: cognitive and reactive agent (Wooldridge and Jennings 1995; Sycara 1998; Huhns and Singh 1998; Ferber 1999; Wooldridge 1999; Bryson 2000; Dastani and Torre 2002).

2.4.1 Cognitive agents

The cognitive agent can be seen as a knowledge based system, which includes all the necessary data and knowledge to make plans, to collect necessary resources, to respond to its environment and to interact with other agents by cooperating, communicating and negotiating in order to achieve its goals. Usually the cognitive agent is intentional, which means all actions are taken in order to achieve its goals; is rational, which means the actions it takes always maximise its own utility; and has a representation, usually a logical representation of its environment, which is used to conduct reasoning. With its internal representation and reasoning mechanisms, the cognitive agent may function independently of other agents and fulfil relatively complicated tasks individually. These features make it more flexible in terms of its behaviour. Since it conducts reasoning based on its internal representation of the world, the cognitive agent is able to memorize and analyze the situations it encounters. Therefore the cognitive agent may predict the possible outcomes of its actions and then make plans for achieving its goals (Ferber 1999). A typical example of the cognitive agent architecture is BDI (Belief-Desire-Intention) (Rao and Georgeff 1995; Wooldridge and Jennings 1995; Nwana 1996; Sycara 1998; Wooldridge 1999).

The BDI architecture is established on the concept of practical reasoning, which includes two processes: deciding on what the agent needs to achieve and deciding on how to achieve it. Each agent has three key internal models: belief, desire and intention. The belief of an agent represents its knowledge of its environment and itself. As the environment changes, the agent attempts to accumulate experiences, checks the consistency of its beliefs with the accumulated experiences, and modifies its beliefs accordingly. Based on its current beliefs, the agent generates a number of options and selects one of them as its goal or desire. The agent then chooses one of the many possible paths to reach its goal or satisfy its desire. The path chosen by an agent represents its intention.

The Procedural Reasoning System (PRS) (Georgeff and Lansky 1987; d′Inverno et al. 1997) is developed on the BDI architecture. PRS consists of four components connected by an interpreter which directs the process of acting, sensing and reasoning. The first component is the database of beliefs or facts about the world. It consists of the knowledge of the environment and the internal states of the agent. The second component is a set of desires, goals or tasks. It is represented as a series of behaviours the system might execute. The third component is a set of plans. They are pre-defined procedures to express the actions and test to achieve the goals or to react to stimulations. These procedures also define the way to manipulate the belief database to generate the next action or perhaps to produce new knowledge. The last component is a sequence of intentions. It specifies the current operating procedures and the series of procedures that will be operated one after another. Through such architecture, PRS could conduct meta-level reasoning, e.g. how to select procedures from a set of procedures.

BDI provides human-like reasoning processes. First it decides what to achieve, then selects a way from possible options and finally takes an action to achieve it. It is intuitive and easy to understand. However, to obtain a logical representation of a problem, two main issues arise (Wooldridge and Jennings 1995; Bryson 2000): how to transfer a real world problem to a logical description and how to represent information of real world entities and processes in a logical format. Since any action is selected through a series of reasoning on the sophisticated representation about the world, scalability is an issue when working with BDI. In addition, Ramamohanarao (Ramamohanarao et al. 2001) argued that current BDI models lack concurrent control, recoverability, and exception handling mechanisms. Some efforts have been made to address these issues. Minsky (1986) proposed an alternate solution to formal logic-based systems. He viewed the human mind as a society of numbers of connected mindless agents. Minds rise from the organisation of interactions between agents instead of from a unified and coherent reasoning. In order to formalise social functions, Castelfranchi (2001) proposed a learning-without-understanding process as non-intentional mental processes. The intentional and rational system of the agent is supersedes by the a metalevel controller.

2.4.2 Reactive agents

The reactive school advocates that exhibiting intelligent behaviours in a system does not require each agent within the system to be intelligent individually, i.e to have a representation about the world and a reasoning mechanism (Ferber 1999; Nolfi 2002). Intelligence can be generated by the interactions between agents, and between agents and their environment where they are situated. The intelligent behaviours may emerge from a set of simple local rules (Brooks 1990; Brooks 1991a; Brooks 1991b). The reactive agent is directed either by its internal drives, such as the motivation mechanism which forces it to accomplish some tasks (e.g. maintaining its energy level), or external stimulations based on which set of rules the agent uses to make decisions. Unlike BDI agents, reactive agents do not have any representation of their environment or any reasoning mechanism. Instead they take actions based on the stimulus received from the environment or other agents. Reactive agents do not make any plans, they make decisions according to the limited information available to each of them and the current status of their environment. Although each individual reactive agent is not intelligent when compared to a cognitive agent, the power of reactive agents comes from their capacity to form a group and the capacity for adaption and evolution which emerges from their interactions. The intelligence in the reactive agent architecture is exhibited at the group level rather than the individual level, e.g. swarm intelligence of an ant colony.

The reactive agent architecture is simple, economic, and robust, and can scale up well because there is no representation about the world and reasoning engine. The intelligence comes from a population of agents which allow the reactive agent architecture to exhibit a high level of fault tolerance. In addition, since its actions are selected purely based on sensing, the reactive agent reacts fast to changes in the environment. This feature makes the system agile. Furthermore, any action is taken based on current information because there is no need to store information. Thus, there is no problem associated with knowledge fusion.

However, several weakness have been identified. The lack of memory may make it very hard to improve performance over time of reactive agents (Sycara 1998; Wooldridge 1999). Without planning and inference mechanisms, the reactive agent does not necessarily reason about its actions and it is hard to understand the relationship among local behaviours, environment and global emerged behaviours (Wooldridge 1999). Because they act upon the local information accessible to each of them, they may act unpredictably and sometimes the system can exhibit unstable behaviours (Sycara 1998). Finally, the mechanism of selecting actions in the reactive agent architecture is predesigned by the developer. This limits the complexity of the problems which can be addressed by these systems (Bryson 2000).

The typical example of the reactive agent architecture is the subsumption agent architecture proposed by Brooks (1986). The subsumption architecture breaks an agent into vertical modules, each of which is responsible for a limited type of behaviours. Each module is computationally simple and independent, and consists only of finite state machines. A finite state machine explicitly specifies the states the module can be in and also defines the output states that can be reached from a given state and the input to the machine. The modules function in parallel. The interactions between the modules are predefined, fixed and effected through a dominance relationship. If two modules are in conflict, the result from the dominated module will be ignored. Although they are extremely simple without any explicit reasoning, the resulting systems still exhibit the intelligence and ability to accomplish complicated tasks.

Another example of the reactive agent architecture is Reynolds's Boids model (Reynolds 1987), a very famous artificial life model, which attempts to model flocking behaviour in nature. There are only three very simple "steering rules" to control the behaviours of the agent (called boids) in the Boids model:

- Separation: maintain certain distance with other local boids;
- Alignment: move towards the centre position of local boids;
- Cohesion: try to match velocity with local boids.

Although these three rules are very simple, the interactions between simple behaviours of individuals do produce complex organized group behaviour, a believable flocking behaviour. The model also demonstrates the unique property of complex systems that a complex system behaviour can emerge from a set of local simple rules.

2.4.3 Hybrid agents

Due to the limitations of both architectures, for most real world problems neither purely cognitive architecture nor purely reactive agent architecture is appropriate. Therefore, many researchers have been trying to develop a hybrid strategy, where a reactive system is designed with a cognitive planner. Usually, a hybrid architecture consists of two components: a cognitive part, containing a symbolic world model and reasoning engine, which develops plans and makes decisions; and a reactive part, which is able to react to stimulations from the environment (Wooldridge and Jennings 1995).

Typically, hybrid systems are implemented with two or three layers in a hierarchy (Wooldridge and Jennings 1995; Bryson 2000). The top layer is a traditional constructive planner and deals with the information at a high level of abstraction. It manipulates or generates a representation about the world, makes long term plans or deals with issues, such as cooperation, coordination and negotiation. It does not act directly on behaviour primitives. The bottom layer is a reactive actuator, which makes decisions directly based on its raw sensor inputs and a set of predefined rules. The middle layer has a kind of implicit knowledge and predefined plan fragments. It selects plans based on the current situation rather than make plans (Bryson 2000).

An example of a hybrid architecture is the 3T system (Bonasso et al. 1997), which includes three layers from bottom to top: skill manager, sequencer and planner. The bottom layer of skill managers is the reactive layer in the system, which includes a set of situated skills. Each skill is specified by the input, output, and when and how it functions. The role of the reactive layer is to control real-time behaviour. The task of the sequencer is to coordinate the skills and configure the skills for the current situation. The role of the sequencer is to control the order of real-time behaviours. The planner is the cognitive layer in the system, which makes plans at the highest level of abstraction. It searches a space of predefined plans and tries to decompose a highly abstract goal into more detailed sub-goals by using preconditions and effects. Although planning based on searching has been proven to be an unrealistic model of intelligence (Chapman 1987), 3T has been successfully used on numerous robots from an academic mobile robot to robotic arms that deal with hazardous substances (Bryson 2000).

The multi-layer architecture has shown its capability for dealing with complex problems in real environments (Bryson 2000). However it still has some problems such as how to optimize the planning mechanism for the top layer, and how to handle the interaction between layers, e.g. how to manipulate and exchange information between layers.

2.4.4 Multi-agent based simulation toolkits

To build a MAS is a complex task and requires both domain knowledge and a high level knowledge of information technology, e.g. programming language. Therefore, a number of multi-agent based simulation toolkits have been developed to facilitate the development process and reduce the workload by introducing libraries. The following are four famous examples of MAS simulation toolkits.

2.4.4.1 SWARM

SWARM (Minar et al. 1996) is one of the first reactive agent-based modelling software libraries and was originally proposed by the Santa Fe Institute. SWARM is centred on the concept of swarm and implemented in an Object-Oriented Programming language. Each object in SWARM has three attributes: name, data and rules.

Name is the identification of this object. Data includes all information the object obtains. The rules define the way that the object sends messages to other objects. Sending messages is the only way to communicate between objects. There is no complex data exchange in SWARM. The basic building block of SWARM is a swarm, a collection of agents with a sequence of actions. There are two types of swarm: model swarm and observer swarm. The model swarm is used to model the characteristics of the real system. The observer swarm is acting as an environment of model swarms and provides inputs to the model swarm and takes outputs from the model swarm. It is also responsible for data analysis and providing a user interface. In SWARM, swarms can themselves be agents while an agent can also be a swarm. So SWARM supports hierarchical structures. Through such a hierarchical structure, users may easily build multi-level models. The modularity and composability of swarms makes SWARM very flexible.

2.4.4.2 RePast

The Recursive Porous Agent Simulation Toolkit (RePast) [1] is an open source, agent-based simulation toolkit created by Social Science Research Computing at the University of Chicago (Collier et al. 2003; North et al. 2006). RePast is composed of a set of software libraries for building, running, visualizing, and collecting data from an agent-based simulation, and implemented in three programming languages: JAVA, Python and Microsoft .NET. It borrows a lot of concepts and design ideas from SWARM. The main differences from SWARM are enhanced adaptive features, such as a generic algorithm and regression, and an enhanced visualization component. RePast may visualize information with a series of charts, e.g. histograms and sequence graphs, and can also take snapshots of running simulations and show them in a 2D QuickTime movie format. The key features of RePast are as follows:

- Repast is a discrete event simulation platform with a fully concurrent scheduler which supports both sequential and parallel discrete event operations.

[1] http://repast.sourceforge.net/index.html

- The stochastic processes in RePast are modelled based on the Monte Carlo technique.
- Users can access and modify agent properties, agent behavioural equations, and model properties during the simulation.
- Genetic algorithms, neural networks and some specialized mathematics are implemented as software libraries in RePast.
- Since Repast is initially developed for social science research, it includes a "network library" which can be used to create social networks and calculate some network measures.
- Repast is able to inter-operate with Geographical Information Systems (GIS).

2.4.4.3 JACK

JACK Intelligent Agents (Kota 2002; JACK 2005) is a commercial agent oriented development environment from Agent Oriented Software [2]. JACK is composed of six components: agents, capabilities, belief sets, views, events and plans.

- Agents – the basic entities that conduct reasoning in JACK and are defined by what kind of capabilities they have, what type of messages and events they may respond to and which plans they will use to achieve their goals.
- Capabilities – the way to implement a function through other components. they can include events, plans, belief sets or even other capabilities.
- Belief sets – a collection of beliefs/data representing the status of other agents and the environment. A belief set looks like a database in which beliefs are represented in a "first order, tuple-based relational model".
- Views – the tool to integrate any data and present in a way that is easily manipulated by JACK.

[2] http://www.agentoriented.com/

- Events – which specify the actions and corresponding messages.
- Plans – which define context dependent responses to event occurrences. A plan is a list of procedural descriptions of what actions an agent will take to handle a given event. All the actions that an agent takes are prescribed and described by the agent's plans.

JACK is built on the BDI agent architecture. The beliefs of agents are represented by the belief set and can be modified during the course of simulations. Desires of agents are modelled by the goals or events that they are trying to reach or handle. The intentions of agents are defined by the plans that they use to achieve the goals or to handle the events. A JACK agent may have multiple plans to achieve a single goal. Therefore if one plan fails, the agent may use other plans to achieve the goal.

2.4.4.4 DIAS

The Dynamic Information Architecture System (DIAS) [3] is a flexible, extensible, object-oriented framework for building complex multidisciplinary simulations (DIAS 1995; Campbell and Hummel 1998). The main components of DIAS are software objects (entity objects) representing real-world entities, and simulation models and other applications specifying the dynamic behaviours of the domain entities. DIAS separates the "WHAT" from the "HOW". The object class only contains abstract descriptions of the various aspects of the object's behaviour (the WHAT), but no implementation details (the HOW). The models/applications implement all behaviours of objects (the HOW) and can be linked to appropriate domain objects "on the fly" to meet specific needs of a given problem domain. In DIAS new or legacy-type models or objects and application tools can be integrated through a Registration Process. This makes DIAS very flexible and extensible.

In DIAS, models/applications communicate only with domain (Entity) objects, never directly with each other. Thus, it is easy to add models, or swap alterna-

[3] http://www.dis.anl.gov/DIAS/

tive models in and out without re-coding. This allows DIAS to be able to scale very well to deal with complex problems.

DIAS's GUI system adopts a GeoViewer module to display outcomes spatially at user-selectable levels of resolution. Through GeoViewer, the data can be manipulated, queried, and analyzed. The results can also be visualized in a photorealistic manner.

2.4.5 Other simulation toolkits

Besides the above agent modelling frameworks, there exists a number of other agent toolkits, such as Ascape (Inchiosa and Parker 2002), NetLogo (Wilensky and Stroup 1999; Tisue and Wilensky 2004), MASON (Luke et al. 2004), AnyLogic [4] (Borshchev et al. 2002), etc. For reviews of the above agent frameworks and other agent-modelling toolkits, refer to the surveys done by Serenko and Detlor (2002), Gilbert and Bankes (2002) and Tobias and Hofmann (2004). These agent toolkits have been used extensively in various research fields from social science, to economics and defence applications. They largely increase the reliability and efficiency of the systems and reduce the development workload. However, they each have limitations. All these agent toolkits require the modellers to have a good working knowledge of certain programming language, e.g. JAVA for RePast (Gilbert and Bankes 2002). It is hard to clearly and fully understand the built-in assumptions and limitations of the modelling options, and even harder to find ways to create models based on them (Gilbert and Bankes 2002; Tobias and Hofmann 2004). Finally the capabilities of agents are embedded and based on the design of these agent toolkits. It is very difficult to extend agents' capabilities. For example, both SWARM and RePast are constructed on the reactive agent architecture. It is almost impossible to conduct human-like reasoning as BDI agents do unless users add their own JAVA class to the program. This requires users not only to understand the structure of the agent framework, but also to dig inside the source code. However, this is sometimes

[4] http://www.xjtek.com/anylogic/

impossible because the agent toolkit is not open source, e.g. JACK.

2.5 Multi-agent combat models

2.5.1 Combat as a complex adaptive system

The limitation of conventional combat models as discussed in section 2.3 have attracted many researchers and defence analysts' attention. The nonlinearity of combat has recently been recognized (Beyerchen 1992; Ilachinski 1997; Beckerman 1999; Ilachinski 2000; Lauren 2000). Ilachinski (Ilachinski 1997) was among the first to argue that land combat is a CAS. He identified the following match between the characteristics of a CAS and land combat:

Nonlinear interaction: Combat forces consist of a significant number of components interacting with each others nonlinearly.

Hierarchical structure: By its nature, forces are usually organized in a command and control hierarchy. This command and control structure is a complex system in its own right (Cooper 1993).

Decentralized control: In operations, each combatant is an autonomous agent that acts reactively based on its sensor information within the overall objective or plan.

Self-organization: While local actions of a combatant may appear chaotic, when seen over time, long-range order emerges.

Nonequilibrium order: By its very nature, equilibrium is not a characteristic of military conflicts.

Adaptation: It is not possible for combat forces to succeed in their designated missions without being able to adapt to changes in the environment.

Collectivist dynamics: The hierarchical structure of forces dictates a command chain, where low-level combatants and high-level command structures continuously communicate and feed back their states and actions.

2.5.2 Emergence

The concept of Emergence is widely used in complex adaptive systems literature, especially in computer sciences and related fields (multi-agent systems, artificial intelligence...). Emergence is the process which creates new patterns or properties of the system from interactions among system constructive components which are guided by a set of simple rules. Emergence rises from the macro level of some patterns, structures and properties of a complex adaptive system that is not contained in the property of its parts. Interactions between parts of a dynamic system are the source of both complex dynamics and emergence. Emergence is usually led by two types of causal relations: intricate causal relations across different scales and feedback (Bar-Yam 2004; De Wolf and Holvoet 2005). For a behaviour to be termed emergent, it should arise from simple low-level rules. The emergent behaviour or properties are not a property of any single low-level entity, nor can they easily be predicted or deduced from behaviour in the lower-level entities. They are irreducible, unpredictable and unprecedented (Holland 1998; Odell 2002; Bar-Yam 2003; De Wolf and Holvoet 2005). At the same time the emergent behaviour feeds back to influence the behaviour of the individuals that produce it. For example swarm intelligence is produced by a swarm of ants. Each ant does not know the optimal path to reach the food. However the optimal path is generated by interaction among ants and then all ants will follow this optimal path.

Holland (1998) in his book "Emergence: from chaos to order" defined the concept of emergence as follows:

- The hallmark of emergence is a sense of "*much coming from little*";
- Emergence involves "*getting more out of a machine than you put in*";

- *"A small set of well-chosen building blocks, constrained by simple rules, can generate unbounded streams of complex patterns"*.

Holland applies the above idea to computers as an example. A computer is the medium where we can most effectively create models of real-world emergence. The computer program is fully reducible to the rules (instructions) that define it, so nothing remains hidden; yet the behaviours generated are not easily anticipated from an inspection of those rules. The program is capable of surprising its programmer. Emergence involves the creation of a program that has clearly identifiable rules, the application of which results in unanticipated behaviours. Further, he concludes:

> "A small number of rules or laws can generate systems of surprising complexity. The systems are animated-dynamic; they change over time. Though the laws are invariant, the things they govern change. The rules or laws generate the complexity, and the ever-changing flux of patterns that follows leads to perpetual novelty and emergence."

A number of other definitions of emergence from different perspectives can also be found in literature. Eight important characteristics of emergence are extensively addressed by them (Holland 1998; Goldstein 1999; Parunak and Brueckner 2001; Odell 2002; Parunak et al. 2002; Bar-Yam 2003; Parunak and Brueckner 2004; De Wolf and Holvoet 2005) as follows:

1. Micro-macro effect: which is the most important property and is extensively explained in literature. A micro-macro effect means that global properties, behaviours, structures, or patterns at a higher macro-level arises from the (inter)actions at the lower micro-level of the system;

2. Novelty: The global behaviour is novel and cannot be exhibited by a single individual at the micro-level. For example, the individuals at the micro-level have no explicit representation of the global behaviour.

3. Coherence: There exists a logical and consistent correlation among individuals which integrates the separate low level individuals into a high level unity;

4. Interacting parts: The system components need to interact with each other. Without interaction, the interesting high level behaviours will not arise;

5. Dynamical: The global behaviour emerges at a certain point as the system is evolving in time.

6. Decentralised control: There is no central control to direct the low level individual behaviours. The actions of individuals are controlled by a set of simple rules. The whole is not directly controllable.

7. Feedback loop: Normally there is a bidirectional influence between the high-level and the low-level. The emergent properties, behaviours or patterns at the high-level arise from the interaction among individuals at the low level. At same time, the emergent properties influence its individuals' behaviour. Higher level properties have causal effects on the lower level. For example, the emergent optimal path influences the movement of each ant.

8. Robustness and flexibility: Since there is no centralized control and no single individual can represent the whole system, the failure or replacement of a single entity will not cause a complete failure of the whole system.

A formal definition of emergence is given by Baas (1994). Given $S_i(i \in J)$ is a set of structures. Int is a set of interactions over S_i. And Obs is an observational mechanism. Let S^2 be a new higher-level structure, where $S^1_{i_1}, i_1 \in J_1, S^2 = R(S^1_{i_1}, Obs_1, Int_1)$, where R is the construction relationship. P is an emergent property of S^2 iff $P \in Obs^2(S^2)$, but $P \ni Obs^2(S^1_{i_1}) \bigvee i_1$. Therefore two types of emergence can be specified:

- Deducible, or computable, emergence: there is a deductive or computational process or theory D such that $P \in Obs^2(S62)$ can be determined by D from $(S^1_{i_1}, Obs^1, Int^1)$ such as thermodynamic systems;

- Observational emergence occurs when P cannot be deduced as in the previous case such as Gdels incompleteness theorem. For some formal systems there can be true statements that cannot be deduced from the system of existing (known) true statements, and the semantic compositionality in languages (i.e., where the semantics of a word can be computationally derived from the semantics of the symbols constituting it).

Other types of emergence can also be found in literature, for example “nominal”, “weak” and “strong” emergence (Bedau 1997; Bedau 2002), or “weak”, “ontological”, and “strong” emergence for (Gillett 2002b; Gillett 2002a).

In multi-agent systems, emergence is a key property of dynamic systems based on interacting autonomous agents. The knowledge of agents’ attributes and rules is not sufficient to predict the behavior of the whole system. Such a phenomenon results from a specific structure of interaction among agents. Therefore, a better knowledge of the generic properties of the interaction structures would make it easier to have better knowledge of the emergence process. From this point of view, to denote a phenomenon as “emergent” does not mean that it is impossible to be explained (Emmeche et al. 1997). Thanks should be given to social network theory, which characterizes the structure or the pattern of the relationships, structural or relational processes among social actors via a number of network measures (Wasserman and Faust 1994; Albert and Barabási 2002; Newman 2003; Dorogovtsev and Mendes 2002). In this thesis, a novel multi-agent architecture, NCMAA which is based purely on network theory, is proposed where emergent behaviours is interpreted by using social network analysis techniques. An new combat simulation system, WISDOM (version II) is developed based on NCMAA and used to demonstrate that emergence can be explained.

2.5.3 ABDs for combat

The view of combat as a CAS opened a recent stream of research to use agent-based simulations to gain insight into military operations. The field is usually known as ABD. ABD emphasizes the concept of embodiment (Brooks 1991a) of agents in the environment. It enables defence analysts to study emergent behaviours in warfare. Simulation is used to glean insight of the dynamics and behaviours that may emerge from the system; thus providing defence analysts with a useful tool for assisting them in making recommendations to decision makers.

MAS is a natural platform for studying CAS. The combatants are modelled as agents, usually with a set of pre-defined characteristics. These agents adapt, evolve and co-evolve with their environment (Schmitt 1997; Lauren 2000). By modelling an individual constituent of a CAS as an agent, we are able to simulate a real world system by an artificial world populated by interacting processes. It is particularly effective to represent real world systems which are composed of a number of nonlinear interacting parts that have a large space of complex decisions and/or behaviours to choose from such as those situations in combat (Ilachinski 2003).

A number of MAS designed specifically for combat has been developed in the literature. These include ISAAC (Ilachinski 1997; Ilachinski 2000) and EINSTein (Ilachinski 1999; Ilachinski 2003; Ilachinski 2004) from the US Marine Corps Combat Development Command, MANA (Lauren 2000; Lauren and Stephen 2002b; Galligan and Lauren 2003; Galligan 2004) from New Zealand's Defence Technology Agency, BactoWars (White 2004) from the Defence Science and Technology Organisation (DSTO), Australia, and CROCADILE (Barlow and Easton 2002; Barlow 2003) from UNSW at ADFA.

2.5.3.1 ISAAC

ISAAC (Ilachinski 1997; Ilachinski 2000) is a skeletal agent-based model of land combat from the US Marine Corps. The goal of ISAAC is to become a fully developed complex system for analyzing nonlinear dynamics in land combat by identifying, exploring, and possibly exploiting emergent collective patterns of behaviours on the battlefield.

ISAAC is working in DOS with no user interface at all. Two forces play against each other in a two-dimensional lattice. Each force has its own flag and attempts to capture the enemy's flag or destroy enemies. The basic element of ISAAC is ISAAC agents (ISAACA), each of which presents an entity on the battlefield, such as an infantryman or a tank. Agents have a set of attributes and personalities, based on which agents take actions. After a simulation, a set of data is automatically generated, based on what defence analysts may need to conduct the military analysis.

Characteristics of agents There are seven attributes for each agent, which guide it to sense, communicate, move and shoot.

- Sensor range (S-range): the maximum range at which an agent can sense other agents in its vicinity.
- Fire range (F-range): the maximum range at which an agent can fire upon an enemy agent.
- Movement range (M-range): the maximum number of grid squares an agent can move in any single time step.
- Communication range (C-range): the maximum distance over which an agent can communicate (share information) with other friendly agents.
- Threshold range (T-range): the maximum range within which the number of friendly agents is above a user-defined threshold, the meta-personality will be activated.
- Probability of Hit (p-shot): the probability that an agent will hit an enemy agent that is within the firing range.

- Maximum targets (MAX TGT): the maximum number of targets that an agent can engage in any single time step.

Personality of agents The personality of an agent presents the tendency to move close to or away from particular agents (friends or enemies). With some certain penalty function, these personalities govern the agents movements. In ISAAC, there are six basic personality weights:

- Weight towards alive red (AR);
- Weight towards injured red (IR);
- Weight towards alive blue (AB);
- Weight towards injured blue (IB);
- Weight towards red goal (RG);
- Weight towards blue goal (BG).

The positive value of these weights means that the agent tends to move close to that type of agents while the negative value means that the agent tends to move away from that type of agents. These personalities can be dynamically changed depending on its situation awareness and following meta-personalities during the course of the simulation.

- Advance meta-personality (ADV): the threshold number of friendly agents within an agent's threshold range in order for that agent to advance toward the enemy flag. That is, the agent won't advance to the goal unless there is a certain level of support. This is achieved by negating the personality weight of moving towards the enemy flag, whenever the threshold is exceeded.
- Cluster meta-personality (CLS): the threshold number of surrounding friendly agents in order for that agent not to move close to its friend. This is achieved by negating the personality weights to friends whenever the threshold is exceeded.

- Combat meta-personality (CBT): the threshold number of surrounding friendly forces over enemy forces in order for that agent to move away from the enemy. This is achieved by negating the weight to enemies whenever the threshold is not exceeded.

Terrain feature The only form of terrain supported in ISAAC is impassable objects. Agents cannot see, shoot or travel through an impassable terrain. Terrain blocks can be defined at the end of the input file by specifying each block's coordinates and the corresponding length and width.

C2 structure A three-level C2 structure is modelled in ISAAC. The levels are:

1. Elementary combatants: which are defined by basic parameters as described above and two additional weights: the propensity to stay close to their commander and the propensity to obey commands. Each elementary combatant should be assigned to a local commander.

2. Local commanders: which coordinate information flow among local groups of elementary combatants. Each local commander determines local goals within its commander area and can order its subordinates to move towards these goals. The movement of a local commander is based on its own personalities along with a propensity to help other local commanders and a propensity to obey orders issued by the global commander.

3. Global commander: which has a global view of the battlefield and coordinates the actions of local commanders.

Movement algorithm For each time step, an agent may move to any of the cells within its movement range or remain still. A penalty is calculated for each of the potential new locations based on its situation awareness and personalities. The cell with the lowest penalty is chosen as the new location. The penalty function is as follows (Ilachinski 1997):

$$Z_{new} = Z_{agent} + Z_{flag} \tag{2.5}$$

$$Z_{agent} = (\frac{W_E}{E * R_s\sqrt{2}} \sum_{i=1}^{E} DE_{i,\ new}) + (\frac{W_A}{A * R_s\sqrt{2}} \sum_{i=1}^{A} DA_{i,\ new}) \tag{2.6}$$

$$Z_{flag} = W_{EF}(\frac{DE_{F,\ new}}{DE_{F,\ old}}) + W_{OF}(\frac{DO_{F,\ new}}{DO_{F,\ old}}) \tag{2.7}$$

where:

R_s: Sensor range of an agent that is deciding to move;

E: Number of enemy entities within sensor range;

A: Number of friendly entities within sensor range;

W_E: Weighting towards enemy agents;

W_A: Weighting towards friendly agents;

$DE_{i,\ new}$: Distance to the ith enemy from the new location;

$DA_{i,\ new}$: Distance to the ith friend from the new location;

W_{EF}: Weighting towards the enemy flag;

W_{OF}: Weighting towards own flag;

$DE_{F,\ new}$: Distance to the enemy flag from the new location;

$DE_{F,\ old}$: Distance to the enemy flag from the current (old) location;

$DO_{F,\ new}$: Distance to own flag from the new location;

$DO_{F,\ old}$: Distance to own flag from the current (old) location.

If communication is taken into account, the penalty function will be:

$$Z_{new} = Z_{s,\ new} + W_c Z_{c,\ new} \tag{2.8}$$

where $Z_{s,\ new}$ means penalty from sensor; W_c means communication weight; $Z_{c,\ new}$ means penalty from communication.

Run mode There are three run modes in ISAAC: interactive run mode, multiple time series run mode and genetic algorithm (GA) run mode. In the interactive mode, the user can interact with the simulation and change the value of all parameters on the fly. It allows the user to see the combat dynamics, what is exactly happening, and how a particular side is winning.

In the multiple time series or data-collection mode, multiple trials of the same combat with different initial configurations of combatants are simulated. The system then automatically generates statistical data files for each single run. The information generated includes:

1. Force strengths - such as the number of alive red, alive blue, injured red, injured blue, total red and total blue forces;

2. Interpoint distance - averages and distributions of the distances between pairs of agents or between an agent and the flag;

3. Number of neighbour agents - averages and distributions of the number of neighbours, such as red, blue and all (either red or blue) agents near red agents, red, blue and all (either red or blue) agents near blue agents, and red and blue agents near both red and blue flags;

4. Interpoint distance of enemy flag - averages and distributions of the distances between red or blue agents and their enemy flags, such as between red agents and the blue flag, and blue agents and the red flag;

5. the size of cluster - averages and distributions of the sizes of clusters of agents;

6. Centre-of-mass positions - keeping track of the coordinates of the centre-of-mass position of the red, blue and total force;

7. Spatial entropy - the spatial entropy of the red, blue and total force. Spatial entropy is intended to measure the degree of disorder of a battlefield state.

The last mode is the GA or "Evolver" mode, which is used to search for optimal solutions for a predefined scenario. In this mode, each force can only have one group with homogeneous personalities. While fixing the personality of the red force, the personality of the blue force is evolved to search for the optimal personality.

2.5.3.2 EINSTein

EINSTein (Ilachinski 1999; Ilachinski 2003; Ilachinski 2004) builds on and extends the DOS-based combat simulator ISAAC from the US Marines Corps. EINSTein provides a user friendly GUI for the system. This makes it easier for a user to set up scenarios, view what is happening during the simulations, and thus lets the user gains a better understanding of embedded dynamics of a combat scenario. The main features of EINSTein include (Ilachinski 2003):

- A windows GUI front-end;
- Integrated natural terrain maps and a terrain dependent decision making mechanism;
- Context-based and user-defined agent behavioural rules;
- Multiple squads for each force;
- Squad based communication;
- Three level C2 structure: global commander, local commander and basic combatant;

- GA toolkit to search for optimal solutions for a certain scenario;
- Data collection and farming functionality;
- Multiple multi-dimensional visualization tools.

Agent Model Like in ISAAC, an agent in EINSTein represents a primitive combat unit, such as an infantryman, a tank, a transport vehicle, etc. Each agent has the following characteristics:

- Doctrine: a set of local rules, which are used to guide the agent's behaviour;
- Mission: each agent has a user-defined goal;
- Situational Awareness: sensors collect information and generate a representation of the agent's local environment;
- Adaptability: a meta-personality based mechanism to alter behaviour and/or rules.

Agents can be in one of three states: alive, injured or killed. Agents in different states may have different characteristics or personalities predefined by users. For example, an aggressive agent may become defensive when it is injured. Each force may have several squads with different sizes and different characteristics. However, no more than ten squads can be defined for a single scenario. The communication is squad based. This means that communication only occurs between squads, not between agents. Agents in EINSTein are also associated with a set of ranges, e.g. C-range, S-range and F-range. The agents collect information from their local environment and act upon it.

EINSTein adopts the same movement algorithm (penalty function) as that in ISAAC to guide agents' movement. Each agent has the same types of personality and meta-personality as those in ISAAC. In addition, several new meta-personalities have been introduced in EINSTein:

- hold - hold current position if more than a threshold number of friendly agents are within the user-specified threshold range;
- Pursuit I - temporarily ignore enemy agents if fewer than a threshold number of enemy agents are nearby;
- Pursuit II - temporarily ignore friendly agents and its own flag, and try to move towards and attack enemy agents if fewer than a threshold number of enemy agents are nearby;
- Retreat - retreat back to its own flag if no more than a threshold number of friendly agents are surrounding its own flag;
- Support I - temporarily ignore all other personalities and try to move towards injured friendly agents to provide help if more than a threshold number of injured friendly agents are nearby;
- Support II - temporarily ignore all other personalities and try to move towards alive friendly agents to seek their help if more than a threshold number of enemy agents are nearby.

Terrain feature Compared with ISAAC, the terrain features of EINSTein are improved. Two types of terrains are defined by the system: passable and impassable. As well, the system allows users to create up to three of their own types of terrain. When creating a new terrain, the user needs to define the effect of the terrain on the characteristics of agents when they are on the terrain block. The characteristics may include sensor, fire, threshold, movement and communication ranges, defence strength, probability of hit, etc. This feature largely enhances the adaptability of EINSTein. In this way, the user may easily simulate a number of terrain types in the real world, such as roads, woods, rivers, etc.

Run mode Five modes have been implemented in EINSTein (Ilachinski 1999):

- Interactive Run Mode - This mode is similar to that in ISAAC.
- Play Back Run Mode - the user may record the simulation and play back at a higher speed.
- Multiple Time-Series Run Mode - the simulation can be run multiple times with different starting configurations and data collected for analysis.
- 2-Parameter Fitness Landscape Mode - Two parameters are selected to vary. Based on the outputs of multiple runs, a 3D mission fitness landscape is generated.
- One Sided Genetic Algorithm Run Mode - with this mode the user may search for an optimal force to perform a user-defined "mission" against a fixed opponent. It is similar to ISAAC but it is much easier to set up the mission and the GA within a window interface.

Two additional features Two features have been introduced in EINSTein: Inter-Squad Matrix and weapons parameters. Inter-Squad Matrix defines the relationship between two squads within one force. This feature allows the user to model cooperation or coordination between squads. EINSTein also introduces the weapon of grenade by specifying a minimum and maximum throwing range and the accuracy of the grenade is a function of the distance.

2.5.3.3 MANA

MANA is inspired by ISAAC and EINSTein and was developed by New Zealand's Defence Technology Agency using the same underlying agent paradigm and design (Lauren and Stephen 2002b). The MANA model is designed to study some important real-world factors of combat such as (Galligan et al. 2003):

- Change of plans while the battle is evolving;

- The effect of situational awareness on agent behaviour;
- The importance of acquisition information.

Like EINSTein, MANA has a user interface which allows a user to set up and run simulations quite easily. While MANA has similar parameters as ISAAC and EINSTein, several new concepts or models are introduced: a situation awareness (SA) map, a communication model, a terrain map and way-points.

Squad MANA is a squad based combat simulator. Each side may be made up of several squads, each of which consists of a number of homogeneous agents; that is all agents within a single squad have the same properties (behavioural and capability parameters), same SA map of enemy contacts and same way-points. However, when certain event happens, the state of agents can be changed individually or as a group.

Situation awareness Each squad has its own "memory" of the location of agents perceived by the squad, which is shared by all agents in this squad. Two types of SA maps are implemented in MANA: the squad SA map, which holds the information directly collected by their sensors, and the inorganic SA map, which holds the information received from by other squads through communication.

Communication model The latest version of MANA (version 3.0.37) has a comprehensive squad based communication model. Since all squad members share the same SA, the information can only be exchanged between two squads. The information received through communication links is stored in the inorganic SA map. MANA adopts a first-in-first-out queuing algorithm to send off messages. Each communication link is modelled by several parameters, as follows:

- Range: the maximum distance between the centroid of two squads;
- Capacity: the maximum number of messages that can be sent out per time step;

- Buffer: the maximum queue size. Once the queue is full, the oldest messages will be removed to make room for new messages;
- Latency: the number of time steps taken for each message to be received by a squad;
- Reliability: the probability that a message can be successfully received by a squad per try;
- Accuracy: the probability that a contact is passed as "unknown".
- Max Age: the maximum time steps the message remains in the queue;
- Trust: the level of confidence that the sender has that the receiver will receive the information;
- Include: the contact types can be sent out;
- Delivery: two types of delivery are supported in MANA - Guaranteed Delivery, which queues messages when out of communication range, and Fire-N-Forget, which ignores messages when out of communication range.

Terrain feature The default battlefield in MANA is a 200 x 200 grid of cells, each of which can be occupied by a single live agent. Several terrain types are modelled in MANA:

- Billiard Table: the plain terrain;
- Easy Going: representing a road or other region that is designed for moving;
- Wall: representing obstacles. No agent may occupy a wall (obstacle) cell. Agents can see through wall only when the option of "Line of Sight" is turned off;
- Light Bush/Dense Bush: Differing density provides different impact on the movement of the agents. It can also be used for concealment;

- Hilltop: modelling a high level of concealment.

MANA can load any standard Windows bitmap file as a terrain map. The terrain types are differentiated visually by colours.

Way-point In MANA, the user may define a number of way-points to guide the agents to reach their ultimate goal. The way-points are predefined and can be changed during a simulation. An agent's personality settings can be used to attract or repel it from way-points.

Event-driven personality changes This feature allows agents to change their personalities when a certain event occurs. These events may include being shot at, taking a shot, reaching a way-point, making enemy contact, etc. Personality changes can occur individually or as a whole squad. The changes may last for only a certain time, after which the personality will be recovered to the previous setting.

Fuel Fuel is a particular concept introduced in MANA and is not modelled for any particular purpose. The aim of this parameter is to give some degree of freedom to the user and allow the user to be able to model any quantifiable concepts. For example, it can be used to represent concepts such as courage or fatigue, the number of interactions with other agents, the logistics of supply of some commodity, etc. Each agent has a fuel tank with an initial allocation of fuel. The amount of fuel in the tank can increase, decrease or remain static during the simulation. An agent can be refuelled by any other agent. The process of refuelling can be used to model the interaction between agents or used as a trigger to change the agent's personality.

Movement algorithm The movement algorithm in MANA consists of the following steps:

- Consider all possible moves within the movement range, including the current location;
- Discard moves entering into cells which contains other agents or terrain features;
- Consider all the agents in the Situational Awareness map, the way-points and terrain based on the agent personality;
- Consider all constraints, such as minimum distance to others, cluster constraints, and so on;
- Select the location with lowest penalty as the new location. If a number of moves are nearly equal, then randomly choose a move from the set.

Like ISAAC and EINSTein, MANA also adopts an attraction-repulsion weighting system to determine the penalty. The penalty for moving to any grid location is the sum of the 27 penalty calculations. For the details of the 27 weighting factors in MANA, refer to Galligan et al. (2003). The algorithm used to calculate the penalty is presented as in Equation 2.9 (Galligan et al. 2003):

$$Penalty = 1 + \frac{\sum_{m=1}^{M}(D_N(m) - D_O(m))D_W(m)}{100\sum_{i=1}^{M} D_W(i)} \tag{2.9}$$

where M is the number of agents within the distance used in the weighting equation; D_N and D_O are the new and old distance from the agent to each entity respectively; and D_W is a weighting factor determined by Equation 2.10:

$$D_W = Round(BDL - D_O) \tag{2.10}$$

where BDL is the length of the main diagonal of the battlefield, which is used to remove the effect of the scale of each scenario when calculating the penalty. D_W is used to increase the influence of closer entities.

Run mode There are only two run modes in MANA: an interactive run mode and the multiple time series run mode. These two modes are similar to those in EINSTein. When running in the latter mode, the information stored in the output file includes: the version of MANA, the start time, then for each run the seed number used, the number of casualties for each side, whether or not one side reaches its final goal, the total number of time steps, the casualties for each squad, and the end time. The average casualties and time step are also given at the end of the results.

2.5.3.4 CROCADILE

CROCADILE (Barlow and Easton 2002; Barlow 2003) is a multi-agent-based combat distillation, which is designed to improve the limitation on generality and fidelity in ISAAC, EINSTein and MANA. The key features in CROCADILE are as follows:

- The environment: CROCADILE implements a 3D environment where the agents interact;
- Probabilistic or Projectile-Physics combat resolution: CROCADILE not only supports the traditional probabilistic model for hit resolution, but also incorporates a projectile-physics model that includes factors such as target size, speed, and distance away and the terrain itself;
- Movement by land, air and water: the agents can move on the land, in the air or sea;
- User extensible agent behaviours: there is an interface in CROCADILE which allows the user to define characteristics for each agent;
- Sophisticated Command, Mission, and Communication structures: the hierarchies of command and communication can be established between groups of agents. Missions may include destruction of enemy agents, reaching a goal or destroying a static feature;

- Higher fidelity combat resolution: this is achieved by incorporating blast effects, round penetration, rates of fire, and line-of-sight;
- Database of world objects: terrain, agents, agent groups, agent behaviours, weapons, movement capabilities, sensors, command structures, and communication structures can be saved in XML format individually and reused in subsequent scenario building;
- Comprehensive simulation event logging: the results of each run are output to a set of log files which include information about the state of the scenario at each time frame. With these log files, the user may visualize and analyse the data;
- Multi-team structure: the agents in CROCADILE are organized in teams. The relationship between teams may be friend, neutral or enemy.

Capability of agents Five types of capabilities are modelled for each agent in CROCADILE: firepower, mobility, sensing, communication and command.

Firepower is implemented by defining different types of weapons possessed by agents. The weapon can be direct or indirect weapon and can fire kinetic or explosive rounds. Each weapon is modelled by its maximum range, rate of fire, damage, penetration, number of rounds, calibre, muzzle velocity and the blast radius of its ammunition if it fires explosive rounds.

The capability of sensing is defined by three parameters: the maximum scan range, the level of detailed threshold range and the maximum scan angle.

The movement capability allows agents to move over ground, water or air. There is a maximum speed for each medium.

The communication capability allows agents to broadcast to all other agents within its communication range. Only broadcast communication is supported in CROCADILE.

The command capability allows agents to send orders to their subordinates. Three types of missions are supported: attack mission, avoid mission and advance mission.

Trigger CROCADILE attempts to separate agents' behaviours from their capabilities. The actions an agent takes are triggered by activating the proper triggers. There are six personality triggers implemented within the instinctual agent control paradigm.

1. Hit trigger: it is activated when an agent is hit by a munition. That means the behaviour template used by this agent will be changed for a fixed number of time steps. After that, the behaviour template of this agent will revert to its previous one;

2. Health trigger: When it is activated, the agent health will be updated;

3. Force-ratio trigger: it is fired when the force ratio is more extreme than the value specified by the user;

4. Mission trigger: it contains an array of mission references and a corresponding array of behaviours. When an agent completes a mission, the corresponding behaviour will be executed;

5. Time trigger: similar to the mission trigger, it consists of an array of times and associated behaviours. If the current time matches one in the time array, the relevant behaviour is assigned as the current behaviour;

6. Command trigger: it consists of a set of commands and a set of corresponding behaviours. When receiving a command, the associated behaviour is assigned as the current behaviour for the agent.

Terrain feature A 3D environment is implemented and a digital terrain map from the real-world can be integrated into the system. This is the key improvement when compared with other existing ABDs. All physical objects such as agents and munitions are located in a 3D space. Three types of terrains are supported: land, air and water, each of which may have a shape and location predefined. This kind of 3D landscape may affect movement, line-of-sight issues such as sensor detection, and hit resolution - the flight of projectiles and blast effects.

Movement algorithm CROCADILE also adopts an attraction-repulsion weighting system to guide the agent's movement. There are nine major weighting factors.

1. Enemy: specifies how an agent will position itself with respect to enemy agents.
2. Neutral: specifies how an agent will position itself with respect to neutral agents.
3. Friend: specifies how an agent will position itself with respect to friendly agents.
4. Commander: specifies how an agent will position itself with respect to its commanders.
5. Mission: specifies how an agent will allow its missions to affect its movement.
6. Feature: specifies how an agent will position itself with respect to environment features, such as water, land or obstacles.
7. Terrain: specifies how an agent will use the terrain to affect its movement.
8. Message: specifies how an agent reacts to the message it receives.
9. Exploration: specifies how willing an agent is to explore the environment.

For each weighting factor, the user may specify an attenuation function (Equation 2.11 and 2.12) governing how each weight varies as a function of distance. Either equation 2.11 or 2.12 can be used:

$$W = w_0 \times r^{\frac{d}{D}} \tag{2.11}$$

$$W = w_0 \times r^{\frac{D-d}{D}} \tag{2.12}$$

where W is final weight after attenuating, w_0 is the initial weight without attenuating, r is the minimum ratio of the full strength of a weight that a weight can attenuate to, D is the distance over which the weight attenuates to that level and d is the current distance between the agent and the object that it is generating the weight with. When increasing the distance, the weight may decrease by using equation 2.11 while the weight may increase by using equation 2.12.

After attenuating all weights, the sum of them is calculated. The agent will choose the direction with the highest weight. Based on the moving speed, the new location of the agent is determined.

Run mode Similar to ISAAC, EINSTein and MANA, both interactive run mode and multiple time series run mode are supported in CROCADILE.

2.5.3.5 BactoWars

Bactowars (Grieger 2002; White 2004) is an ABD being developed at Land Operations Division, DSTO, Australia. BactoWars focuses on problem representation and attempts to provide a simple framework which allows analysts to model real world problems more adaptively and flexibly.

BactoWars adopts modern artificial intelligence techniques, such as semantic networks and frame theory, and software engineering techniques, e.g. the strategy design pattern, to allow the modeller to reuse the pre-developed agents and contextually build agents for specific problems by dynamically recombining behaviours for the agents, and to allow for adaptation at the individual level by dynamically choosing behaviours based on their perception of their environment. Typically there are four components in BactoWars (White 2004):

- Lexical components: the building block of the system, which include agents, behaviours, triggers, parameters, BactoMaps, scenarios and simulations. One can create these building blocks to represent the real world entities.

- Structure components: the way to hierarchically organize these building blocks. In BactoWars, simulations consist of scenarios which contain BactoMaps and agents with a set of behaviours. Simulations, scenarios and agents are typically defined by a set of parameters, which are specified by triggers, variance, logged values and an understanding of context.

- Procedural components: the controller to control the simulation process and the order of behaviours.

- Semantic components: the understanding of how the things are represented and what they mean.

There are two types of agents in Bactowars (Grieger 2002), one is a physical agent and the other is called a marker (or meme) agent. The meme represents how the physical agents interact with each other. The physical agents are actuators in the simulation. They have the same general properties as those in ISAAC, such as sensor range, firing range, movement range, etc. BactoWars also has stealth and fuel parameters similar to those in MANA. The qualitative analysis via interactive simulation is the only function supported in Bactowars for data collection and analysis (Grieger 2002). BactoWars can use a bitmap file as terrain map and the terrain type can be defined with different colours. BactoWar is written in JAVA and suffers from the same problems as other JAVA program with regard to slow run times (White 2004).

In summary, BactoWar attempts to provide a simple tool which may be used to represent real world adaptive systems, which may or may not be complex. The crucial drawback is how to develop customized algorithms to properly represent the research problem (Grieger 2002).

2.5.4 Comparison of ABDs

These agent based combat systems have been widely used by defence analysts and facilitated military analysis (Brown 2000; Lauren and Stephen 2000; Grieger and Gill 2001; Lauren 2001c; Lauren 2001a; Lauren 2001b; Cioppa 2002; Epstein 2002; Gill et al. 2002; Lauren 2002; Wheeler 2005; Perry 2006). They offer an opportunity to exhibit the behaviours that we would intuitively expect on the battlefield. Through the use of these systems defence analysts are able to gain understanding of the overall shape of a battle and what factors are more important than others in determining the outcome of a battle. They all build on the same understanding that combat can be modelled as a CAS and global behaviours emerge from a set of local rules. However these systems have different features. Table 2.2 summarises the features of these ABDs discussed above.

Table 2.2: Comparison of ABDs

	ISAAC	**EINSTein**	**MANA**	**CROCADILE**	**BactoWars**
Run Time	Fast	Fast	Fast	Medium	Slow
Programming language	C++	C++	Delphi	JAVA	JAVA
Ease to setup	Slow, no user interface	Fast, user friendly GUI			
Terrain	Impassable objects only	Passable, impassable, and up to three user defined terrains	Billiard table, easy going, wall, bush and hilltop	Land, water and air	User defined terrains
Multiple time series run mode	No	Yes	Yes	Yes	Yes
Interactive simulation	Yes	Yes	Yes	No	Unknown
Embedded visualization tool	No	Yes	Yes	No	Unknown
Automated parameter space exploration	Yes, via MHPCC	Yes	Limited	No	No
Personality changes	Yes	Yes	Multiple triggers	Yes	No
C2 structure	three tier	three tier	No C2 structure in version 3.0.37	User defined	No
Situation awareness map	No	No	Yes	No	No
Communication	Individual based	Squad based	Squad based, with extensive parameters	Broadcasting	unknown
Way-point	Only has ultimate goal	Only has Ultimate goal	a set of user defined way-points	Only has Ultimate goal	Only has Ultimate goal
Movement algorithm	Attraction/Repulsion weighting system				
Agent architecture	Reactive agent architecture				

2.6 Summary

In this chapter, a comprehensive literature review is undertaken in modelling and simulation of combat. Historically, conventional combat models or simulations, such as ELAN, JANUS, CASTFORME, ModSAF and OneSAF were used in military analysis, However, many aspects, e.g. human emotions and politics, have not been modelled in conventional combat models or simulations. Recently the idea that combat can be modelled as a CAS has widely been accepted in military analysis based on the matched characteristics between combat and CAS.

MAS is a natural platform to study CAS. Typically there are two agent architectures: cognitive agents and reactive agents. Cognitive agents conduct human like reasoning in order to achieve their goals. They are intentional, rational and flexible. However, it is quite hard to represent a real world problem in a symbolic format in order to conduct reasoning. Reactive agents respond to their environment purely based on redefined simple rules. They are simple, economic, and robust, and can scale up well. However, without reasoning and planning, reactive agents may exhibit unexpected and weird behaviours and it is very hard to connect the local rules to the emerging behaviours. To overcome the limitations of both architectures, a hybrid architecture is developed by simple mixing cognitive and reactive agents within a single system. However connecting the cognitive agent layer with the reactive agent layer is also not a simple task.

In order to facilitate the developing process of MAS, a number of multi-agent simulation toolkits are developed, such as SWARM, RePast, JACK, DIAS and so forth. They largely increase the reliability and efficiency of the systems and reduce the development workload. However, their limitations are also very obvious. For example, all these agent toolkits require modellers to have an essential good working knowledge of certain programming languages, e.g. JAVA for RePast. It is hard to clearly and fully understand the built-in assumption and limitations of modelling options, and even harder to find ways to create models based on them.

Mainly based on the reactive agent architecture, several ABDs are proposed for defence operations, which includes ISAAC, EINSTein, MANA, CROCADILE and BactoWars. ISAAC and EINSTein created a new era for warfare analysis. They were the first two systems which modelled warfare as a CAS. Almost all later ABDs were inspired by them. MANA first introduced the concept of way-points and internal situational awareness (SA) map. These new features largely improve the adaptability of the agents to a changing battlefield. The version released at the end of 2004 concentrated on the model of network centric communication, including different parameters of a communication network, such as reliability, accuracy, capacity and latency. BactoWars focused on problem representation and attempted to provide a simple framework which allows analysts to model real world problems more adaptively and flexibly. CROCADILE was the first system to use a 3D continuous environment with a higher fidelity than that of ISAAC, EINSTein and MANA. The detailed differences among these systems are listed in Table 2.2.

In the next chapter, a new agent-based combat simulation system, WISDOM - A Warfare Intelligent System for Dynamic Optimization of Missions - is developed and used to establish an understanding of combat as a CAS and to help us to identify theoretical causes of the drawbacks in existing ABDs.

Chapter 3

WISDOM - A Warfare Intelligent System for Dynamic Optimization of Missions [1]

3.1 Introduction

In this chapter, version I of a warfare intelligent system for dynamic optimization of missions (WISDOM-I) is proposed to study the theory of CAS and MAS, and how to apply them in military analysis. WISDOM-I is inspired by existing multi-agent combat models, such as ISAAC, EINSTein, MANA and CROCADILE. Similar to these systems, WISDOM-I employs a low-resolution abstract model in which the detailed physics of combat are ignored while only essential characteristics of combatants, defence operations or behaviours are modelled. Combatants in WISDOM-I are modelled as agents defined by a set of characteristics. These agents inhabit a two-dimensional discrete space. Their behaviours are guided by a set of simple rules of local interaction with other agents. The status of agents vary as agents evolve over time. WISDOM-I can help defence analysts to explore the decision space, study

[1] This chapter is based on the publications of Yang et al. (2005d) and Yang et al. (2004a).

how the behaviours of individual combatants influence the outcome of the battle, capture the common patterns of a combat, etc.

The main features of WISDOM-I include:

- Development in the object-oriented programming language Java, which makes WISDOM-I platform independent;
- A GUI allowing users to create scenarios and run simulations quite easily;
- 2D environment using a natural terrain map and terrain features differentiated by colours;
- Scenario dependent and user defined rules of agent interactions;
- Adopting an agent based communication and attraction-repulsion movement algorithm;
- Tactical decisions made based on agent situation awareness and agent personalities;
- Heterogeneous multi-group structure;
- Embedded evolutionary computation toolkit to evolve agent local rules for desired group-level behaviour;
- Using a relational database - MySql - to store data and facilitate post-analysis.

3.2 System design

The agent in WISDOM-I is living in a territory which consists of physical and social environments. With the system evolving, the status of a territory keeps changing. The physical environment includes road, water, tree, mountain, etc. The social environment includes surrounding agents no matter of whether they are in the same team. The status of the physical environment is normally fixed while that of the social environment is dynamically changing during the simulation.

3.2.1 The agent model

Each agent has five characteristics: perception, capability, execution, reasoning and a decision making mechanism. It also includes some other attributes (see Figure 3.1).

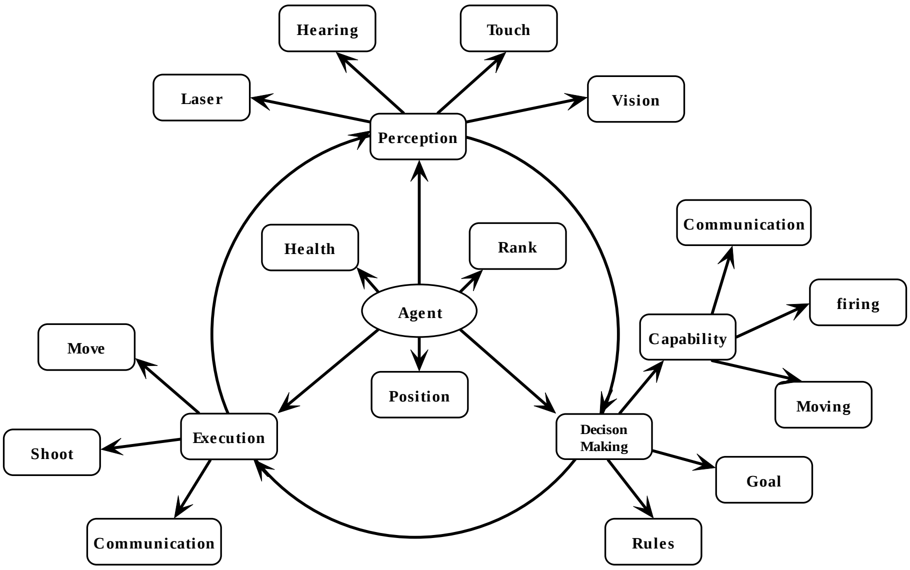
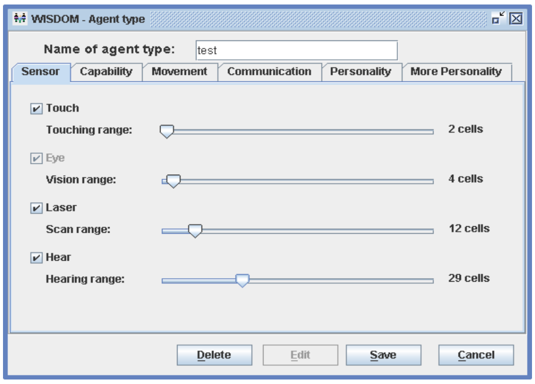


Figure 3.1: The agent design framework

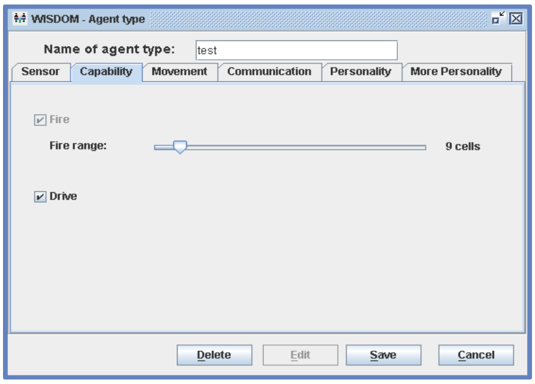
- Perception: the internal representation of the agent's environment through its sensors. The sensors may include touch, hearing, vision, laser scanner, etc (see Figure 3.2(a)).
- Capability: what the agent is able to do. It may include the ability of speaking, moving, firing or communicating (see Figure 3.2(b), 3.2(c) and 3.2(d)).
- Decision making mechanism: the way to determine which action the agent should take under certain circumstances. The decision should be based on its objective, perception, capability and a set of predefined rules.

- Execution: the agent executes the selected action. The agent may shoot its enemy, move to a new place, exchange messages with its friends, buy or sell goods, etc. Four modes of mobility are available: leg walking, wheel movements, swimming, and flying (see Figure3.2(c)). They are differentiated by both the speed or the terrain type the agent can go through.

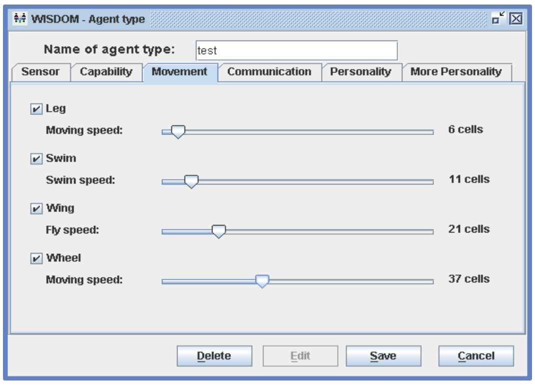
- Attribute: an agent may have attributes such as how good the agent is, what its physical position is, what its social position is, etc.



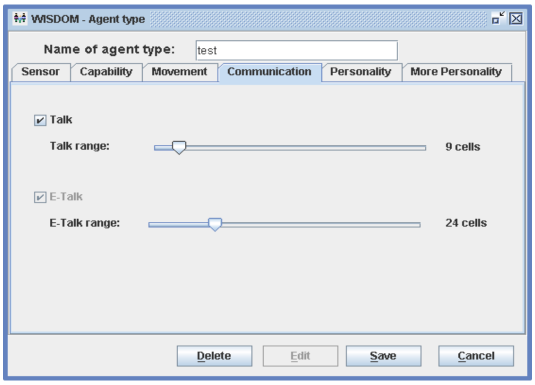
(a) Sensor



(b) Capability



(c) Movement



(d) Communication

Figure 3.2: Screen dump for agent characteristics

At each simulation step, agents build their perceptions of their environment through their sensors and then make decisions based on their perception, goal and capability. Finally they take actions to influence the system. The changed status of the system will in turn lead to new perceptions of agents in the next simulation step.

3.2.2 Agent interaction

The interaction between agents is embedded in the agent model. Agents may communicate with each other, fire at their enemy or move close to or far from other agents. After interacting with other agents, the status of all interacting agents may be changed.

3.2.3 Decision making mechanism

A rule based decision making mechanism is employed in WISDOM-I. The user may specify the rules for each type of agents to make decisions. Regarding the decision of movement, an attraction-repulsion weighting algorithm is adopted. This type of algorithm is widely used in existing combat ABD models, e.g. ISAAC, EINSTein and MANA. The basic idea is that each agent is assigned a set of personalities, which define the preference of moving close to or far from certain type of agents or its goal. At every simulation time step, the penalty or the weight of each possible place, as a function of the agent's personality, is calculated based on its situation awareness. The agent always moves to the place with the lowest penalty or the highest weight.

3.2.4 Data storage, collection and analysis

As an exploration tool, WISDOM-I should be able to run multiple times for defence analysts to investigate the problem space and then help them to make decisions. It then generates a large amount of data for analysis. Storing the data into a database (see Figure 3.3) can largely facilitate the data analysis process.

3.2.5 Terrain feature

The terrain feature largely influences the outcome of combat. It not only affects the movement of agents, but also influences their perception and firing activities.

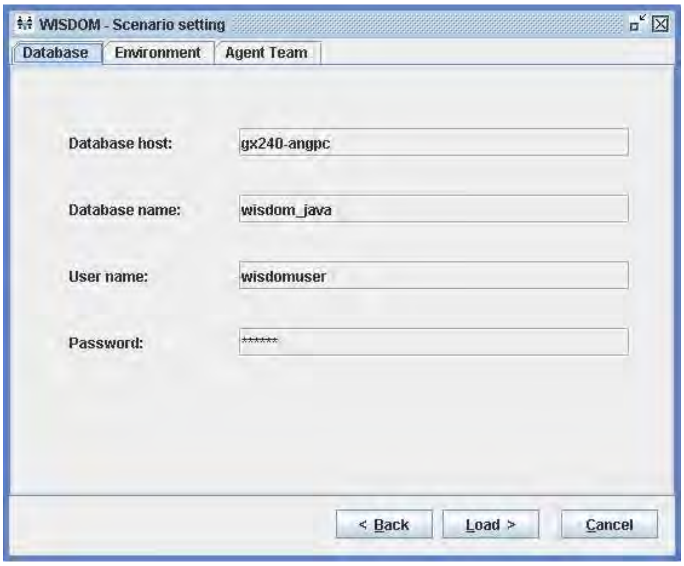


Figure 3.3: Database configuration

WISDOM-I may integrate with a natural terrain map and use colour to differentiate the terrain types. The terrain types may include urban land, water, pasture and rough terrain (see Figure 3.4).

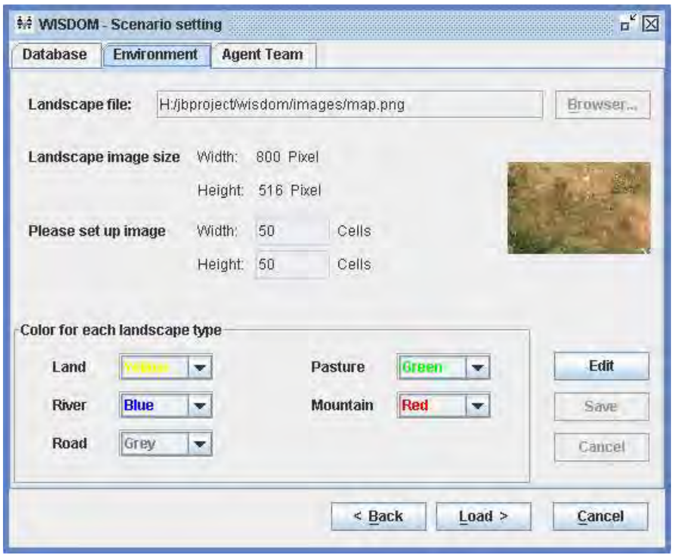


Figure 3.4: Terrain features

3.3 System implementation

In order to be platform independent, the pure object-oriented programming language Java is used to implement the system of WISDOM-I. The interface and simulation engine are completely separated. It makes WISDOM-I highly extendable, flexible and reusable.

3.3.1 Agent characteristics

An agent in WISDOM-I is characterized by: sensors, capabilities, movements, communications, health and rank. Three types of sensors are programmed with their associated ranges; these are: touch, vision and laser scanner sensors. Since WISDOM-I is a low-resolution model, these sensors are distinguished by its range. The range of touch is short while the range of laser scanner is long. The range of vision is in between.

The capabilities implemented in WISDOM-I include firing and mobility. Only a direct fire weapon is supported in WISDOM-I. The "Probability to hit" is a measure of the agent's firing skills.

Agents can communicate with their friends when they are within the communication range of each other. The structure of the communication network (who is allowed to communicate to whom, how, when, and where) is defined using a graph.

The health parameter defines the level of energy for an agent. Initially all agents start with a maximum level of energy defined by the user for different agent types. When an agent gets fired, the degradation of its health depends on the strength of the firing weapon. Based on the user predefined threshold, the level of health determines the status of agents: healthy, injured or dead.

The rank defines the position of the agent in the C2 hierarchy.

3.3.2 Decision making mechanism

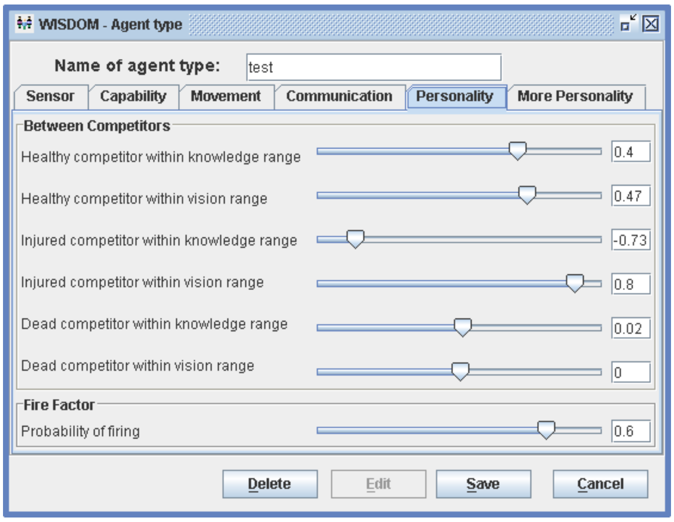
At each simulation time step, an agent can move, fire, and/or communicate with other agents. Seven simple rules are employed to make decisions. They are:

1. If alive, the agent scans its environment;
2. If there are friendly agents within the agent's communication range, the agent communicates with them;
3. If there are hostile agents within its fire range, the agent fires at the closest hostile agent;
4. If there is more than one enemy with the same shortest distance, the agent selects one enemy to shoot at random;
5. If there is no collision between agents, the agent moves to the cell within its movement range with the highest weight (lowest penalty);
6. If there are multiple cells with the same highest weight, the agent selects one cell to move to at random;
7. If collision occurs, a collision resolution mechanism is activated to solve it.

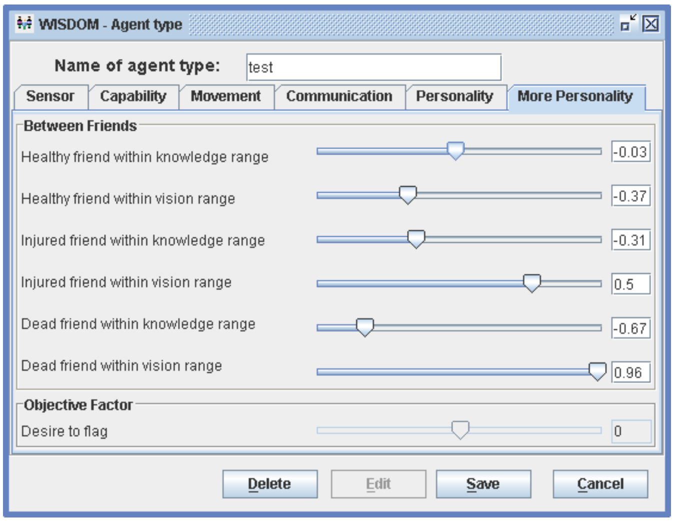
3.3.2.1 Agent personality

Similar to other existing combat ABD models, agents have a number of parameters defining their personalities in WISDOM-I. Figure 3.5(a) and 3.5(b) are the screen dumps of the interface used to define the personalities of agents for a scenario. The movement of each agent is driven by five different categories of weights. These are the desire to move towards:

1. a healthy friend;
2. an injured friend;



(a) Personality 1



(b) Personality 2

Figure 3.5: Interface for agent personality

3. a healthy enemy;

4. an injured enemy;

5. the goal.

The first four categories each have two weights associated with information gleaned from vision and communication respectively. Overall, we have nine personality weights for each agent.

The value of each personality is a continuous number between -1 and $+1$. A positive weight implies the level of desire to move in the direction associated with the characteristic, while a negative weight implies the level of desire to avoid this direction. For example, a very aggressive agent can be modelled by assigning high value to the personalities related to the enemy and a very defensive agent can be modelled by assigning high value to the personalities related to the friend.

3.3.2.2 Movement algorithm

Movements of agents in WISDOM-I are determined by an attraction-repulsion weighting system based on agents' personalities. A penalty function as in Equation 3.1 is constructed using the weights and an agent moves in the direction of the highest weight. Both calculations and moves are done synchronously with collisions resolved as described below. This process is repeated for each time step in the simulation.

$$W_{new} = \sum_{i=1}^{n} \frac{P_i^v}{D_{new}^i} + \sum_{j=1}^{m} \left(\frac{P_j^c}{D_{new}^j}\right) + \frac{P_{new}^t}{D_{new}^t} \tag{3.1}$$

where:

W_{new} denotes the weight for each possible new location that is available for the agent to move to;

P_i^v denotes the personality weight for an agent in the vision range;

D_{new}^i denotes the distance between the new location that is available for the agent to move to and agent i;

P_j^c denotes the personality weight for an agent in the communication range;

D_{new}^j denotes the distance between the new location that is available for the agent to move to and agent j;

P_{new}^t denotes the desire weight to move towards the target (flag);

D^t_{new} denotes the distance between the new location to the target (flag);

n denotes the number of agents within the vision range;

m denotes the number of agents within the communication range.

The rationality of this attraction-repulsion weighting system is as follows: let us assume that the weight for moving towards an enemy is positive. Using this penalty function, the desire to move to an enemy decays with distance. Agents are encouraged to move to a close-by enemy rather than to a far enemy. During the calculations, we remove duplicates when the sensor and communication ranges overlap. For example, a friend agent in cell (i, j) can see an enemy agent in cell $(i+2, j)$ while simultaneously receiving information from another friend about that same enemy agent. In this case, the system does not duplicate the calculations for the weights.

The agent will always move to the cell with maximum weight. If there is a tie, the agent selects a cell, among the cells in tie, at random.

The penalty function used in WISDOM-I is different from that in ISAAC, EINSTein and MANA version 2.0. Gill argued that the movement algorithm may generate strange behaviours in ISAAC, EINSTein and MANA version 2.0 (Gill 2004). To avoid these behaviours, each weight is normalized with its distance in WISDOM-I. In this way, the influence of closer agents is stronger than that of the agents far away.

3.3.2.3 Collision resolution mechanism

In WISDOM-I, each cell can only accommodate one alive agent. If more than one agent would like to move to the same cell, a collision occurs; then the collision resolution mechanism is used to remove it. The collision resolution mechanism is defined by a set of rules:

- The agent occupying this cell at the previous time step has the highest priority to stay in this cell;

- The agent with higher rank has the second highest priority to move to this cell;
- The injured agent has the third highest priority to move to this cell;
- If multiple agents with the same priority wish to occupy the cell, one is randomly chosen to move and the others stay in their original cells.

3.3.3 Terrain feature

Several formats of graphic files can be loaded and used as terrain map, such as jpg, gif and png file. However only the plain terrain is supported in WISDOM-I.

3.3.4 Interactive simulation

Interactive simulation in WISDOM-I (see Figure 3.6) is similar to existing combat ABDs. This mode enables users to interactively control the simulation. The user may pause, resume and restart each simulation, and track each agent's status. Due to the use of different icons for different teams and different status of agents, it is easy for users to see what is happening on the battlefield.

3.3.5 Embedded EC engine

WISDOM-I embeds an EC engine (see Figure 3.7 and 3.8) which can call the simulation engine to evaluate potential configurations. The differential evolution algorithm (Abbass and Sarker 2002) is adopted to search for optimal combinations of personalities under a certain predefined scenario. It can be run under windows or unix environments from an input file in text format. Under the windows environment, some statistical graphs are automatically generated after the end of evolution (see

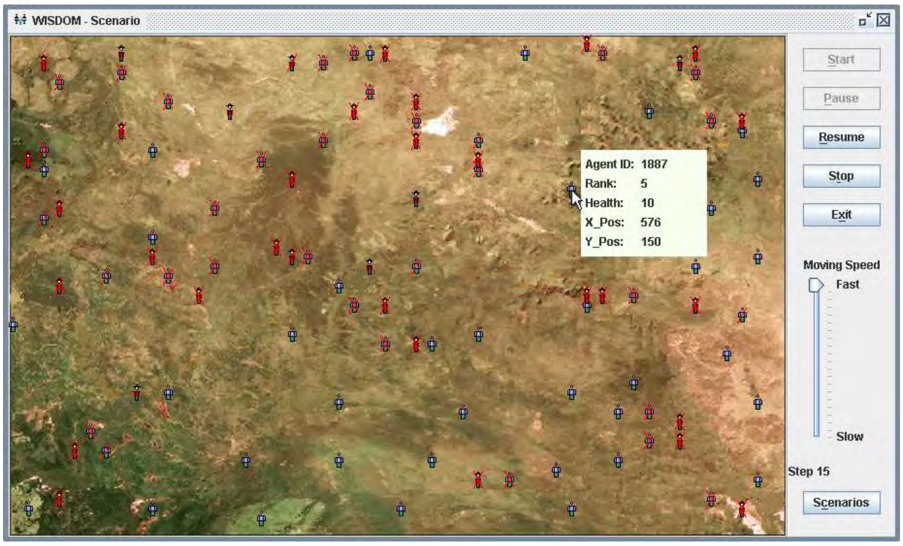


Figure 3.6: Interface for simulation

Figure 3.9). Under the unix environment, analysts need to provide an input file which contains all necessary information required by the simulation engine. At the end of the evolution, an output file is generated automatically, which contains some statistics, such as the fitness value in each run, genotypes in each run and alive agents in each team. Then statistical analysis can be conducted; and the fitness landscape or any other statistic graphics can be visualized with this output file by using a third party graphics program.

3.3.6 Data farming

One of the critical differences from existing ABDs is that WISDOM-I uses the Mysql database engine (see Figure 3.3) to store information. This allows users to record as much data as they want and makes the system more efficient, and easier to maintain. With this database engine, the system can easily run in different places with the same configurations. Other advantages of the database engine are that analysts can play back any simulation previously run and any statistical information can easily be extracted and presented in graphics. The schema of the database engine is shown in Figure 3.10.

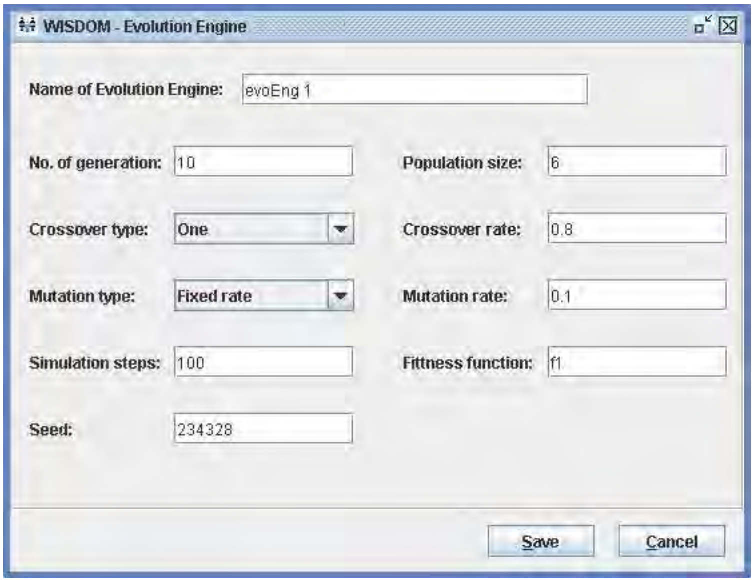


Figure 3.7: Evolutionary computation configuration

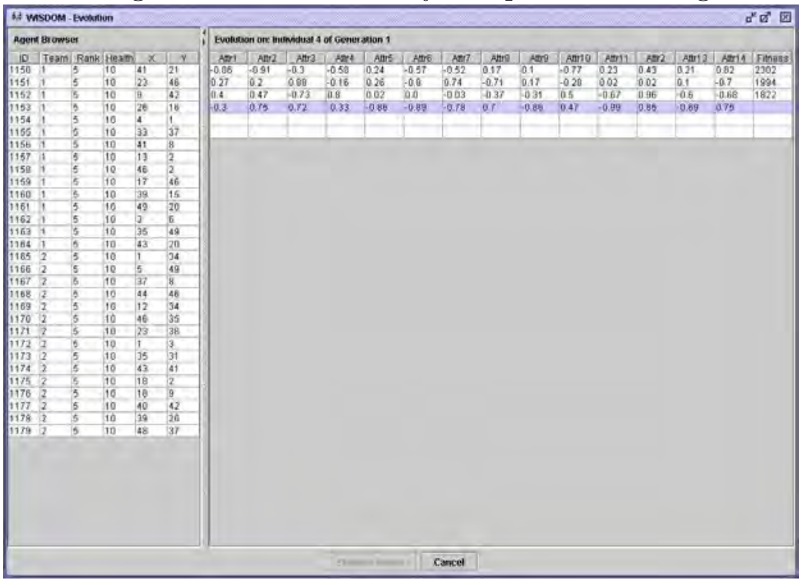


Figure 3.8: Evolutionary computation mode

3.3.7 Program flow

There are two levels of processes in WISDOM-I: user level (Figure 3.11) and system level (Figure 3.14). At the user level, the user needs to follow five basic steps:

1. Customize the system. This step includes activities of selecting the run mode (see Figure 3.12), creating combat teams (see Figure 3.13) and environment

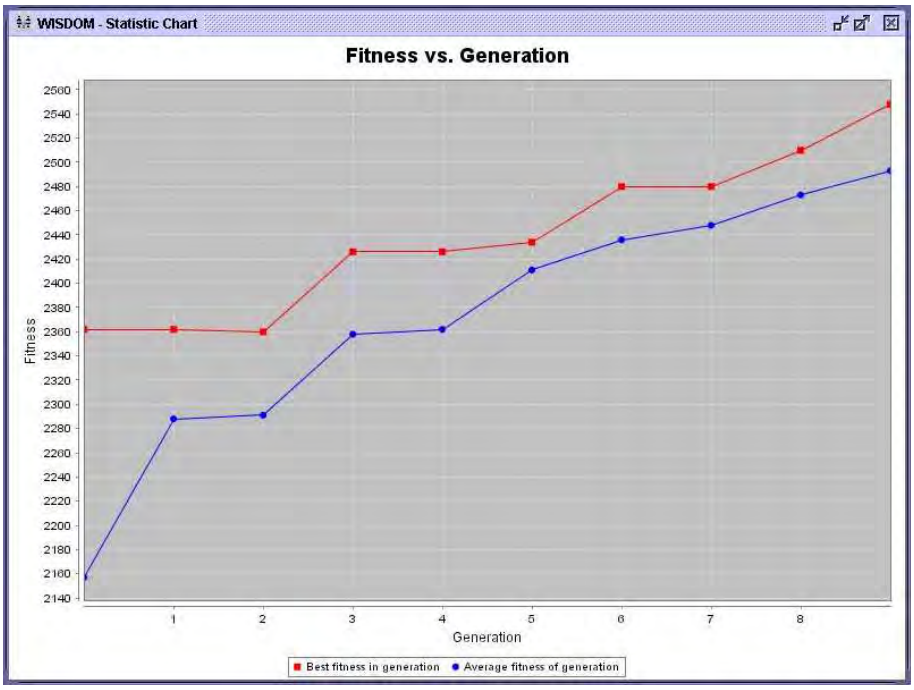


Figure 3.9: Example of statistic graphics

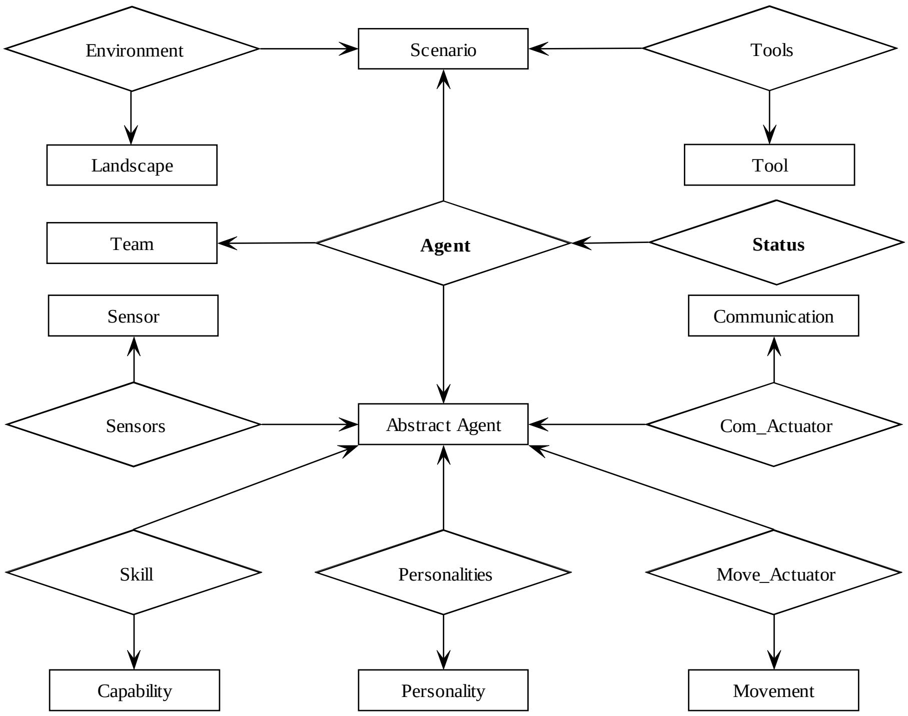


Figure 3.10: HERM diagram for database

(see Figure 3.4), and configuring the database (see Figure 3.3);

2. Start the simulation based on the scenario that has just been created or selected.

3. Pause simulation. During the simulation, the user may pause it to collect the information or change agent status.

4. Resume simulation.

5. Terminate simulation.

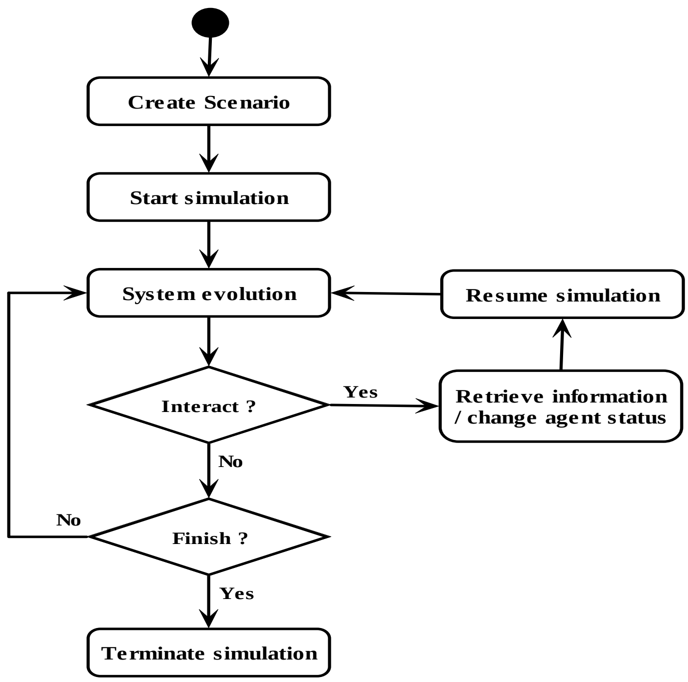


Figure 3.11: System process – user level

A typical sequence of steps during a simulation consists of multiple loops (Figure 3.14) that include the following basic steps:

- Initialize agents, battlefield and time-step counter;
- Go through each agent of each combat team;

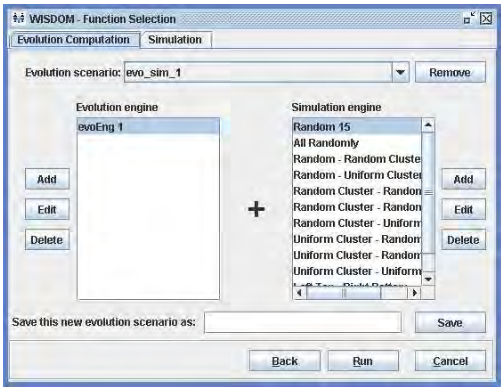


Figure 3.12: Select run mode

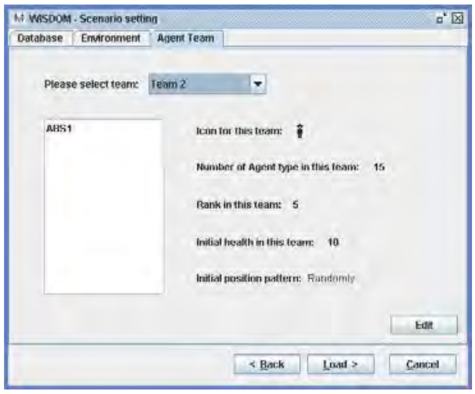


Figure 3.13: Create combating team

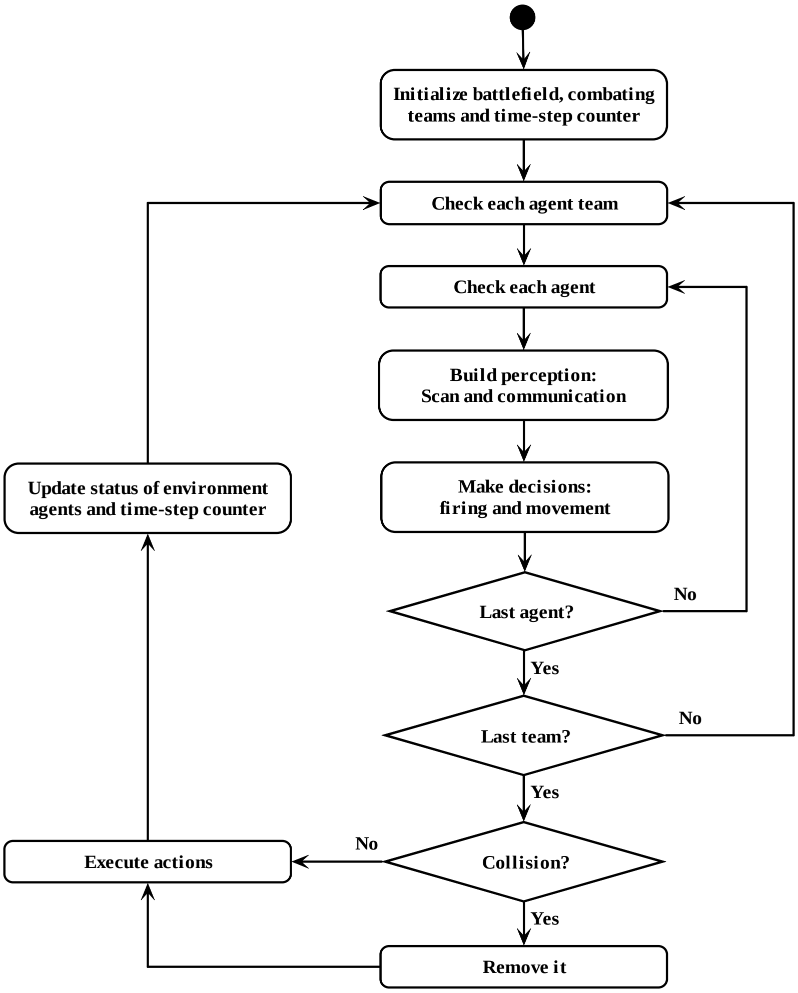


Figure 3.14: System process – system level

- Each agent builds its perception by scanning its environment and communicating with its friends;

- Based on its perception and goal, each agent makes decisions;
- If there is collision from proposed movement decisions, a rule based collision resolution mechanism is adopted to remove it;
- When decisions determined for all agents, they execute the actions;
- Update status of agents, e.g. update their health, position and time-step counter.

3.4 Summary

In this chapter, version I of WISDOM (WISDOM-I) is proposed, which is built on the idea that combat is a CAS. Similar to ISAAC, EINSTein, MANA and CROCADILE, WISDOM-I is a low-resolution abstract model for combat. The detailed physics of combat are ignored while only essential characteristics of the combatant, defence operation or behaviors are modelled.

Each agent in WISDOM-I has six characteristics: perception, capability, execution, reasoning or decision making mechanism, health and rank. Agents may communicate with each other, fire at their enemy or move close to or far from other agents. Rule based decision making mechanism is employed in WISDOM-I. The user may specify the rules for each type of agents to make decisions. An attraction-repulsion weighting algorithm is adopted to decide where the agent should move to. This type of algorithm is widely used in existing ABDs, e.g. ISAAC, EINSTEin and MANA. Every simulation time step, the penalty or the weight of each possible place is calculated based on its situation awareness. The agent always move to the place with lowest penalty or highest weight.

Two running modes are supported in WISDOM-I: interactive simulation and offline batch mode. Interactive simulation enables users to interactively control the simulation. The simulation can be paused, resumed and restarted, and each agent's status can be tracked. In the offline batch mode, the optimal solutions can be searched for predefined scenarios with a build-in evolutionary computation engine.

One of the major differences from existing ABDs is that WISDOM-I uses Mysql database engine to store information. This makes the system more efficient and easier to maintain. The system can be easily run on different place with same configuration. Analysts can easily play back any simulation set up and run. Any statistic information can easily be extracted and presented as graphs.

The main improvements of WISDOM-I are:

1. Using agent based communication instead of squad based communication in existing ABDs, which increases the complexity of the system and makes the system more realistic;

2. Using a relational database (MySql) to store information. This facilitates post-analysis;

3. Improving the movement algorithm to avoid strange behaviors, which may be exhibited in other existing ABDs;

4. Embedding EC engine (single objective or multiple objectives) to search for optimal capability of a force for certain predefined scenarios.

In the next chapter, WISDOM-I is used as a simulation platform, based on which a fitness landscape analysis is conduced to characterize the search space and the problem difficulty of combat simulations.

Chapter 4

Fitness Landscape Analysis [1]

In 1859 Charles Darwin proposed the theory of evolution in the *Origin of Species*, which is the central mechanism of development of natural living systems. Currently many scientists or researchers borrow this idea and apply it into various research areas, especially artificial life. GA has been thought of as an ideal computational model of Darwinian evolution based on the theory of genetic variation and natural selection. It has been largely employed in many artificial life systems through evolving artificial organisms, simulating ecologies and modelling population evolution. One of the main tasks of evolutionary computation is to search for a good solution (high fitness value) within a certain fitness landscape (Mitchell et al. 1992).

The structure and property of the fitness landscape play a major role in determining the success of the search method and the degree of problem difficulty (Horn and Goldberg 1999; Kallel et al. 2001; Mitchell et al. 1992; Teo and Abbass 2003; Vassilev and Miller 2000; Stadler 2002). Smith *et al.* (Smith et al. 2001), and Teo and Abbass (Teo and Abbass 2004b) used fitness landscape analysis to characterize problem difficulties in robotics and artificial organisms. The analysis was useful in designing efficient EC methods (Teo and Abbass 2004a). When EC methods are used to optimize problems where the objective is evaluated through multi-agent

[1]This chapter is based on the publications of Yang et al. (2006b) and Yang et al. (2004a).

simulations, it is essential to understand the underlying nature of the search space and gain insight of the problem difficulties. Although GA and fitness landscape have been adopted in ISAAC (Ilachinski 1997) and EINSTein (Ilachinski 1999) to search for the optimal capability of a force for a predefined scenario, the characteristics of the search space associated with multi-agent combat simulations have not been touched yet. In this chapter, the structure and property of the fitness landscape associated with WISDOM–I is analysed by comparing the search spaces and analyzing the fitness landscapes in six scenarios. First, the basic concepts and methodologies of fitness landscape analysis are described in the next section. Then a series of experiments is created. Finally, the results of the experiments are analysed.

4.1 Introduction

The concept of fitness landscape was first introduced by Wright (1932) (Wright 1932) in biology to represent adaptive evolution as the population navigates on a mountainous surface where the height of a point specifies how well the corresponding organism is adapted to its environment. It is a powerful tool for visualizing the relationship between genotypes (or phenotypes) and reproductive success (fitness value)(Stadler 1995; Stadler 1996). The landscape is usually perceived as mountains with a number of local peaks, valleys, and flat areas representing solutions with equal fitness values. The fitness landscape is rugged when there are many local peaks surrounded by deep valleys. A fitness landscape is characterized by the following three components (Hordijk 1996; Merz and Freisleben 1999; Vassilev and Miller 2000; Stadler 2002):

- A set of genotypes;
- A fitness function that maps each genotype to a scalar; and
- A topological neighbourhood structure that denotes the proximity of genotypes in the search space.

Formally, a fitness landscape for a given problem can be defined as a tuple $\Gamma = (S, f, d)$ and consists of a set of points (solutions) S, a fitness function $f : S \rightarrow \Re$, which assigns a real valued fitness to each of the points in S and a distance metric $d : S \times S \rightarrow \Re$, for which it is required that:

$$d(s,t) \geq 0, \quad d(s,t) = 0 \Leftrightarrow s = t, \quad d(s,t) \leq d(s,u) + d(u,t) \quad \forall s,t,u \in S.$$

Furthermore, $d_{min} \leq d(s,t) \leq d_{max} \quad \forall s,t \in S \wedge s \neq t$. The fitness landscape can be interpreted as a graph $G_\Gamma = (V, E)$ with vertex set $V = S$ and edge set $E = (s, s') \in S \times S | d(s, s') = d_{min}$. The diameter of the landscape is the maximum distance between two points in the graph and is denoted $diamG_\Gamma$, thus $d_{max} = diamG_\Gamma$. The vertices are points in a search space of possible inputs and outputs for the specific operator that is being considered. Each vertex is labelled with a fitness value that is evaluated by the fitness function. An arc from point a to point b is labelled with the probability that point a is transformed to point b by the specific operator (for example mutation). A vertex in such a graph does not necessary match a single genotype. The crossover operator, for example, may take two genotypes as its input, and produce two genotypes as output. In this case, a vertex in the graph is a pair of genotypes.

Unfortunately, there exists no comprehensive theory that formalizes sufficient measures to characterize the difficulty of problems. However, there are some guidelines which have been suggested by a number of researchers that can help with this characterization. Merz and Freisleben (1999) suggested four properties that influence the search space difficulties; these are:

- the fitness differences between neighbouring points in the landscape;
- the number of peaks (modality);
- the distribution of the peaks in the search space;
- the topology of the basins of attraction of the peaks.

Other properties include epistasis or linkage, which represents how dependent genes are on each others. The more epistatic the problem is, the harder it is to find its optimum (Smith et al. 2002; Teo and Abbass 2004b; Kallel et al. 2001; Merz and Freisleben 1999; Goldberg 2002). Two main approaches were used in the literature to analyze the fitness landscape; these are: statistical analysis and information analysis. The former approach usually uses autocorrelation while the latter approach depends on information theory.

4.1.1 Statistical analysis

The most famous technique in statistical measure category is correlation analysis. Correlation analysis is a set of techniques that are intended to characterize the difficulty of a search problem for a genetic algorithm (or any other search technique) by exploiting the fitnesses between neighbouring search points and the correlation of the fitnesses between parents and their offspring. The autocorrelation function of a fitness landscape and the fitness distance correlation are two important measures in correlation analysis.

4.1.1.1 Autocorrelation function

As a measure for characterizing a fitness landscape the autocorrelation function was first introduced by Weinberger (1990). A time series of fitness values is generated through a random walk on the landscape via neighbouring points. Given a fitness landscape (s, f, d), select a random start point s_0 and select a random neighbour s_1, i.e. $d(s_0, s_1) = 1$, repeat this process N times, and collect the fitnesses $f(s_i)$ of the encountered search point s_i $i = 0, ...N$. This way a time series

$$F = (f(s_0), f(s_1), ..., f(s_{i-1}), f(s_i), f(s_{i+1}), ..., f(s_N))$$

is obtained in which the pairs s_{i-1}, s_i and s_i, s_{i+1} for $i = 0, ...N-1$, are neighbouring search points.

So the autocorrelation function ρ of the time series for lag i is defined as:

$$\rho(i) = \frac{E[f(s_t)f(s_{t+i})] - E[f(s_t)]E[f(s_{t+i})]}{V[f(s_i)]} \tag{4.1}$$

where $E[f(s_t)]$ is the expected value of $f(s_t)$ and $V[f(s_i)]$ is the variance of $f(s_t)$. It always holds that $-1 \leq \rho(i) \leq +1$. If $|\rho(i)|$ is close to one, that means there is high correlation between two neighbouring points. If it is close to zero, it means there is hardly any correlation. So higher $|\rho(i)|$ indicates a smooth landscape while lower $|\rho(i)|$ indicates a rugged landscape.

Based on the autocorrelation function, the correlation length τ can be calculated as:

$$\tau = -\frac{1}{\ln(\rho(1))} \tag{4.2}$$

In a statistical sense, the correlation length gives an indication of the largest "distance" between two points at which the value of one point still can provide some information about the expected value of the other point. In other words, the correlation length τ is the distance beyond which the two fitness points become uncorrelated. Therefore, the higher the correlation length, the smoother the fitness landscape and hence the easier the search for an algorithm based on the underlying neighbourhood of the landscape, since the neighbouring points have a higher correlation.

4.1.1.2 Fitness distance correlation

The fitness distance correlation (FDC) coefficient was proposed by Jones and Forrest (1995) as a measure for the problem difficulty for EC. The FDC measures the correlation between the fitnesses of search points and the distances of these points to the (nearest) global optimum. Suppose we have n number of fitnesses $F = \{f_1, f_2, ..., f_n\}$ and n corresponding distances to the (nearest) global optimum $D = \{d_1, d_2, ..., d_n\}$, then FDC coefficient r can be calculated as:

$$r = \frac{C_{FD}}{S_F S_D} \tag{4.3}$$

where:

$$C_{FD} = \frac{1}{n}\sum_{i=1}^{n}(f_i - \overline{f})(d_i - \overline{d}) \tag{4.4}$$

is the covariance of the series F and D, and $S_F, S_D, \overline{f}$ and $\overline{d}$ are the standard deviations and the means of F and D, respectively. It can be shown that $-1 \leq r \leq +1$. The maximal correlation corresponds to $r = -1$ since the search points at short distances are highly correlated in fitness. Using the FDC coefficient r, three classes of problem difficulty can be defined (Jones and Forrest 1995):

- easy: $r \leq -0.15$;
- difficult: $-0.15 < r < +0.15$;
- misleading: $r \geq +0.15$

4.1.2 Information analysis

The second category of fitness landscape analysis is information analysis. This approach is inspired by classical information theory (Shannon 1948) and algorithmic information theory (Chaitin 1987) and built on the assumption that a fitness landscape can be seen as a set of basic objects each of which is represented by a point and the possible outcomes that may be produced by the corresponding evolutionary operator at that point. Four measures (Vassilev et al. 2000) were proposed for characterizing the structure of a fitness landscape $\mathcal{L}$ through analyzing the time series of fitness values $\{f_t\}_{t=1}^{n}$, which are real numbers taken from the interval $\mathcal{I}$ and obtained by a random walk on this fitness landscape : Information content, Partial information content, Information stability and density-basin information.

4.1.2.1 Information content

Information content ($H(\varepsilon)$) approximates the variety of shapes in the fitness landscape, thus it evaluates the ruggedness of the landscape path with respect to the flat area in the path. It is defined as:

$$H(\varepsilon) = -\sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]} \tag{4.5}$$

where $H(\varepsilon)$ is the entropy of the system and is also referred to as the information content. The probabilities $P_{[pq]}$ present the frequencies of the possible blocks pq of elements from the set $\{\bar{1}, 0, 1\}$. They are given by:

$$P_{[pq]} = \frac{n_{[pq]}}{n} \tag{4.6}$$

where $n_{[pq]}$ is the number of occurrences of pq in the string $S(\varepsilon) = s_1 s_2 s_3 ... s_n$, $s_i \in \{\bar{1}, 0, 1\}$. The string $S(\varepsilon)$ is calculated by:

$$S(\varepsilon) = \Psi_{f_t}(i, \varepsilon) \tag{4.7}$$

where

$$\Psi_{f_t}(i, \varepsilon) = \begin{cases} \bar{1} & \text{if } f_i - f_{i-1} < -\varepsilon \\ 0 & \text{if } |f_i - f_{i-1}| \leq \varepsilon \\ 1 & \text{if } f_i - f_{i-1} > \varepsilon \end{cases} \tag{4.8}$$

for any given value of the parameter ε, which is a real number selected from the range $[0, l_{\mathcal{I}}]$ where $l_{\mathcal{I}}$ is the maximum fitness distance in the sequence of $\{f_t\}_{t=1}^{n}$.

When ε is zero, Ψ_{f_t} is most sensitive to the fitness difference and $S(\varepsilon)$ will be presented as a string of 1's. Thus it provides as much information of the landscape as possible. If ε is $l_{\mathcal{I}}$, Ψ_{f_t} is least sensitive to the fitness difference and $S(\varepsilon)$ will be presented as a string of 0's. Thus it provides the least detailed information of the landscape. ε determines the accuracy of $S(\varepsilon)$, therefore it determines the reliability of the information analysis.

4.1.2.2 Partial information content

The modality encountered during a random walk on a fitness landscape can be characterized by partial information content which may be obtained by removing non-essential parts from $S(\varepsilon)$. It is defined as:

$$M(\varepsilon) = \frac{\mu}{n} \tag{4.9}$$

where n presents the length of the string $S(\varepsilon)$ and μ is the length of new string $S'(\varepsilon)$ which is obtained by removing non-essential parts from $S(\varepsilon)$. The value of μ is evaluated as $\Phi_s(1, 0, 0)$. The function $\Phi_s(i, j, k)$ is defined below to count the slops of the optima that are involved in the string $S(\varepsilon)$:

$$\Phi_s(i, j, k) = \begin{cases} k & \text{if } i > n \\ \Phi_s(i+1, i, k+1) & \text{if } j = 0 \text{ and } s_i \neq 0 \\ \Phi_s(i+1, i, k+1) & \text{if } j > 0, s_i \neq 0 \text{ and } s_i \neq x_j \\ \Phi_s(i+1, j, k) & \text{otherwise} \end{cases} \tag{4.10}$$

When the partial information content $M(\varepsilon)$ is zero, that means no slop is in the path and the landscape is flat. If the partial information content $M(\varepsilon)$ is one, that means the landscape path is maximally multi-modal. Based on the partial information content, the number of optima during the random walk on the landscape can be calculated as $\lfloor \frac{nM(\varepsilon)}{2} \rfloor$.

4.1.2.3 Information stability

The information stability (ε^*) is defined as the smallest value of ε for which the fitness landscape becomes flat. That mean $S(\varepsilon^*)$ is a string of 0's. The higher the information stability, the more flat the fitness landscape.

4.1.2.4 Density-basin information

The density-basin information ($h(\varepsilon)$) evaluates the density and the isolation of the peaks in the landscape. Thus it is an indication of the variety of flat and smooth areas of the fitness landscape. It can be calculated by:

$$h(\varepsilon) = -\sum_{p\in\{\bar{1},0,1\}} P_{[pp]} \log_3 P_{[pp]} \tag{4.11}$$

where pp are sub-blocks of 00, 11, and $\overline{11}$, and $P_{[pp]}$ are the frequencies of the sub-blocks pp. Higher density-basin information means a number of peaks are within a small area while lower density-basin information means an isolated optimum. Therefore it is easier for an evolutionary search process on a fitness landscape with high density-basin information and harder for that with low density-basin information.

4.1.2.5 Summary

In summary, information analysis provides a new approach for fitness landscape analysis. Information content and partial information content measure the amount of information of the landscape. Information stability is a result of filtering out the estimated information content. Density-basin information measures the density of the local optimum. With these four indications information analysis presents comprehensive details of the fitness landscape. Higher information content, partial information content and information stability means higher degree of epistasis and modality which leads to a rugged landscape that is hard to search. Vassilev argued that the correlation analysis only give us a vague notion of the structure of fitness landscape of a given problem (Vassilev et al. 2000). Therefore information analysis approach is adopted to characterize the fitness landscape in this study.

4.2 Experimental setup

The aims of these experiments are twofold. First, the effect of the personality characteristics of the red team on the fitness landscape for evolving best personality characteristics for the blue team is studied. Second, the findings by applying a straightforward $(1 + 1)$ evolutionary strategy (ES) to evolve the personality characteristics for the blue team are compared with the findings of the previous fitness landscape analysis.

Six different scenarios are created for the red team; these scenarios are listed in Table 4.1. In the *Balanced* scenario (BAL), the team members tend to group together, attack the enemy and reach the goal (flag). In the *Goal Oriented* scenario (GOL), team members are neutral about grouping together or attacking the enemy; however, they are determined to get to the flag. In the next four scenarios, the members are neutral about getting to the flag and the emphasis is more on their relationship with the enemy and themselves. In the *Very aggressive* scenario (VAG), the team members tend not to cluster and being focused more on attacking the enemy. In the *aggressive* scenario (AGG), the members tend to be more rational than those in the VAG scenario by being neutral about clustering together while attacking the enemy. In the *Defensive* scenario (DEF), the members tend to cluster together while being neutral about following the enemy. In the *Coward* scenario (COW), they are neutral about clustering together but they run away from the enemy.

In all six scenarios, the "probability of hit" of the red team is fixed to the maximum of 1. The decision variables are represented with a vector of 10 real numbers representing different characteristics of personalities for the blue team as follows:

Table 4.1: Different strategies for the red team used in the experiments

Scenario	**Friend**	**Enemy**	**Goal**
Balanced (BAL)	Cluster	Attack	Target
Goal Oriented (GOL)	Neutral	Neutral	Target
Very Aggressive (VAG)	Avoid	Attack	Neutral
Aggressive (AGG)	Neutral	Attack	Neutral
Defensive (DEF)	Cluster	Neutral	Neutral
Coward (COW)	Neutral	Avoid	Neutral

1. $P_1 - P_4$: attraction/repulsion towards a healthy or an injured enemy in the communication range or in the vision range.

2. $P_5 - P_8$: attraction/repulsion towards a healthy friend or an injured friend in the communication range or in the vision range.

3. P_9: probability of hit.

4. P_{10}: attraction/repulsion towards the flag.

All personalities (decision variables) are real numbers in the range of $[-1, 1]$. The environment is a 50x50 grid and the flag is located at the middle cell of the bottom row. Each red and blue team has 20 agents initialized at random in a 7x7 square area. Both teams are initialized starting from the top to the seventh top row. The red team is initialized two columns away from the middle column in the right direction while the blue team is initialized two columns away from the middle column in the left direction. The initial board is depicted graphically in Figure 4.1. The evaluation of the game involves repeating the simulation 100 times, each for 500 time steps.

The objective of a scenario can be simplified to the following question: "if we are faced with a red team with specific characteristics in an operation, what should be the characteristics of the blue team to achieve maximum damage to the red team and minimum loss to the blue team?". The objective function is chosen to maximize the differential advantage of blue over health; that is, the larger the gap between the damage in the blue team and the damage in the red team, the more likely that

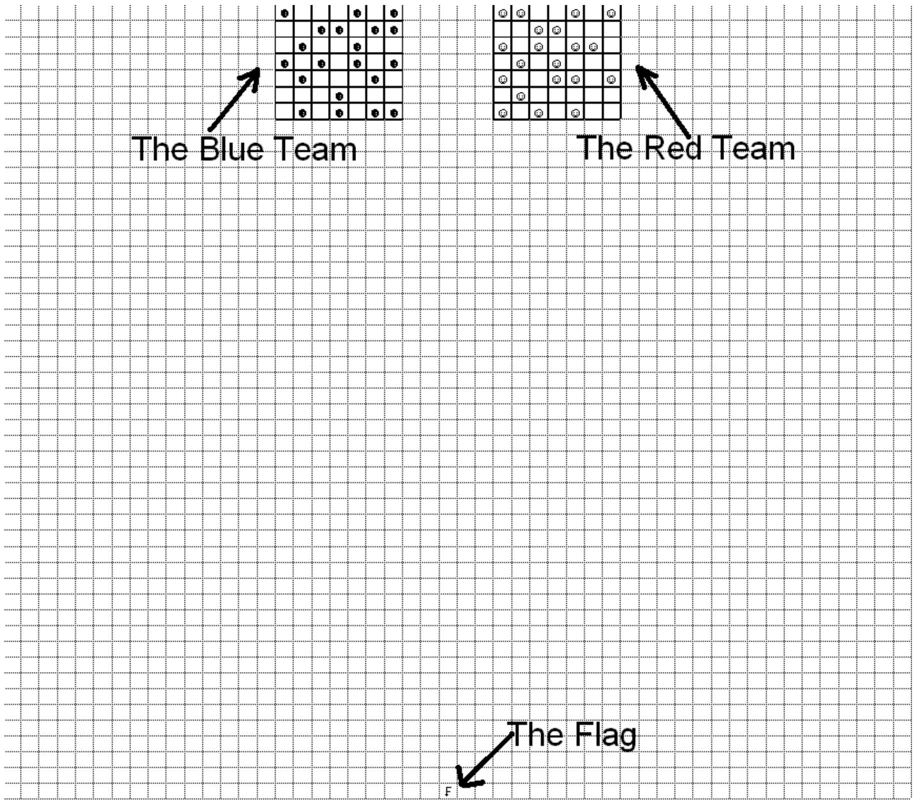


Figure 4.1: A graphical representation of the initial setup of agents in the environment

the team with this differential advantage wins. Formally, the objective is defined as follows. Let

N_b, N_r denotes the number of agents in blue team and red team respectively.

h_i^b, h_i^r denotes the health of blue agent i and the health of red agent i at the end of the simulation respectively.

H_i^b, H_i^r denotes the health of blue agent i and the health of red agent i at the start of the simulation respectively.

$$\Uparrow objective = (\sum_{i=1}^{N_r} H_i^r - \sum_{i=1}^{N_r} h_i^r) - (\sum_{i=1}^{N_b} H_i^b - \sum_{i=1}^{N_b} h_i^b) \tag{4.12}$$

Due to the fact that $\sum_{i=1}^{N_b} H_i^b$ and $\sum_{i=1}^{N_r} H_i^r$ are constants because all agents have the same initial health, the objective function can be reduced to Equation 4.13:

$$\Uparrow objective = \sum_{i=1}^{N_b} h_i^b - \sum_{i=1}^{N_r} h_i^r \tag{4.13}$$

With this objective function, many situations can arise. Three extreme situations are identified:

1. The lower bound in Equation 4.13 that will occur when the blue team does not shoot at the red team (*i.e.* the probability of hit is -1) is equal to $-\sum_{i=1}^{N_r} H_i^r$. In this case, the red team will eliminate all members of the blue team while maintaining their health value to maximum.

2. The upper bound in Equation 4.13 that will occur when the blue team achieves maximum damage to the red team is equal to $\sum_{i=1}^{N_b} H_i^b$. Since the probability of hit is fixed for the red team to a positive value, the only possibility here is that the blue team has manoeuvred efficiently to achieve this result or the red was running away from the blue team while the blue team was shooting at the red team.

3. The situation with the objective value of zero will take place when the loss of both teams is the same, or the loss of both teams is zero (the teams did not engage and were simply running away from each other).

Two different fitness measures are used in this study. To shift the negative values and ensure that all values are non-negative, $\sum_{i=1}^{N_r} H_i^r$ is added to the objective function and average its value over the 100 different simulation repeats as the first fitness function (see Equation 4.14). The second (see Equation 4.15) is the normalized average which is the average fitness normalized by the standard deviation. The first fitness will promote individuals with high average regardless of the stability or consistency of the weight vector in achieving the same results, while the latter will penalize solutions based on their standard deviations. In other words, a fitness value of 300 is better than 400 if the former has a standard deviation of 2 while the latter has a standard deviation of 20.

$$F_1 = \frac{\sum^{100}(\sum_{i=1}^{N_b} h_i^b + \sum_{i=1}^{N_r} H_i^r - \sum_{i=1}^{N_r} h_i^r)}{100} \tag{4.14}$$

$$F_2 = \frac{F_1}{1 + standard \ \ deviation} \tag{4.15}$$

1 is added to the standard deviation to avoid division by 0 and to bound the fitness between the actual average fitness (when the standard deviation is 0) and 0 (when the standard deviation is very large). Thus $0 < F_2 \leq F_1$. This equation has a strong bias for stable solutions.

4.3 Results and Analysis

4.3.1 Random walk

To undertake the fitness landscape analysis, ten different random walks are taken, each of length 10,000 solutions using both fitness functions. Each stochastic neighbourhood in the search space was obtained by adding a random number drawn from a Gaussian distribution with zero mean and 0.1 standard deviation to each variable in the genotype. If the value of any personality is out of the range $[-1, 1]$, the value is truncated. Table 4.2 presents some statistics of the best solutions encountered in the ten random walks.

According to the average fitness in Table 4.2, the best solution expresses some interesting behaviours as the fitness values demonstrate real engagements between the forces and the superior performance of the blue force. It is astonishing to look at the average fitness and compare it to the corresponding characteristics of the red

Table 4.2: Statistics of the best solution found using each fitness function over the 10 runs in the random walk experiment

	Average Fitness (F_1)		Normalized Fitness (F_2)	
	Max	Mean±Stdev	Max	Mean±Stdev
BAL	229.80	219.78 ± 7.74	200.00	200.00 ± 0.00
GOL	229.20	204.45 ± 9.03	200.00	200.00 ± 0.00
VAG	270.32	268.46 ± 1.11	22.22	19.47 ± 1.20
AGG	288.94	285.52 ± 1.60	25.68	19.20 ± 2.75
DEF	232.64	220.07 ± 6.84	200.00	200.00 ± 0.00
COW	208.38	205.61 ± 1.87	200.00	200.00 ± 0.00

team. The best wins for blue occur when red is either VAG or AGG. These two scenarios share the same tendency of the red team members to attack their enemies. The worst win occurred when the red team is either GOL or COW. In the VAG mode, the red agents tend to run after any blue in their vision or communication ranges and avoid grouping with their friends. Therefore, without cooperation among the red team members, the red may be completely damaged although blue may be damaged as well to some degree. In the COW mode, despite that blue can run after red, red is running away from blue and cluster with their friends. Therefore, it may minimize its own damage and annihilate blue invaders effectively. In the GOL scenario, the red team members were also neutral about grouping but their common tendency to get to the flag imposed an implicit tendency to group together. These findings are very interesting as they support the recent development in defence theory and the work on swarm attacks.

However, the best solutions, based on using the average fitness as the objective function, are not stable as can be seen when compared to the runs using the normalized average fitness as the objective function. In the latter case, the search algorithm is biased to more stable solutions. Interestingly, the normalized average fitness is giving a different side of the same story. Table 4.2 shows that the normalized average fitness value in BAL, GOL, DEF and COW scenario converged to the attractor with the average fitness of 200 and zero standard deviation. This attractor covers two cases: (1) there is no engagement taking place between blue and red; and (2) the loss in blue is equal to the loss in red. After a closer look at the runs, the first option is dismissed. Therefore, in these four scenarios, both teams had the same amount of loss. The much lower value of normalized average fitness implied that stochasticity plays an important role in the VAG and AGG scenarios; that is, all solutions encountered were unstable.

Figures 4.2 depicts a representative time series for each experiment being generated by random walk. The figures reveal that the landscape is indeed rugged. However, the landscape for the BAL, GOL, DEF and COW strategies contains many flat areas. In particular, it is clear as shown in the right column of Figure 4.2 that the landscapes for BAL, GOL, DEF, and COW are filled with solutions that lead to tit-for-tat strategies (Axelrod 1984); resulting in an equal red and blue loss, while the landscapes for VAG and AGG are void of solutions leading to tit-for-tat. The reason that this tit-for-tat is common is because the game is symmetric. Researches (Axelrod 1984; Milinski 1987; Nowak and Sigmund 1992) has shown that tit-for-tat is a very common and successful strategy in symmetric games such as Prisoner's Dilemma (IPD) game while the player usually cannot mimic the behaviour of its opponent (tit-for-tat) since different players have different strategies in asymmetric game.

The fitness signal is usually defined in the literature as the difference between the fitness of the best solution and second best. We call this signal as signal-best. A more generic definition used in this analysis is to define the signal as the difference between the best and worst fitness values encountered during the search. We call this signal as signal-worst. The concept of signal-worst provides a simple mechanism to understand the range of fitness values in a landscape. Accordingly, in term of average fitness, one can see that the value of signal-worst with the BAL, GOL, DEF and COW scenarios is lower than that with the VAG and AGG strategies. The good solutions (those with fitness values over 200) in BAL and DEF scenarios seem to be more isolated and surrounded with low fitness values. There is almost no good solution found in the GOL and COW scenarios.

However, the previous findings do not continue to hold when looking at the nor-

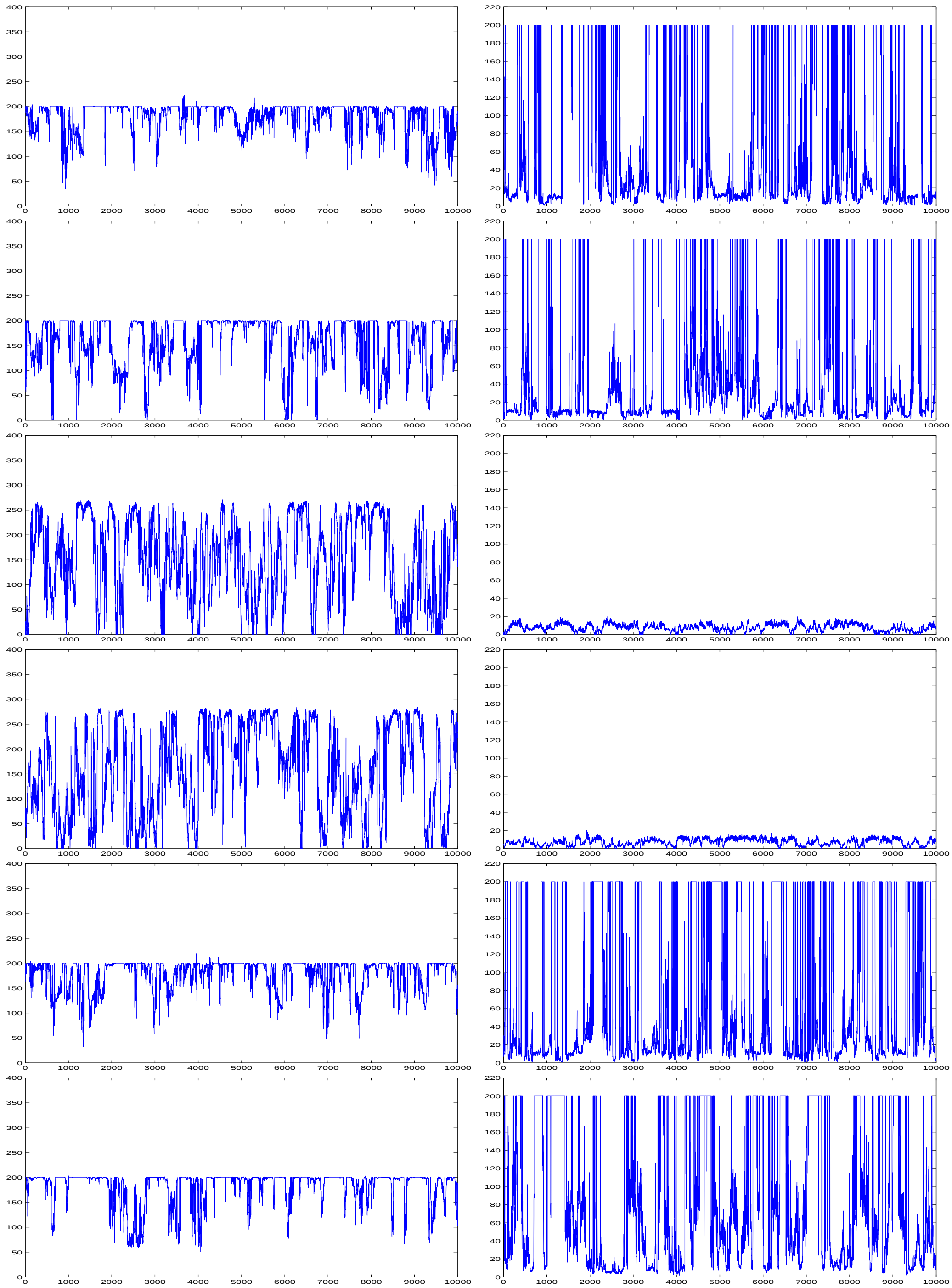


Figure 4.2: Fitness value over time for random walk using average fitness(left column) and normalized average fitness (right column). The order from top down is: Balanced, Goal Oriented, Very Aggressive, Aggressive, Defensive, Coward, respectively

malized fitness time series. One can see that the minimum signal-worst occurs with the VAG and AGG strategies while the BAL, GOL, DEF and COW strategies have

Table 4.3: The information theoretic measures over six strategies using both fitness functions

		ϵ^*	$H(\epsilon)$	$M(\epsilon)$	Exp. # of Opt.
Average Fitness	BAL	129.90 ± 14.59	0.43 ± 0.01	0.39 ± 0.02	1966.20 ± 115.57
	GOL	177.30 ± 17.66	0.45 ± 0.01	0.42 ± 0.03	2104.90 ± 128.82
	VAG	140.60 ± 7.37	0.44 ± 0.01	0.52 ± 0.01	2608.20 ± 41.13
	AGG	182.30 ± 7.66	0.41 ± 0.00	0.54 ± 0.01	2684.70 ± 39.69
	DEF	143.40 ± 21.44	0.43 ± 0.01	0.39 ± 0.04	1948.00 ± 195.39
	COW	123.30 ± 13.70	0.46 ± 0.01	0.42 ± 0.03	2079.50 ± 160.72
Normalized Average Fitness	BAL	199.00 ± 0.47	0.43 ± 0.01	0.43 ± 0.02	2127.00 ± 112.07
	GOL	199.00 ± 0.94	0.45 ± 0.01	0.47 ± 0.03	2336.20 ± 135.86
	VAG	8.80 ± 0.79	0.45 ± 0.01	0.59 ± 0.01	2959.70 ± 31.00
	AGG	9.60 ± 1.35	0.41 ± 0.00	0.60 ± 0.01	2988.30 ± 40.68
	DEF	199.00 ± 0.67	0.43 ± 0.01	0.42 ± 0.04	2106.50 ± 217.28
	COW	197.20 ± 1.32	0.44 ± 0.01	0.45 ± 0.03	2238.30 ± 164.48

almost the same value of signal-worst. It is also clear that the landscape is very rugged using the VAG and AGG strategies while it contains a number of flat regions when using the other four strategies.

Table 4.3 lists the results of the fitness landscape analysis by using information content approach. It is clear that the findings between the two fitness values are consistent with the previous discussion. It is also apparent that both landscapes are similar except for the value of ϵ^* in the VAG and AGG scenarios using normalized average fitness, where they have the highest number of peaks. The partial information content has the highest value with these two scenarios. This implies that the fitness landscape of these two scenarios under the normalized average fitness function is highly multi-modal. Referring to the information content, the similar values occurred in all scenarios suggest that the degree of ruggedness is similar between the landscapes of all scenarios regardless of which fitness function is used.

Interestingly the VAG and AGG scenario have similar information stability when using the average fitness as the objective function, but have much lower value when using the normalized average fitness. What is intriguing here is that the fitness landscapes for both fitness functions have very similar characteristics despite the differences in the distribution of fitness values embedded in these spaces.

In terms of information stability, one can see that it requires high value of ϵ^* except for the normalized average fitness in the VAG and AGG scenarios. The high value of ϵ^* is almost 50% of the upper bound on the fitness function. This entails that the highest difference between two solutions in the neighbourhood is less than or equal to 50% of the upper bound on the fitness value.

By scrutinizing Figures 4.3, one can see that despite the similarities between the fitness landscapes of the two fitness functions in BAL, GOL, DEF and COW scenarios, there are small peaks between zero and 20 in terms of normalized average fitness. This implies that there are a number of local optima appearing at the fitness value between zero and 20 in the landscape of both fitness functions. The big peak at 200 tells us that most optimal solutions are occurring at the fitness value of 200 and also suggests that many of the strategies are tit-for-tat. For the VAG and AGG scenarios, the frequencies for encountering solutions is very similar, but the likelihood for the blue team to find a good solution with fitness value above 250 is higher when using the average fitness as the objective function. In terms of normalized average fitness, all the solutions are clustered between 0 and 20. This suggests that solutions in the VAG and AGG scenarios are very unstable (the standard deviation is high).

4.3.2 $(1+1)$ Evolution strategy

A $(1+1)$ evolution strategy ($(1+1)ES$) can be seen as a special case of evolutionary methods with a population size of one. Similar to random walk, a solution is a vector of ten real value numbers. To test the findings of the fitness landscape analysis, a straightforward $(1+1)ES$ is adopted. In order to maintain the same fitness landscape, the same neighbouring mechanism is used; therefore, the step adopted in the $(1+1)ES$ is fixed and is not adaptive. This setup is certainly not the best to achieve good results for our problem. However, the aim of these experiments is to compare the effect of the bias generated from the use of the fitness to guide solutions in the search space on the quality of solutions obtained.

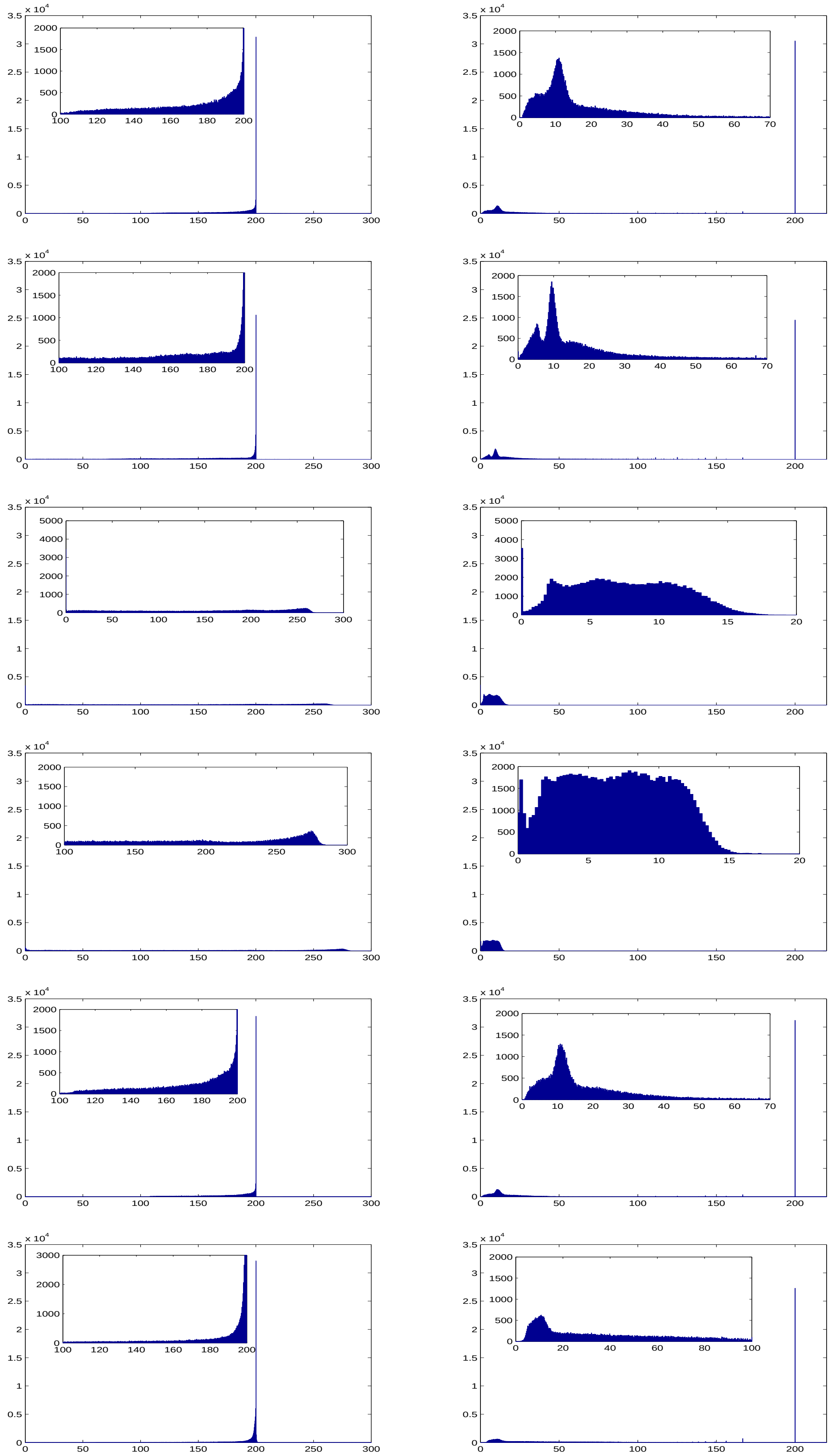


Figure 4.3: Fitness histogram for random walk using average fitness (left column) and normalized average fitness (right column). The order from top down is: Balanced, Goal Oriented, Very Aggressive, Aggressive, Defensive, Coward, respectively

The $(1 + 1)ES$ adopted starts by generating a solution at random. The initial solution is considered as the best solution found so far. The new solution is obtained by adding a random number drawn from a Gaussian distribution with zero mean and 0.1 standard deviation to each personality of the best solution found so far. If the new solution is better than or equal to the best solution found so far, the former replaces the latter. If not, a new solution is generated and the process continues until the maximum number of objective evaluations allowed is reached; after which, the algorithm terminates. This simple and straightforward $(1+1)ES$ can be seen as a stochastic hill climber which also allows movements on plateaus in the landscape. Therefore neutral mutations are accepted to help escaping flat areas and possibly jump from shallow areas; thanks to the role of the stochastic neighbourhood.

Similar to random walk, the experiments are repeated ten times and in each, the run is stopped after a total of 10,000 solutions are generated. This setup is equivalent to 10^6 objective evaluations (calls to the simulators) after taking into account the 100 evaluations per solution. The experiments are performed for both fitness functions: the average fitness (called ESAvg) and the average normalized fitness (called ESNorm).

Table 4.4 lists the best solution found using ESAvg and ESNorm for each scenario over all runs respectively. Figure 4.4 also shows the progression of the best solution over time for each of the ten runs. The best solution found by ESAvg is similar to that encountered by random walks. Similar to the results of random walk using normalized average fitness, ESNorm got stuck in the attractor with the normalized fitness value of 200 except in the case of VAG and AGG scenario despite that from the fitness landscape analysis, the VAG and AGG scenario had a higher number of peaks when compared to the other scenarios.

The best overall fitness is achieved with the VAG and AGG scenarios in terms of average fitness; that is, when the enemy does not care about its own safety and only care about attacking blue agents. Due to the stochasticity, the fitness value of the best solution is pretty low in terms of normalized average fitness for the VAG and

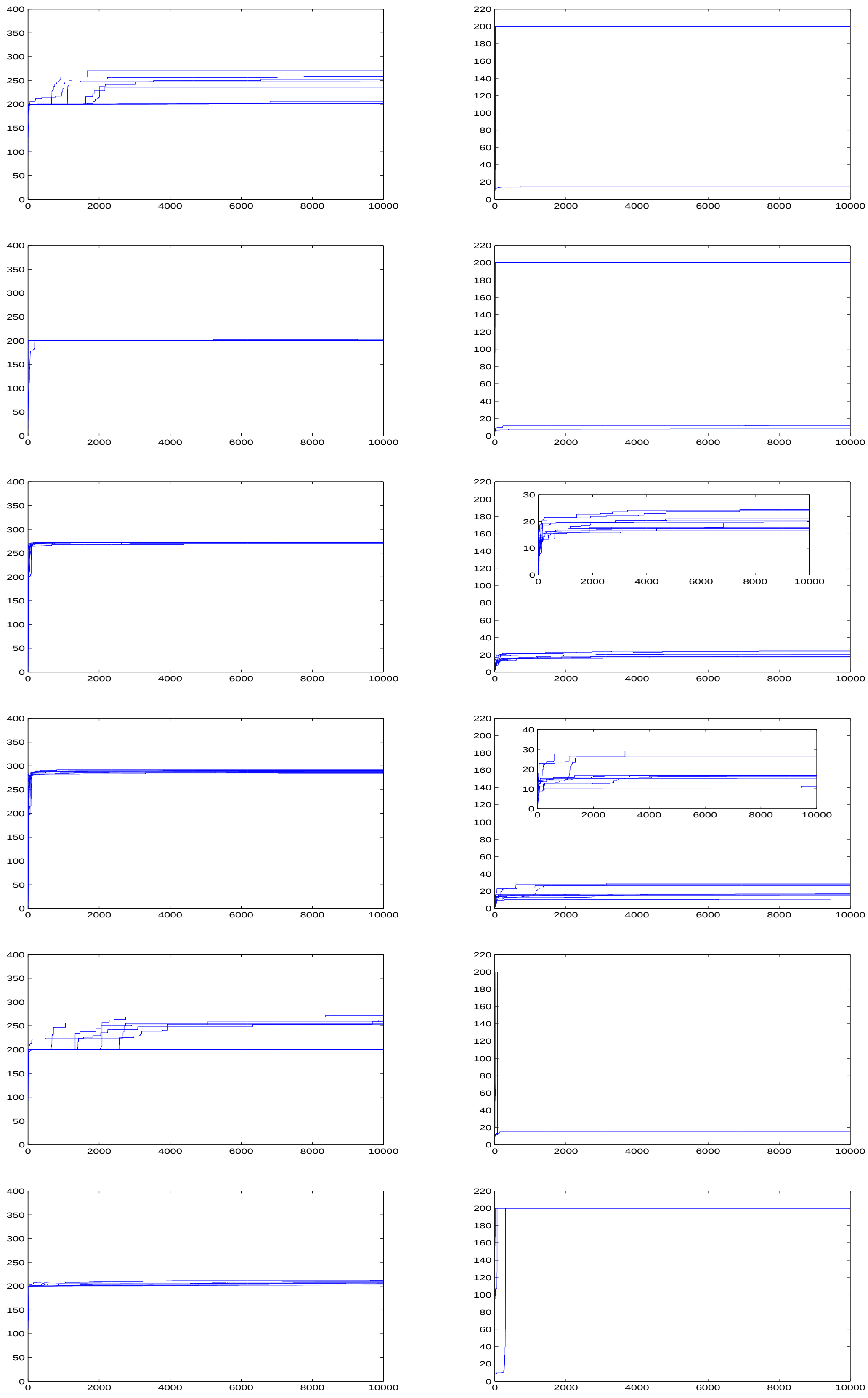


Figure 4.4: The average fitness of the best solution found over time for ESAvg (left column) and ESNorm (right column). The order from top down is: Balanced, Goal Oriented, Very Aggressive, Aggressive, Defensive, Coward, respectively

Table 4.4: Best solution found for $(1+1)ES$ using both fitness functions over 10 independent runs

	Average Fitness		Normalized Average Fitness	
	Max	Mean±Stdev	Max	Mean±Stdev
BAL	270.56	227.50 ± 28.42	200.00	181.53 ± 58.41
GOL	202.60	200.97 ± 0.61	200.00	161.96 ± 80.20
VAG	273.38	272.15 ± 1.14	24.55	19.95 ± 2.73
AGG	291.56	288.41 ± 2.53	29.10	19.32 ± 6.06
DEF	271.78	235.97 ± 30.91	200.00	181.51 ± 58.48
COW	211.70	206.98 ± 3.19	200.00	200.00 ± 0.00

AGG scenarios. These results are consistent with the findings based on the earlier fitness landscape analysis.

By looking at the fitness histograms encountered by the evolutionary strategies as shown in Figure 4.5, one can certainly find clear differences as the attractor with a fitness value of 200 did not dominate the frequencies as much as with the random walk. Since there is a large amount of the points in the fitness landscape having fitness value of 200, the highly explorative search, such as random walk, will very likely obtain the fitness of that level. However, a more exploitive search, such as $(1+1)ES$, is able to make use of neighbouring information thus would obtain better solutions.

In the remaining analysis of this section, the behavioral characteristics of agents corresponding to the best solutions found by the evolutionary strategy using the average fitness is studied. Figure 4.6 shows the amount of damage caused to blue and red according to each scenario. Despite that the highest level of fitness occurred with the AGG then the VAG scenarios, the highest level of damage to the red team occurred with the DEF scenario followed by the BAL then the AGG. Same as red team, the highest level of damage to the blue team occurred with the DEF scenario followed by the BAL scenario. However, the third highest level of damage in the blue team occurred with the GOL scenario followed by the COW scenario then VAG and AGG. The best ratio of damage between red and blue occurs when the red follows an AGG scenario. The worst ratio of damage occurs when the red follows the GOL

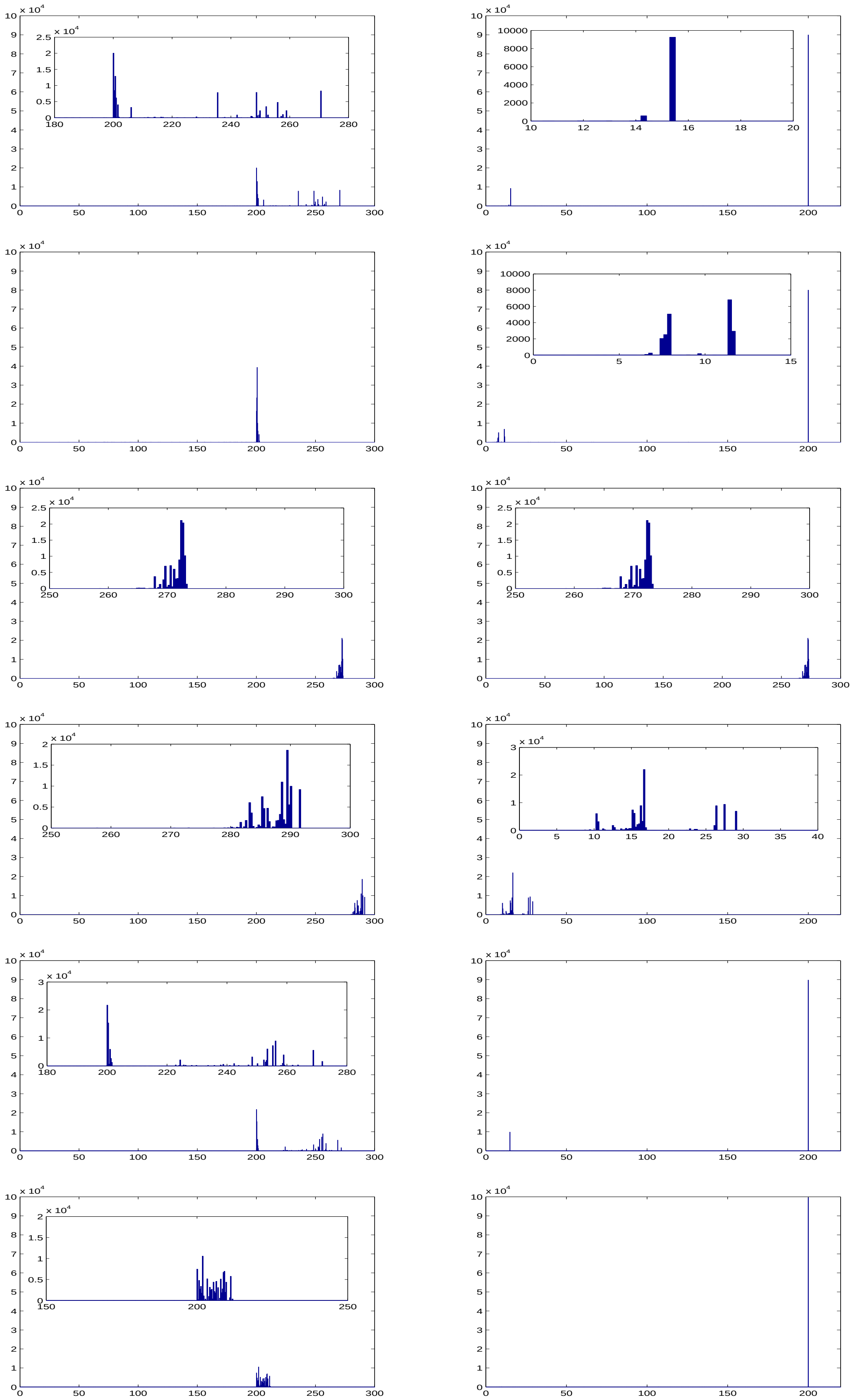


Figure 4.5: Fitness histogram for $(1+1)ES$ using average fitness (left column) and normalized average fitness (right column). The order from top down is: Balanced, Goal Oriented, Very Aggressive, Aggressive, Defensive, Coward, respectively

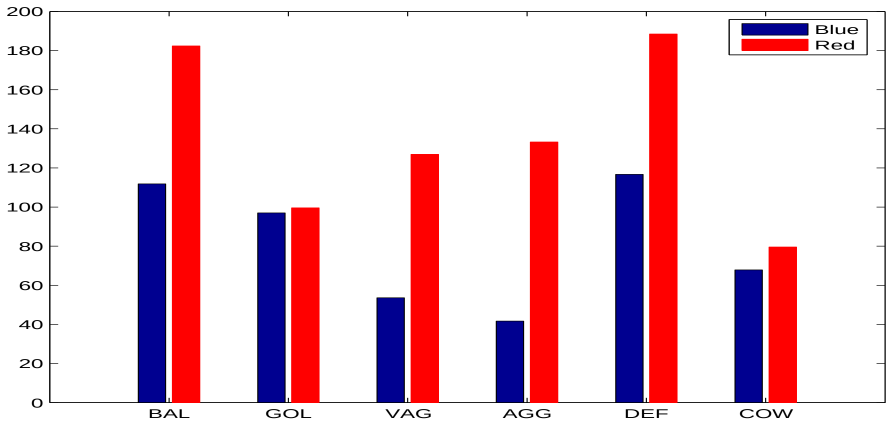


Figure 4.6: The level of damage achieved to each team according to the best solution found for each scenario. BAL: Balanced, GOL: Goal Oriented, VAG: Very Aggressive, AGG: Aggressive, DEF: Defensive, COW: Coward

scenario. Recalling the results in Table 4.4, they are consistent with the ratio of damage.

The second type of analysis looks at the personalities evolved by the blue team corresponding to the best solution found for each scenario. Table 4.5 provides the personality weights.

Table 4.5: The characteristics of the personalities corresponding to the best evolved solution for each scenario. 'H' denotes healthy, 'E' denotes enemy, 'C' denotes communication, 'V' denotes vision, 'I' denotes injured, and 'F' denotes friend

	BAL	GOL	VAG	AGG	DEF	COW
HEV	-0.50	0.89	-0.18	0.28	-0.66	-0.16
HEC	0.85	0.72	0.79	0.90	0.82	0.95
IEV	0.02	-0.63	0.48	-0.63	-0.63	-0.86
IEC	0.57	0.32	-0.64	0.44	0.95	0.91
HFV	-0.01	0.00	-0.06	-0.05	-0.01	-0.07
HFC	0.01	0.00	0.80	0.30	0.00	0.00
IFV	0.22	1.00	0.20	0.35	0.64	-0.06
IFC	0.37	0.07	1.00	0.87	-0.02	-0.12
Prob. of hit	1.00	1.00	1.00	0.91	1.00	1.00
Desire Flag	-1.00	-0.59	0.26	0.54	-1.00	0.73

Interestingly, there is a general trend in the behaviour of the blue to be neutral or attracted to friends, especially to injured friend. There is also a general trend of being attracted to enemy within communication range except in the case when red is VAG, the attraction to injured enemy within communication range becomes negative. In most cases the blue agent tends to escape from the enemy within vision range except when the red team took the GOL and AGG strategy, the blue agents prefer to be closes to healthy enemy within the vision range. When the red team follow VAG and BAL scenario, the blue agents prefer to be closes to injured enemies within the vision range. In any case, the blue agents always fire at the enemy. Overall, it seems that the general strategy achieved by the blue in all experiments is to defend injured friends, and hit and cause maximum damage to enemies within communication range. This strategy makes sense. Defending injured friends helps to reduce death in the blue force. In addition, because the communication range is larger than the vision range, attacking the enemy within communication range leads also to destroying close-by enemies.

After these experiments, a critical question for defence analysts is how sensitive WISDOM–I is to the personality parameters used in the simulations. The following section describes the sensitivity test used to analyse this issue.

4.3.3 Sensitivity test

In order to test the sensitivity of the model, some experiments are conducted based on the VAG scenario using random walk after slight changes in the personality weights for the red agents. Three scenarios are created by perturbing the weights with a small step of 0.05 around the original values of some parameters while maintaining the rest fixed. The three scenarios correspond to changes in the weights for friendly agents (FRD), enemy (EMY), and probability to hit (SHT).

It is found that almost no difference in the fitness landscape when the parameters are perturbed. The plots for the histogram are not presented because they are very similar to the original plots shown before. The information theoretic measures confirm the findings as shown in Table 4.6.

Up to now, a fitness landscape analysis is conducted by using single fitness function.

Table 4.6: The information theoretic measures over four testing scenarios using both fitness functions

		ϵ^*	$H(\epsilon)$	$M(\epsilon)$	Exp. # of Opt.
Average Fitness	VAG	140.60 ± 7.37	0.44 ± 0.01	0.52 ± 0.01	2608.20 ± 41.13
	EMY	140.70 ± 7.78	0.44 ± 0.01	0.52 ± 0.00	2602.70 ± 24.93
	FRD	135.30 ± 8.41	0.44 ± 0.01	0.52 ± 0.00	2591.90 ± 19.60
	SHT	148.00 ± 4.42	0.44 ± 0.00	0.52 ± 0.00	2612.90 ± 21.70
Normalized Average Fitness	VAG	8.80 ± 0.79	0.45 ± 0.01	0.59 ± 0.01	2959.70 ± 31.00
	EMY	8.50 ± 1.35	0.45 ± 0.01	0.59 ± 0.01	2959.50 ± 37.33
	FRD	7.80 ± 1.23	0.45 ± 0.01	0.59 ± 0.01	2941.30 ± 35.06
	SHT	8.00 ± 0.82	0.45 ± 0.00	0.59 ± 0.00	2968.60 ± 20.34

When looking at Figure 4.6, one may notice that higher fitness values do not always mean higher damage to the red team and also do not always mean lower damage to the blue team. But the desired objective is to minimize the damage of the blue team and maximize the damage of the red team during capturing the flag. Although the integration of these two objectives into a single fitness function simplifies the mathematics, mixing the health of the blue and red teams hides information during the search for a good strategy. Two solutions can exhibit the same fitness value with very different characteristics. One suggestion is to use the ratio of damage as the fitness instead of the health. Still, the ratio of damage hides the amount of damage. For example a ratio of damage of 1:2 does not tell us much about how much the red or blue team lost.

To overcome this problem, a further analysis is performed in the next section by using a Pareto-based multi-objective evolutionary approach.

4.3.4 Multi-objective analysis

An evolutionary multi-objective optimization (EMO) approach attempts to search for optimal solutions to a problem with multiple conflicting objectives by means of evolutionary computation techniques (Zitzler 1999; Deb 2001; Coello et al. 2002; Abbass and Sarker 2002). In the last decade, EMO has been a hot research field for solving both theoretical and practical problems. One may find a large number of practical applications of EMO to real-life problems in the books by (Deb 2001; Coello et al. 2002). There are two key concepts in EMO: dominance and the Pareto-optimal set (Deb 2001; Coello et al. 2002). If solution A is not worse than solution B in all objectives and is better than solution B in at least one objective, solution A is said to dominate solution B. If A is better than B in one objective and worse than B in another objective, then A and B are non-dominated solutions. The set of all non-dominant solutions in the feasible region of the search space is called the Pareto-optimal set.

The experiments with the same settings as before are conducted for both random walk and $(1+1)ES$ with two objectives: minimizing the damage of the blue team (Equation 4.16) and maximizing the damage of the red team (minimizing the health of the red team) (Equation 4.17).

$$\Downarrow \quad objective1 = \sum_{i=1}^{N_b} H_i^b - \sum_{i=1}^{N_b} h_i^b \tag{4.16}$$

$$\Uparrow \quad objective2 = \sum_{i=1}^{N_r} H_i^r - \sum_{i=1}^{N_r} h_i^r \tag{4.17}$$

The left column of Figure 4.7 depicts the scatter and Pareto-optimal diagram for all solutions by using random walk. For the GOL and COW scenarios, the Pareto-optimal set is almost the diagonal, which means both blue and red have equal damage. The reason why the low maximum damage of the red team occurs with the COW scenario is that the red agents were running away, therefore there is no engagement between blue and red. The Pareto-optimal set in the BAL and DEF scenarios leans towards the blue team, which implies that in these two scenarios the blue team may cause proportionally more damage to the red team. The VAG and AGG scenarios have similar Pareto-optimal solutions. In both scenarios, the blue team may cause around 75% damage in the red team.

The density of the scatter diagram tells us that solutions found in the BAL, GOL,

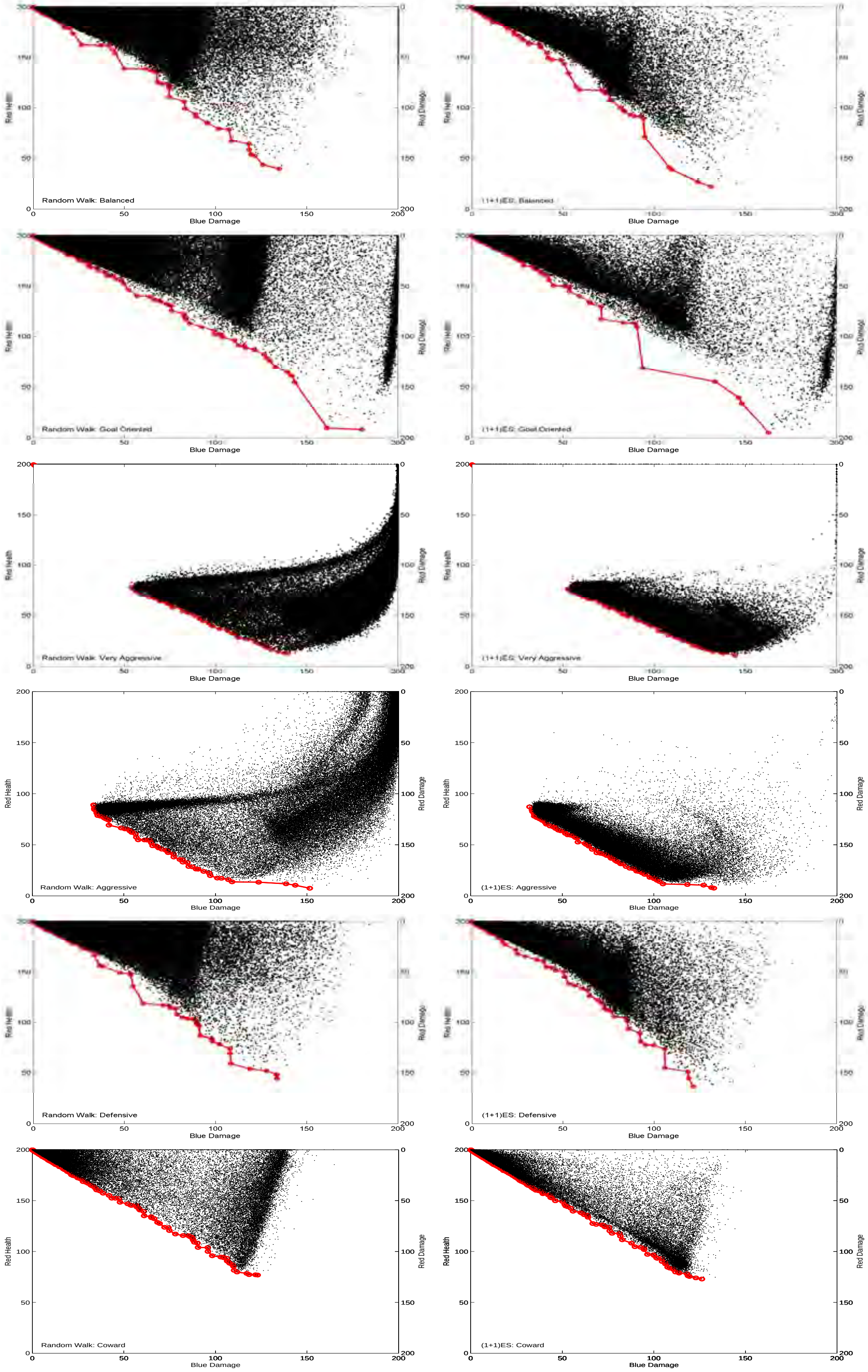


Figure 4.7: Scatter and Pareto-optimal diagram for all scenarios by using random walk (left column) and $(1+1)ES$ (right column)

DEF and COW scenarios frequently appear in the upper left triangle area. Therefore in most cases the blue team finds it hard to find a good solution. However, in the VAG and AGG scenarios, the solution found appear more frequently in the lower left area, which means the the blue finds good solutions easily.

The right column of Figure 4.7 presents the scatter and Pareto-optimal diagram for all solutions by using $(1+1)ES$. When comparing with the left column of Figure 4.7, one may find that both are similar to each other. But it is slightly better in $(1+1)ES$ than that in random walk in terms of the Pareto-optimal set. The density of the scatter diagram reveals that most solutions found by $(1+1)ES$ are quite close to the Pareto-optimal, especially for the VAG and AGG scenarios. In the GOL, VAG and AGG scenarios, solutions can be found where full damage can be achieved to the red team.

4.4 Summary

The results of the experiments were intriguing. The fitness landscape is rugged and multi-modal for all scenarios. The degree of difficulty in finding the right combination of personalities for the blue team is largely dependent on the strategy of the red team. Each strategy changes the fitness landscape of the problem. But the fitness landscapes of the VAG and AGG scenarios are much different from those of the other four strategies. When revisiting the above analysis carefully, one may find that the fitness landscape of BAL is quite similar to that of DEF while the fitness landscape of GOL is quite similar to that of COW. So the six scenarios can be classified into three classes: VAG and AGG, BAL and DEF, and GOL and COW, which are ordered from the lowest degree of difficulty to highest degree of difficulty. If the red team follows the first class of strategies, the blue team may find a good solution relatively easy while if the third class of strategies was taken, the blue team finds it hard to discover a good solution. The second class of strategies is in the middle. By examining the definition of these six scenarios (Table 4.1), one can see that the tendency of clustering with friends or attacking the opponent influenced the fitness landscape more than that of neutrality or evasion. Also, the weights associated with friends have stronger effect on the fitness landscape than those referring to enemies.

Although the results from $(1+1)ES$ is similar to those from random walk, it shows that it is easier to find a good solution in $(1+1)ES$ than in random walk. This implies that exploitative search is better than exploration search in the model of WISDOM-I.

The search for a stable strategy can be misleading in this type of problem. Stability is what the analysts would look for but defining it is challenging. One possibility is to bound the variance; therefore solutions falling in certain interval of minimum and maximum variance are considered to posses the same level of stability.

The results from the multi-objective analysis shows that the composed single fitness function hides many information during the search for a good solution.

During developing and using WISDOM–I, we realized that the limitations are very obvious but critical to the study of CAS. In the next chapter, these limitations are analysed and a new network centric multi-agent architecture is proposed, called NCMAA which may highly improve existing agent architectures.

Chapter 5

NCMAA: a Network Centric Multi-Agent Architecture [1]

5.1 Introduction

The theory of complex adaptive systems (CAS) is the study of many nonlinearly interacting components, where the interaction is governed by simple rules while the overall behavior of the system exhibits a certain level of complexity. The research area of complex system theory encompasses many sub-fields such as *adaptation:* the study of how systems respond to changes in the environment (Holland 1992; Holland 1996; Flake 1998; Levin 1998); *network theory:* the study of how the network topology and properties influence the behaviour that the network exhibits (Wasserman and Faust 1994; Albert and Barabási 2002; Newman 2003; Dorogovtsev and Mendes 2002); and *emergence:* the study of how global phenomena arise from the lower level interaction of the components (Waldrop 1992; Holland 1998; Johnson 2001; Strogatz 2003). All the discoveries in complex system theory are finding their ways to practitioners and real life applications. For example, existing ABDs have already been applied in a number of areas (i.e. education, policy analysis) by a number of

[1] This chapter is based on the publications of Yang et al. (2006a) and Yang et al. (2005a).

organizations (i.e. universities, defence departments, consulting companies) (Lauren and Stephen 2002a; Galligan and Lauren 2003; Ilachinski 2004).

Due to the difficulty faced by analytical methods in analyzing the high degree of nonlinear interactions between components within a CAS, agent based modelling has been widely adopted to model, simulate and study CAS. However some limitations of existing multi-agent simulations have recently been discovered, as follows.

- It is hard to validate these systems because they are representation free (Yang et al. 2005b).
- Reasoning during the simulation becomes difficult with increasing numbers of entities (Yang et al. 2005b). Cognitive agent systems (Barringer et al. 1989; Rao and Georgeff 1991; Fisher 1994; Rao 1996; Lesperance et al. 1996) are able to reason about the actions taken by each entity in the system but unable to scale up to include many agents or account for the high degree of nonlinearity that is featured in most real life problems. On the other hand, pure reactive agent systems (Wooldridge and Jennings 1995; Nwana 1996; Sycara 1998) can scale up well but it is hard to understand the behaviour exhibited by the system or validate it because there is no reasoning.
- Inability to find a common ground between agent-centric or organizational-centric (Vázquez-Salceda et al. 2005) methods. Existing multi-agent systems (MAS) either focus on the model of individual agents with limited support on the interactions between agents such as GAIA (Wooldridge et al. 2000), or concentrate on the model of the agent society by limiting the autonomous behaviours of a single agent, such as SODA (Omicini 2001) and ISLANDER (Esteva et al. 2002).
- Lack of an explicit and auditable model of interaction. Existing systems always combine the entities (agents) and their interactions (relationships) in a single model. There is no distinction between the social value of an entity generated by its interactions with other entities, and the individual value generated by its

own properties and capabilities (Vázquez-Salceda et al. 2005). It is important to have an explicit model of interaction to understand the group behaviours of agents.

In order to address these limitations, a novel Network Centric Multi-Agent Architecture (NCMAA) is proposed. It maps perceived reality to high resolution in the simulation while still able to reason about the actions and the emergent behaviours in the system. NCMAA provides a powerful real time reasoning engine, which is built on network theory and causal models. It helps users to understand the dynamics and outcomes of the simulation by conducting inductive reasoning during the simulation. The method is based on sequential construction of the different elements into a useable model.

The rest of the chapter is organized as follows. The following section is a discussion on what kind of properties of a MAS are essentially required to model and study CAS, based on which a comparison among existing agent based models and NCMAA is presented. After that, NCMAA is described in detail, which includes three sections: the description of the NCMAA, the embedded real time reasoning engine and its developing procedure.

5.2 Multi-agent systems for CAS

MAS is the natural platform for studying CAS. The constituent parts are modelled as agents with a set of pre-defined characteristics. These agents adapt, evolve and co-evolve with their environment (Schmitt 1997; Lauren 2000). By modelling an individual constituent of a CAS as an agent, we are able to simulate a real world system by an artificial world populated by interacting processes. It is a particularly effective way to represent real world systems which are composed of a number of nonlinear interacting parts that have a large space of complex decisions and/or behaviours to choose from such as those situations in combat (Ilachinski 2000).

Therefore a good MAS should be able to model each property of a CAS, and should at least have the following characteristics:

1. High scalability: a CAS usually consists of hundreds or thousands of agents interacting with each other. For example, colonies are made of ants and brains are made of neurons. This normally requires agents to be designed on simple principles.

2. Heterogeneity: obviously a CAS is composed of a number of heterogeneous constituent, e.g. ecosystems. The diversity is essential to maintain an ecosystem.

3. Explicit model of interaction: within a CAS, the constituent parts interact with each other nonlinearly. The global behaviours emerge from these interactions. An explicit model of interaction could help us to study the role of each type of interactions within a CAS.

4. Reasoning on emergent behaviours: the aim of studying a CAS is to understand when, why and how the emergent behaviours occur, and how they link to the simple local rules applied to each individual agent.

5. Reasoning on individual agent: decisions of each individual agent should be based on certain mechanisms in order to take reasonable actions.

6. Rationality: the agent needs to take actions rationally in order to achieve its goals. It may help to avoid exhibit weird behaviours.

7. Adaptivity: the system should be able to improve its performance over time. For example, for a living organism, it may learn from experience and alter its behaviour based on its perception of its environment.

8. Sociality: the overall pattern of a CAS is based on interactions among a group of agents. Therefore an organisation model is required to understand the group behaviour.

Table 5.1: Comparison of agent architectures

Essential characteristics	Cognitive	Reactive	Hybrid	**NCMAA**
Scalability	low	high	medium	**high**
Heterogeneity	yes	yes	yes	**yes**
Explicit model of interaction	no	no	no	**yes**
Reasoning on emergent behaviours	no	no	no	**yes**
Reasoning on individual agent	yes	no	yes	**no**
Rationality	yes	no	yes	**yes**
Adaptivity	yes	yes	yes	**yes**
Sociality	no	no	no	**yes**
Credibility	yes	yes	yes	**yes**

9. Credibility: the overall MAS should be transparent in its construction and demonstrably "fit for purpose" in its applicability to understanding and design of a CAS.

Table 5.1 is a comparison among existing agent architectures and the proposed requirements for NCMAA based on the essential characteristics of a good MAS for CAS. All three existing agent architectures have a limited capability to study the role of each interaction and to understand emergent behaviours and patterns. As discussed in chapter 2, existing agent architectures focus on modelling individual agent. Interaction among agents is embedded in this individual agent model. They do not have an explicit model of interaction to facilitate the study of the role of interactions in a system. Since they focus on the model of single agent, it is hard to use them to capture the sociality of agents. As well, reactive agents do not have any reasoning while cognitive agents only reason their actions. Both of them do not have any reasoning mechanism to interpret emergent behaviours of the system. The proposed multi-agent architecture, NCMAA, attempts to address these drawbacks. The architecture capitalizes on the large literature existing in the area of network theory.

5.3 NCMAA: a network centric multi-agent architecture

The proposed NCMAA architecture is mainly based on network theory. The system is designed on the concept of networks, where each operational entity in the system is either a network or a part of a network. The engine of the simulation is also designed around the concept of networks. Figure 5.1 depicts a coarse-grained view of the system. Each type of relationship between the agents forms a network. Two types of network are defined: the static network, where the topology does not change during the course of the simulation, and the dynamic network, where the topology changes over time. As an example, the former can be the network of families, while the latter can be the communication network. It is important to emphasize that the definition of static or dynamic may vary from one application to another. The decisions of which actions should be taken by the agents are completely constrained by the state of networks and environment. The actions taken by the agents may trigger a change in agents' states, environmental states, or simply the simulation clock advances. These triggers affect the dynamic relationships and the cycle continues.

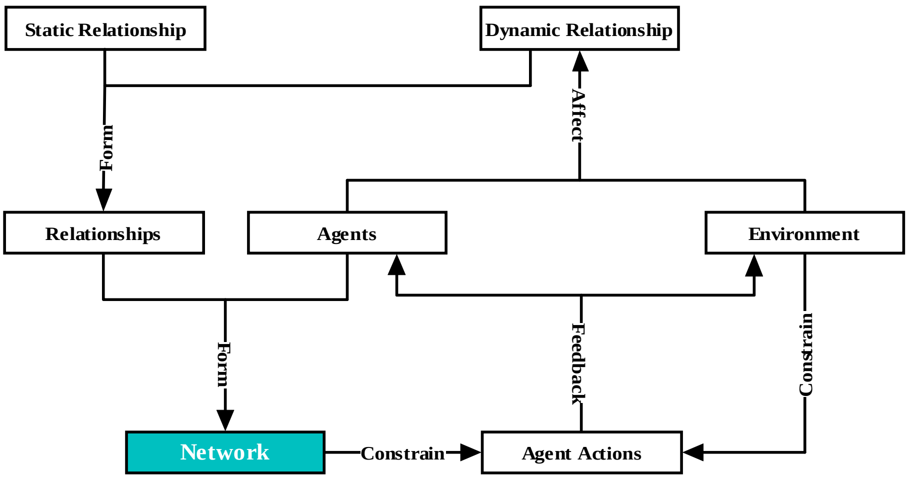


Figure 5.1: A coarse-grained view of NCMAA

One needs to go through two stages to design a system based on NCMAA: developing a causal network among concepts, and designing the finite state machine which will control the simulation oracle.

In NCMAA, a concept means a type of relationship between agents. The causal network is a directed graph of concepts defining the interdependency of concepts in the concept space. It provides the basis for establishing a meta-level reasoning system. A causal network usually has sources and sinks (showing the boundaries of the system being analyzed). The finite state machine is a collection of states, each representing the state of a network in the system. The finite state machine represents the sequence of executing each network in the system and the control of the system clock.

The overall architecture of the NCMMA is a two–layer architecture (see Figure 5.2). In the top conceptual layer, the causal network defines the different types of relationships among agents in the system and how one type of relationship influences other types. The bottom implementation layer defines the instances of each concept defined in the causal network. For example, communication may appear as a node in the causal network. At the lower level, there can be many instances of communication such as P2P, broadcasting, etc. An agent can use both P2P and broadcasting to communicate with other agents. So each agent may participate in the different networks and play several different roles in the system.

Each agent in NCMAA is modelled by a series of states, triggers, actions and consequence. An agent state is defined by a series of properties which are problem specific. If an agent is in a certain state, the trigger is activated and an action is taken by the agent. It may lead to its own state changed, other agents' states changed or environmental states changed. Different actions may result in different consequences. Figure 5.3 shows a generic finite state machine for a network. Each network state includes agents' states and environmental states and is the constraint for agents to take actions. At time t, an agent selects and executes an action which causes a transition from one or more of the network states to another.

The simulation engine in NCMAA is depicted in Figure 5.4. First the system is

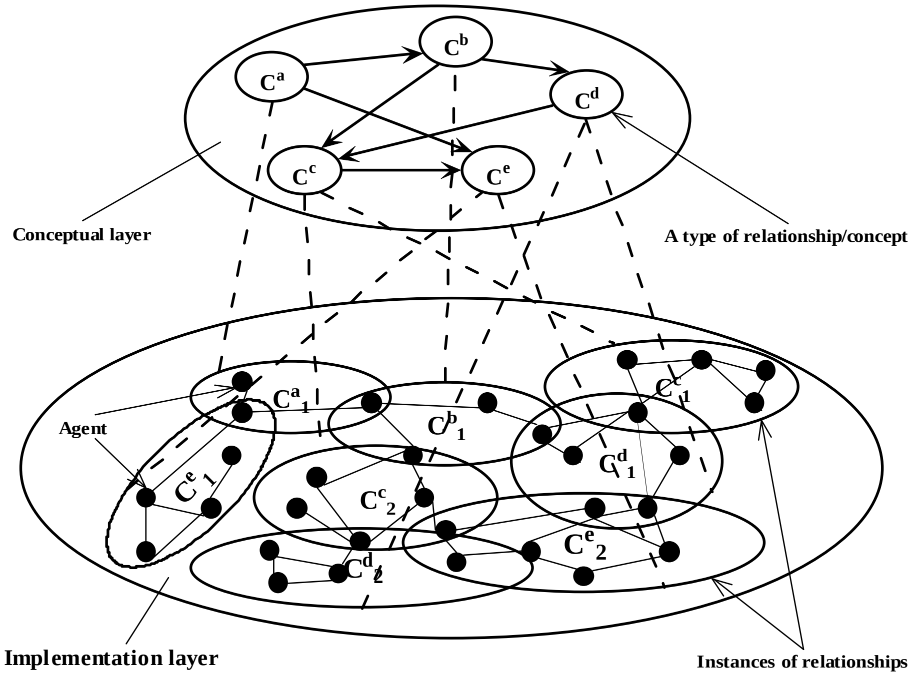


Figure 5.2: The two-layer architecture in NCMAA

initialized and the clock is set to zero. The initialization triggers formation of some networks, agents sense their environment, recognize their internal states, and based on networks' state and internal motivations, take actions which may cause a change in the system. The simulation cycles continue till termination occurs. The system terminates if either the maximum number of simulation time steps is reached or a pre-specified condition is satisfied.

5.4 Causal model and network based reasoning

5.4.1 Causal model

Causal knowledge is widely used to predict future events, to interpret the occurrence of present events, and to achieve objectives by taking actions. To obtain correct causal knowledge is not an easy task, e.g. how to distinguish the causal relations from spurious relations. Recent researches (Waldmann 2000; Waldmann 2001) show

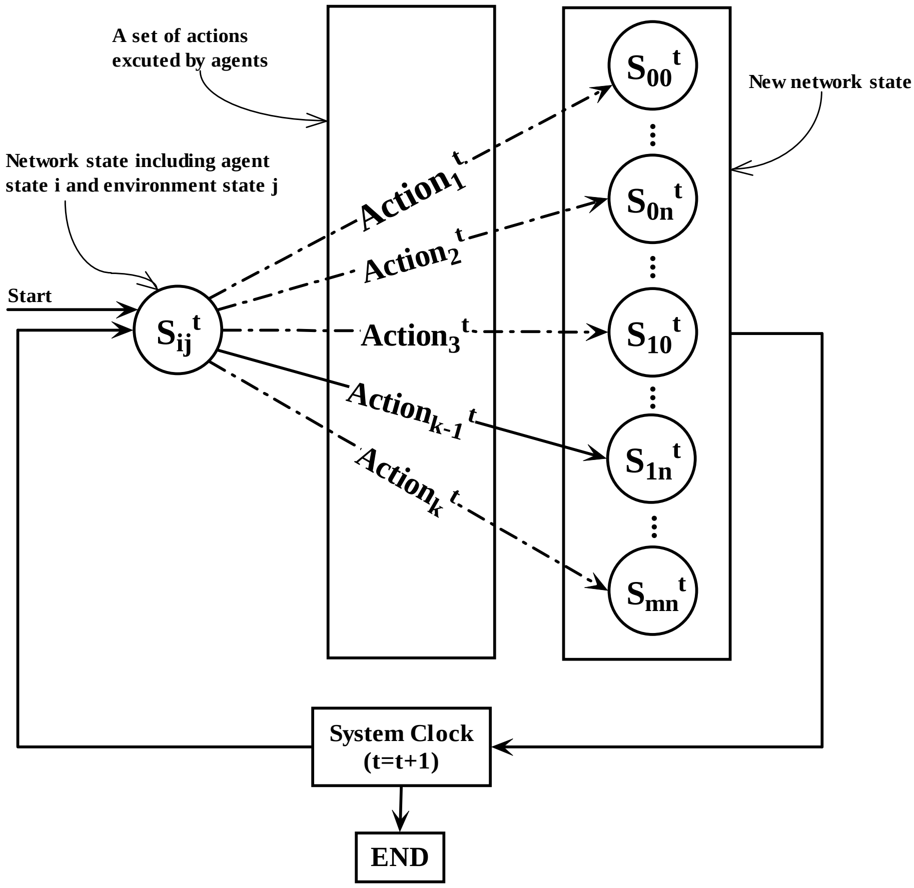


Figure 5.3: Generic finite state machine in NCMAA

that causal-model theory (Pearl 2000) is a very complete approach to deal with this kind of problem. Causality is characterized by its structure and strength (Lagnado et al. 2006). A causal structure concerns the qualitative causal relationships between two events while causal strength concerns the quantitative aspect of the causal relations. Conceptually the structure is more basic than the strength. One must know the existence of a link before estimating the strength of it. Otherwise, one may make wrong predictions, understandings and decisions. This is the reason why NCMAA requires one to build a causal structure (causal network) first based on prior domain knowledge about the causal status of relationships, which forms the direction of the causal arrow within causal models and then a series of covariation information is collected to estimate the strength of each causal relation (Waldmann 1996). The way covariation measures are computed and interpreted is totally based

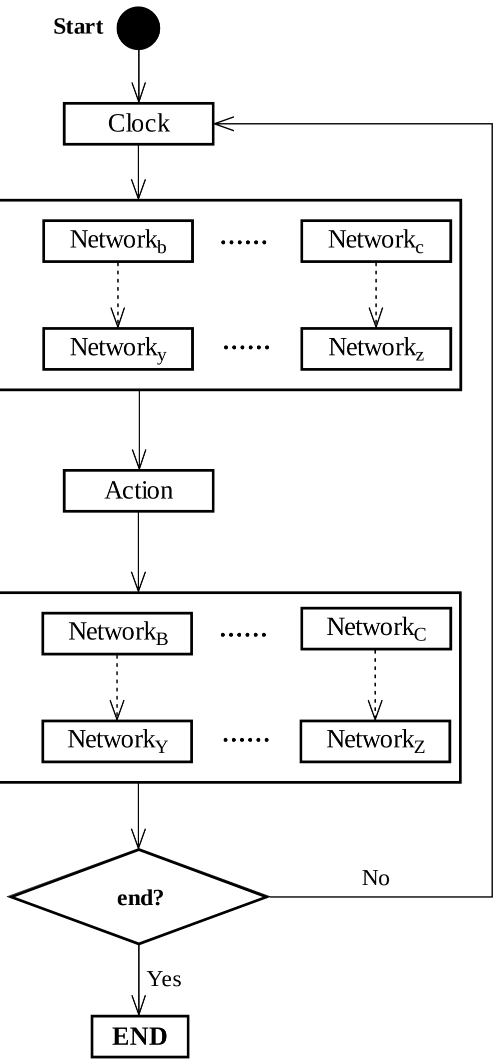


Figure 5.4: The simulation engine in NCMAA

on the causal structure.

Formally a causal structure (Pearl 2000) of a set of variables V is a directed acyclic graph (DAG) where each vertex corresponds to a distinct variable in V, and each link represents a direct functional relationship between two variables. It can also be represented by a set of structure equations:

$$I_i = f_i(P_i, \epsilon_i), \qquad i = 1, ..., n, \tag{5.1}$$

where P_i stands for the set of interactions judged to be immediate causes of I_i and the ϵ_i represent errors due to some omitted factors. Equation 5.1 is a nonlinear formation of the linear structural equation models (SEM), a dominant tool used for causal analysis in economics and social sciences (Pearl 2000).

A causal model is defined as a pair $< D, \Phi_D >$ where D is a causal structure in the model and Φ_D is a set of parameters compatible with D. The parameters Φ_D develop a function of the equations defined in Equation 5.1 to each $I_i \in V$ and a probability measure $P(\epsilon_i)$ to each ϵ_i.

In NCMAA, the causal network representing the causal structure is a DAG of a set of interactions (relationships) between agents. Each node in the causal network represents a type of interaction while each link represents how a type of interaction is influenced by other types of interactions. After having the causal network which forms the top layer in NCMAA, a series of statistical analyses could be conducted to collect covariation information from the bottom layer in NCMAA to estimate the strength of each causal relationship.

5.4.2 Network theory

Since each node in the causal network is a type of interaction between agents which is represented as a network or a set of networks at the bottom layer in NCMAA, these networks at the bottom layer can be analysed by means of network theory to characterize or qualify each node. Network analysis techniques have been widely adopted to capture the dynamics, mechanism and patterns of complex systems across many disciplines of social science, economics, defence, biology and ecology. Network theory focuses on the relationships, structural or relational processes among social actors. It provides a number of network measures to characterize the structure or the pattern of complex networks (Wasserman and Faust 1994; Albert and Barabási 2002; Newman 2003; Dorogovtsev and Mendes 2002).

5.4.2.1 Size of the network

1. The number of agents in the network (n)
2. The number of edges in the network (E)

5.4.2.2 Average path length

Average path length (Mean Geodesic Distance) (l) means the average shortest distance between two agent in the network.

$$l^{-1} = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i \geq j} d_{ij}^{-1} \tag{5.2}$$

where d_{ij} denotes the shortest distance between agent i and agent j. The distance between two unconnected agents is infinite. Average path length offers a measure of a network's overall navigability. The largest path length is called diameter of the network.

5.4.2.3 Degree (connectivity) and Degree distributions

The degree k_i means how many agents connect to agent i. The average degree of the network ($\overline{k}$) is:

$$\overline{k} = \frac{1}{n} \sum_{i} k_i \tag{5.3}$$

p_k is the fraction of agents in the network that have degree k:

$$p_k = \frac{N_k}{n} \tag{5.4}$$

where N_k is the number of agents with k degree. It is also the probability that an agent chosen uniformly at random has degree k. The degree distribution of the network is the histogram of the degree of agent.

In order to clearly see the distribution, we may use cumulative distribution function:

$$P_k = \sum_{k'=k}^{\infty} p_{k'} \tag{5.5}$$

which is the probability that the degree is greater than or equal to k.

In a directed network the degree is usually divided into the in-degree and the out-degree. The in-degree of an agent A is the number of edges with A as their terminal vertex while the out-degree of an agent A is the number of edges with A as their initial vertex.

5.4.2.4 Transitivity or Clustering

Clustering coefficient (C) measures the density of triangles in a network. It is also called network density. The clustering coefficient of agent i is:

$$C_i = \frac{2E_i}{k_i(k_i - 1)} \tag{5.6}$$

where E_i is the number of edges between agents which are connected to agent i and k_i is the degree of agent i.

The clustering coefficient for the network is defined as:

$$C = \frac{1}{n} \sum_i C_i \tag{5.7}$$

which characterizes the overall tendency of agents to form clusters or groups.

An important measure of the network's structure is the function $C(k)$, which is defined as the average clustering coefficient of all agents with k degree, which is an indication of a network's hierarchical character:

$$C(k) = \frac{1}{n} \sum_{i} \frac{2E_i^k}{k(k-1)} \tag{5.8}$$

where E_i^k is the number of edges between agents connected to agent i which has k degree.

Based on network theory and the knowledge of the causal relationship among the concepts from the causal mode, a real-time networked based reasoning can easily be established.

5.4.3 Real-time network based reasoning

Dynamics and patterns of a complex network determine many features of complex systems. Analysis of network measures may help analysts to understand how global organization and behaviour emerges from local interactions, and then to plan or manipulate the behaviours emerging from the networks. There are various statistical or other types of analyses which can be applied to these network measures, such as time series analysis. The reasoning engine in NCMAA operates on this latter analysis.

The state x_i of a system is often defined as follows (Shalizi 2004):

$$x_{i+1} \equiv F(x_i, i, \mu_i) \tag{5.9}$$

where the function F depends on the time index i and a sequence of independent random variables μ_i. Under most circumstances, the evolution of x is Markovian (Shalizi 2004), i.e the future state is totally dependent on the current state and the earlier states are irrelevant. However, normally it is almost impossible to directly observe the state x. Instead what can be observed is a measure y, which is generally a noisy, nonlinear function of the state x (Shalizi 2004):

$$y_i = f(x_i, \phi_i) \tag{5.10}$$

where ϕ_i is a sequence of measurement of noise. Unfortunately we usually do not know the observation function f and the state-dynamics function F. The goal of time series analysis is to make a reasonable guess about these functions so as to predict and better understand the evolution process of the state. Since raw time series data (y) often contain noise and other obscuring factors, a common approach adopted to gain better understanding of the dynamics involved in the raw time series data is to extract features which fall into two categories: the time-domain which capture temporal properties and the frequency domain which capture the spectral properties (Shalizi 2004). Typically no single measure can fully describes a time-series. Therefore a set of measures is usually adopted, such as mean, median, mode, variance, correlation coefficient, linear distance and auto-correlation distance. In order to conduct real-time analysis and reasoning during the simulation, the data of network measures, performance measures or any other effect measures have to be streamed and fed into a windowing engine which in turn produces a series of data frames with different window sizes. A number of analyses can then be done within a single data frame or between data frames, such as Granger causality test, path analysis and root cause analysis.

5.4.3.1 Granger causality

Granger causality was introduced by Granger (1969). It is a technique for determining causal relationships between variables and built on the assumption that the information for the prediction of the value of a variable is embedded only in the time series of the causal variables and itself. A time series x_t Granger-causes another time series y_t if the future values of y_t can be predicted by the values of x_t. The idea is that the cause must appear before the effect is materialized. Thus if x_t causes y_t, then the values of x_t must be helpful for forecasting the values of y_t. Formally, let x_t and y_t be two stationary processes, and $E(y_{t+1}|\Phi_t)$ be the optimal (minimal mean squared error) predictor of the process y_t. Then x_t Granger-causes y_t if

$$E(y_{t+1}|(x_t \cup y_t)) \neq E(y_{t+1}|x_t)$$

and x_t Granger-causes y_t instantaneously if

$$E(y_{t+1}|(x_{t+1} \cup y_t)) \neq E(y_{t+1}|x_{t+1})$$

In practice, if the prediction error of the values of y_t is reduced by including measurements of x_t in the regression model, then x_t is said to be able to causally influence y_t. The standard Granger causality test usually involves a regression test as in Equation 5.11:

$$y_t = \sum_{i=1}^{k} \alpha_i y_{t-i} + \sum_{j=1}^{k} \beta_j x_{t-j} + u_t \tag{5.11}$$

Equation 5.11 suggests that the value of y at time t is a function of the past values of y_t itself as well as of x_t. Generally, if x_t Granger causes y_t, then any changes happening in y_t should be after some changes occurring in x_t. Therefore, if the prediction of y_t can be significantly improved through a regression of y_t on other variables, including its own past values and the past or lagged values of y_t, then we can say that x_t Granger causes y_t. The tests of Granger-causality can also be based on a vector autoregressive model, a multivariate MA (Moving Average) representation or a regression of y_t on $x_t \cup y_t$ (see Hamilton (1994) for a review of such tests).

Since it is based on correlation rather than causation, Granger causality can be led to a spurious correlation. However in NCMAA, time series and correlation analysis are not conducted in an ad hoc manner. Instead, such kind of analysis is totally based on a pre-developed causal network. Therefore it may easily avoid spurious correlations and find the real causal relationships between variables. The Granger causality test has been widely used in different research fields, such as defence (Dunne and Vougas 1999; Dunne et al. 2001; Kollias et al. 2004) and economics (Hiemstra and Jones 1994).

5.4.3.2 Path analysis

Path analysis (Harris 2001) is an extension of the regression model, used to validate the causal relationships between two or more variables. The model is usually presented as a graph where arrows indicate causal relations pointing from cause to effect. A regression is conducted between each dependent variable and its causes in the model. That is, the regression is done between every pair of variables which are connected by an arrow. The regression weights or path coefficient indicates the degree of the effect of a variable assumed to be a cause on another variable assumed to be an effect. Normally there are four steps for path analysis:

1. Build a causal model (hypothesis) that all variables are causally connected. Variables that have no explicit causes are called exogenous variables while variables caused by other variables are called endogenous variables in the model.
2. Select measures of these variables and translate the causal relationships to a series of structured equations.
3. Calculate statistics, e.g. the path coefficient, to show the strength of relationship between each pair of causal variable and effect variable in the causal model.
4. Interpret statistics to see if they support or repute the assumption in the causal model.

The path coefficient used in the path analysis is a standardized regression coefficient, which represents the direct effect of an exogenous variable on an endogenous variable in the path model. Thus when there are two or more endogenous variables in the path model, the path coefficients are partial regression coefficients which measure the extent of effect of one variable on another in the path model. If in a path model, a variable is dependent only on a single exogenous variable, the path coefficient in this special case is a zero-order correlation coefficient.

5.4.3.3 Root cause analysis

Root cause analysis (RCA) (Nelms 2003; Rooney and Heuvel 2004) is a methodology for discovering the underlying important reasons for performance problems. It is designed to help identify what, how and why an event occurred. Only when the underlying reasons why an event or failure happened are determined, the workable corrective countermeasures can then be established to prevent that event or failure from occurring again. RCA attempts to dig below the symptoms, and investigates the fundamental, underlying reasons (root causes) leading to the undesired consequences while troubleshooting and problem solving try to find immediate solutions to resolve the user visible symptoms. RCA traces the cause and effect trail from the end effect back to the root causes. To solve a problem, the best option is to kill its causes at the root. Rooney and Heuvel (2004) proposed four steps to conduct RCA:

1. Data collection: all necessary data describing the failure or event has to be collected.
2. Causal factor charting: it is a sequence diagram showing all direct and indirect events leading to an occurrence of the failure and can be represented as a causal model discussed above.
3. Root cause identification: this is the key process in RCA. Based on the causal model or the causal factor charting, some analysis techniques, such as a deci-

sion diagram called the Root Cause Map, Granger causality test and correlation analysis, can be employed to identify the root causes.

4. Recommendation generation and implementation: the countermeasures are produced to prevent the failure from occurring again. Normally the root cause analysts do not implement these recommendations.

Although it is often used to identify the causes of a problem or failure, it by no means can not be used to find the underlying reasons of an event if the event is seen as an undesired behaviour. RCA provides critical information on what to change and how to change it. Therefore RCA may help improve the system performance and also may help attack the system. Currently it is widely used in manufacturing, construction, healthcare, transportation, chemical industry, networking, software engineering and power generation (DOE 1992; Tuli and Apostolakis 1996; Fernandes et al. 15; Handley 2000; Leszak et al. 2000; Rex et al. 2000; Boyer 2001; Burroughs et al. 2000; Thornhill et al. 2001; Carlson and Söderberg 2003; Julisch 2003; Zhang et al. 2004; Siekkinen et al. 2005).

5.5 System developing procedure

In practice, of course, NCMAA is an architecture that allows and indeed encourages flexibility in the way in which the system is modelled and implemented. Implicit in this is the notion that any system description should balance, or at least acknowledge, the competing paradigms of "correctness" and "usefulness". "Correctness" - based on an ontological premise of "the system is" is likely to produce overly complicated descriptions of systems that are not properly understood. This will especially be the case where humans are involved as in the military example given in this thesis. The usual adage of the Operations Researcher (OR) then comes into play - we need "usefulness" through reasonable epistemological abstraction of how we see the system. A four phase, 12 step (see Figure 5.5) procedure is proposed to achieve this. After each phase or step, the developer may need to review the current design and perhaps go back to a previous step again.

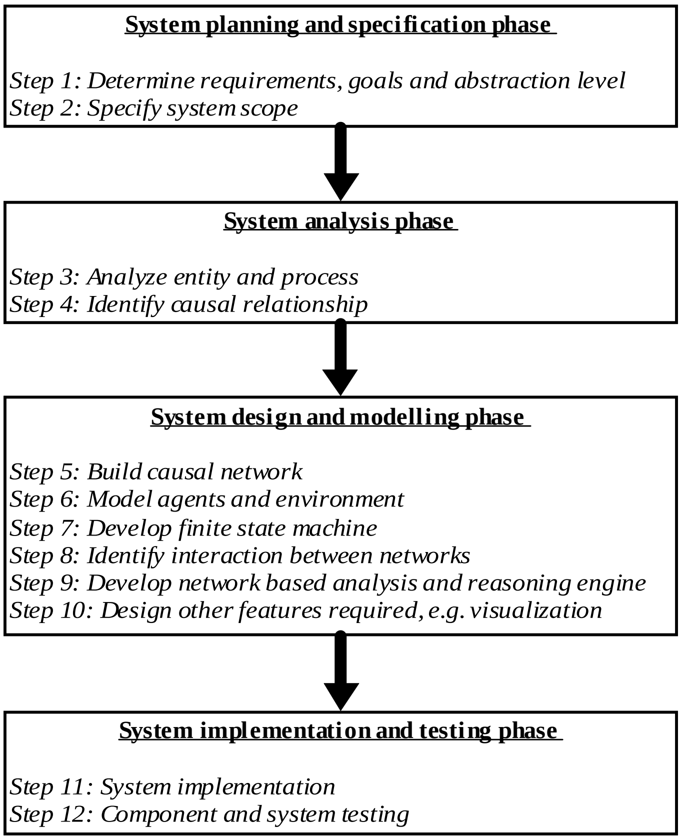


Figure 5.5: Generic procedures for developing a new application based on NCMAA

5.5.1 System planning and specification phase (step 1 – 2)

In the first phase the developer needs to establish a high-level view of the intended system which includes the goals, the level of abstraction and the requirements of the end-user. Then the developer needs to specify what kind of problem the system will deal with and the boundary of the problem. This is very much a creative phase and as such needs to be transparent to show the reasoning behind the formulation of the issues. In Operations Research (OR) terms, this phase relates to the formulation aspects of Ackoff "problems" or "messes" (Ackoff 1979a; Ackoff 1979b). Unlike "puzzles" that are usually unambiguous with clear objective functions, "problems" and "messes" tend to have degrees of freedom in how they are approached. Thus the emphasis is on agreement with potential stakeholders on the validity of the model. There is a range of problem structuring methodologies that could be used in this phase (Curtis et al. 2005), and these are sometimes referred to as "soft OR" given that there are usually degrees of freedom available in how any problem may be tackled. As a simple list this formulation phase should include the context, an understanding of the processes involved in the system under study, the dynamic interactions, auditable option identification and viable measures (Curtis et al. 2005).

Success in this phase requires both domain expertise in the processes involved and in the methods of formulation of the issues.

5.5.2 System analysis phase (step 3 – 4)

In this phase, the developer examines the system under study to derive an analytic formulation of the constituent processes. Thus models of the sequences of actions in the various processes are discovered, the entities involved and the data that describe their operation, the "concepts" (do P, by means of Q to achieve R (Dortmans et al. 2005)) of each action and the cause-and-effect relationship between the processes. Again domain expertise and skills in soft OR need to be involved in this phase. A critical step to include is the notion that the analytical description still retains enough credibility as a system when the different processes are linked in a synthetic manner.

5.5.3 System design and modelling phase (step 5 – 10)

The next phase involves putting flesh on the abstract ideas determined in the previous two phases. This is very much an engineering activity and comprises an agent model, an environment model, a network model and a reasoning model for each process and the overall interactions. A first cut verification examination needs to be done at this phase - does the coding adequately represent what occurs in the system

(Caughlin 2000)? If difficulties in representation occur, then it might be appropriate to return to phase 2 to determine what can be investigated.

5.5.4 System implementation and testing phase (step 11 – 12)

At this phase, the developer adopts a proper programming language and software engineering techniques to implement the system. Although an object-orientation language is not required, it is preferable by its nature. According to the results from the design, the developer needs to identify classes including attributes, methods, events and exception handling, components and subsystems. Three major aspects are very crucial and needed to be well addressed: control structure, algorithm and data structure (Pfleeger 1998).

Another important task in this phase is to test the system. The testing procedure includes validation and a detailed verification process. Verification is the process of determining whether a model is accurately implemented according to the system description and specifications (Caughlin 2000). Validation is the process of determining whether the model is accurately representing the real world from the perspective of the end-user of the model (Caughlin 2000). There exists a number of models for validation and verification (Balci 1994; Sargent 1999; Caughlin 2000; Coylea and Exelby 2000; Walton et al. 2001). One example is the model proposed by Sargent (Sargent 1999). Sargent's approach consists of four components (see Figure 5.6): conceptual model validity, which is integrated into the system planning and specification phase, system analysis phase and system design and modelling phase, computerized model verification, which is integrated into the system implementation step, operational validity, which is integrated into the system testing step, and data validity, which is used for all phases.

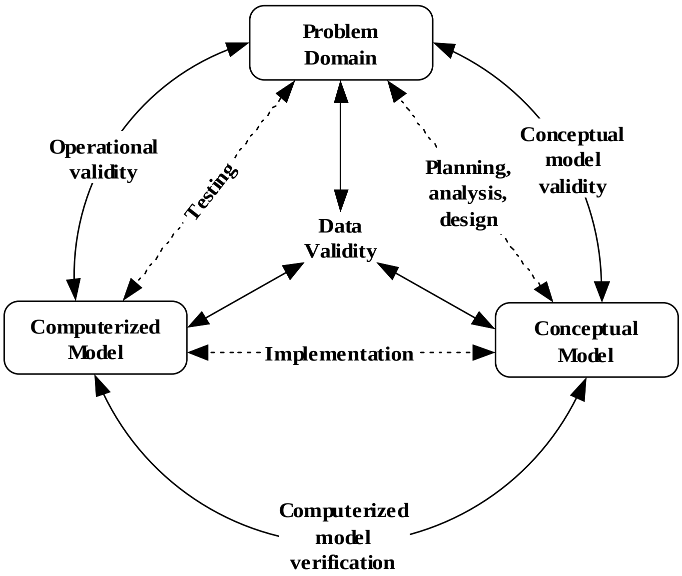


Figure 5.6: Sargent's validation and verification approach

5.6 Summary

MAS is a popular tool adopted to model, simulate and study CAS. Eight essential characteristics are proposed in order to be a good MAS for CAS: high scalability, heterogeneity, explicit model of interaction, reasoning on emergent behaviors, rationality, adaptivity, sociality and credibility.

By examining these essential characteristics, limitations are found in existing agent architectures. To address these problems, NCMAA is proposed, which is based largely on network theory. The system is designed on the concept of networks, where each operational entity in the system is either a network or a part of a network. Each type of relationship between the agents forms a network. The decisions of which actions should be taken by the agents are completely constrained by the state of networks and environment.

A two–layer architecture is adopted in NCMAA. In the top conceptual layer, the causal network defines the different types of relationships among agents in the system and how one type of relationship influences other types. It provides the basis for establishing a meta-level reasoning system. The bottom implementation layer defines the instances of each concept (relationship) defined in the causal network.

NCMAA provides a powerful real-time reasoning engine, which is built on network theory, causal models and various statistical analysis. It helps users to understand the dynamics and outcomes of the simulation by conducting inductive reasoning during the simulation. The method is based on sequential construction of the different elements into a useable model. In order to conduct real-time analysis and reasoning during the simulation, the data of network measures, performance measures or any other effect measures have to be streamed and fed into a windowing engine which in turn produces a series of data frames with different window sizes. A number of analysis can then be done within a single data frame or between data frames, such as Granger causality test, path analysis and root cause analysis.

In practice, a four phase, 12 step developing procedure is proposed to help user to build systems based on NCMAA. After each phase or step, the developer may need to review current design and perhaps go back to a previous step again.

Despite that many agent systems can be imagined as being operating on the concept of networks, designing and implementing the system on the concept of network is a more powerful approach.

The proposed agent architecture NCMAA provides for the first time a new type of reasoning in CAS. The network based architecture facilitates structural reasoning, creates a new angle to gain insight into a complex system and easily overcome the limitations associated with current agent based simulations. In the next chapter, version II of WISDOM is developed based on NCMAA and used to study warfare simulations.

Chapter 6

WISDOM version II [1]

6.1 Introduction

Based on interpreting warfare as a CAS (Ilachinski 1997; Ilachinski 2000; Lauren 2000; Scherrer 2003; Ilachinski 2004), a number of reactive agent based distillation systems have been developed and facilitated the analysis and understanding of combat. From an analysts point of view these systems allow rapid examination of a very large environmental space leading to identification of high and low points and areas of instability. However, all these combat systems are based on the reactive agent architecture and lack reasoning. Accordingly, the version II of WISDOM (WISDOM-II) is developed on the NCMAA.

WISDOM-II not only introduces a number of new features, but also provides a powerful real-time reasoning engine which allow the analysts easily to capture what is going on, why it is happening and how it is happening during the simulation, and a series of visualization tools which make it possible to easily verify and validate the system itself.

[1] This chapter is based on the publications of Yang et al. (2006c), Yang et al. (2006d), Yang et al. (2005a), Yang et al. (2005b), Yang et al. (2005c) and Yang et al. (2005e).

In this chapter, WISDOM-II is described through each step in the development procedures of NCMAA discussed in the previous chapter.

6.2 System development

6.2.1 Determine system goals, requirements and abstraction level

6.2.1.1 The goals

WISDOM-II attempts to be a decision making aid for defence analysts to explore a large parameter space, to conduct scenario analysis and planning, and to answer "what if" questions that allow investigation of concepts and development of capability options in a short period of time. Like other existing combat distillation systems, WISDOM-II is a complement to the very high-detailed simulation systems, which do not allow for the examination of a very large mount of possibilities and outcomes because of their very high fidelity.

6.2.1.2 The requirements

In order to achieve its goals, WISDOM-II should:

- represent the system under study in a credible and useful manner;
- be able to rapidly modify entity characteristics and behaviors: this may allow defence analysts to study the effect of the characteristics of entities on the force performance and allow different options to be examined;
- be able to capture emergent behaviors: the process in combat should be driven by a set of simple rules and some "surprises" may come out from them. Then defence analysts may backtrack to see how these new behaviors emerge;

- be able to interpret the simulation outcomes;
- be able to generate data based on meaningful metrics for military analysis;
- be amenable to rapid, repeatable concept exploration;
- be simple to use; and
- be extendable to a range of scenarios.

6.2.1.3 The level of abstraction

Since WISDOM-II is an exploration tool for concepts, doctrine, and capability requirements in military operations, it has to be a high abstract conceptual model of combat. Like existing ABDs, WISDOM-II is a low fidelity, effect-based simulation system, which only models the essential conceptual entities and processes without touching on any detailed physics of combat. The outcomes of the simulations may guide further analysis with other high fidelity models.

6.2.2 Specify system scope

There are three levels of military operations: strategic, operational and tactical (DOD 2005):

- Strategic level: Activities at this level usually include the development of national and multinational military objectives, and development of global plans or theater war plans to achieve these objectives.
- Operational level: Activities at this level connect tactics and strategy by developing operational objectives needed to accomplish the strategic objectives. The logistic and administrative support are at this level.
- Tactical level: Activities at this level attempt to win a small-scale conflict by focusing on the ordered arrangement and maneuver of combat elements.

WISDOM-II mainly explores the problem space at the tactical level. Each simulated team/force in the system has a predefined goal and tries to achieve it. WISDOM-II is designed for land combat because of its complex environment, although abstract aircrafts and maritime vessels can be modelled in the system. Since it is a conceptual model, there is no physics modelled in the system and use 2-D environment without elevation input.

6.2.3 Analyze entity and process

In traditional analysis of combat, normally there are two forces playing against each other. In our model of the system for each side there is a dominant command and control (C2) structure that uses vision and communication networks that create a situation awareness network, that may in turn direct the engagement network. Figure 6.1 depicts the C2 hierarchy in WISDOM-II. Each force may have several teams, each of which may include several groups. Each group may have a number of agents with different characteristics. Heterogeneous agents at the group level is first introduced in WISDOM-II. Each team has a mission to achieve. The high level commander may send commands to a low level combatant based on its situation awareness and the mission of that combatant. Each agent will follow the OODA (Observe-Orient-Decide-Act) loop model introduced by Boyd (Coram 2002):

- Observe: Scan the environment and collect information from it, including communicating with other agents. In WISDOM-II, each agent has sensors and communication channels it can use. At each time step, the agent scans the environment and communicates with other agents from the same team. The details of the sensor and communication model will be discussed later.
- Orient: Build up its situation awareness based on information collected. As more information is collected, the agent modifies its situation awareness. This is the process to transfer data into information and then into knowledge. In WISDOM-II at each time step, each agent fuses the information via sensor

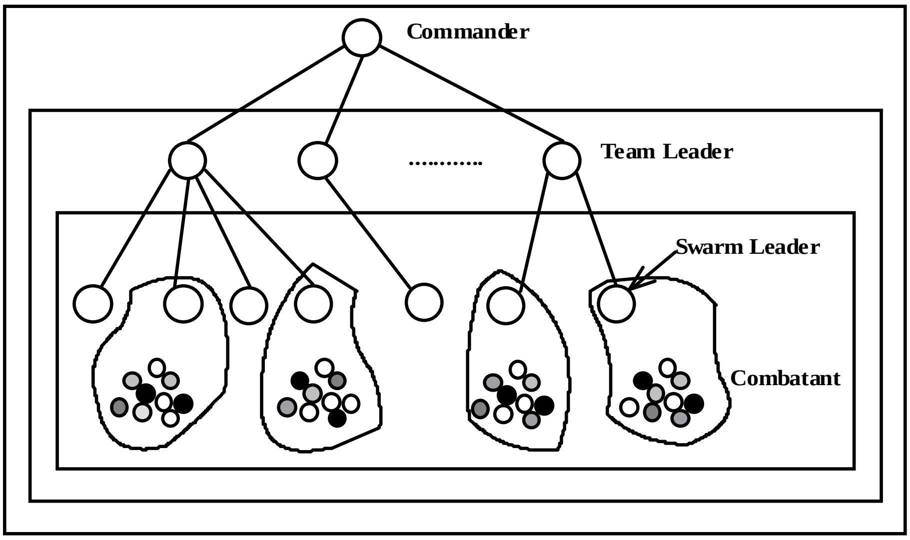


Figure 6.1: The C2 hierarchy in WISDOM-II

and communication with the old situation awareness and construct its new situation awareness. WISDOM-II firstly introduces the situation awareness at the agent level.

- Decide: Consider options and select actions. For a combatant, it needs to decide whether it needs to fire at its enemy, which enemy it will fire at and which direction it will move to. For a commander, it needs to make plans to achieve the mission and send commands to the low level agents. WISDOM-II adopts a heuristic model to make firing decisions. The enemy with short distance has high probability to be fired at. Two decision making mechanisms (Yang et al. 2005b) are used in WISDOM-II to make plans for commanders and movement decisions for combatants.

- Act: Execute the chosen actions. The position of agents and status of networks are updated to reflect the results of actions.

The simulation runs as depicted in Figure 6.2. The system first initializes agents, networks and environment based on the scenario definition XML input file. Then the agents scan their environment and the system updates the vision network. Based on the vision network and C2 network, the agents communicate with all possible

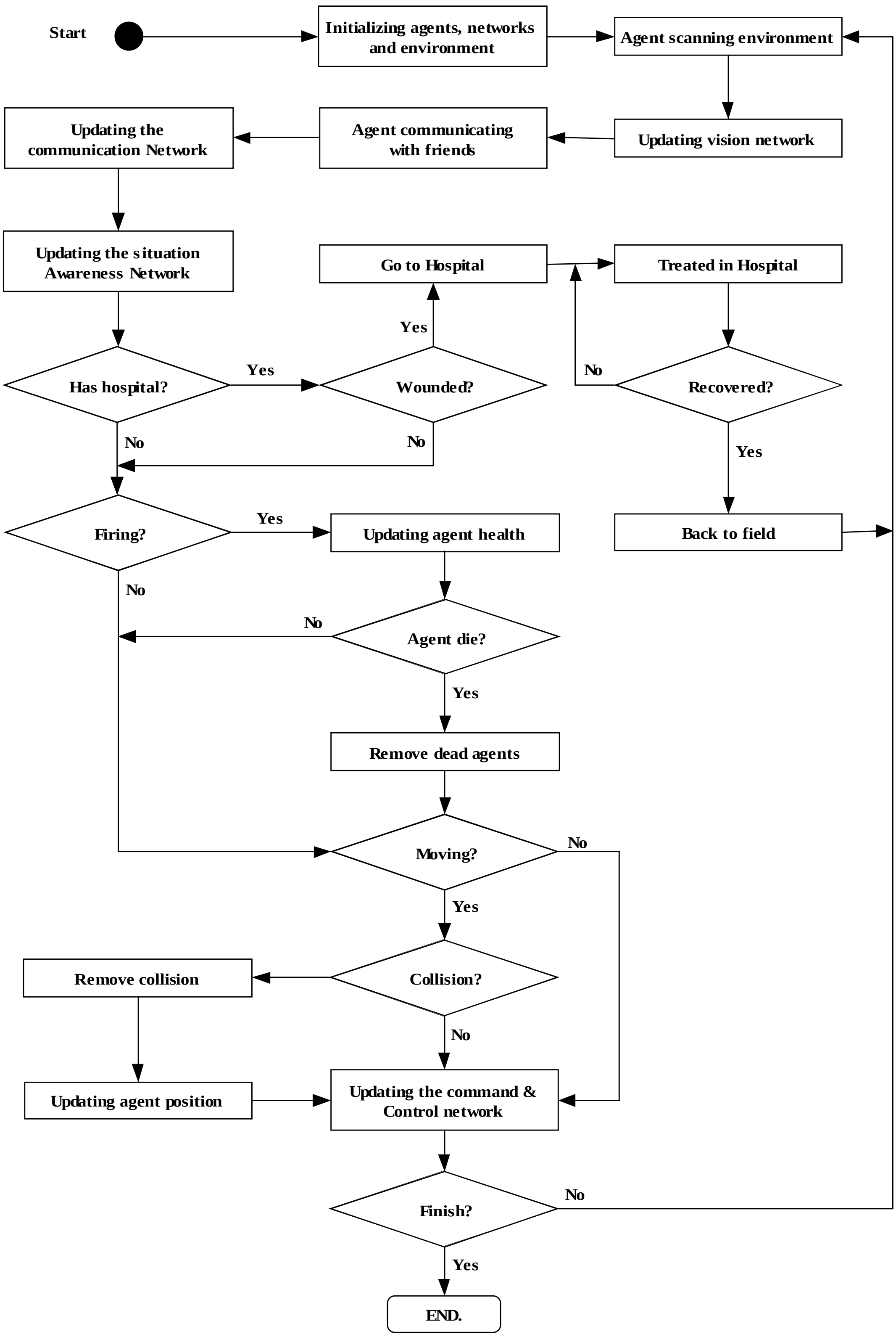


Figure 6.2: The simulation process

friends and the communication networks are updated. The updated vision and communication network then construct the new situation awareness network. After that, the agents make decisions, e.g. fire at their enemy, move to new place or stay at current place, according to their missions and situation awareness. Finally the system updates the C2 network since the agent may die.

Of course in modern conflict there may well be other players present. For instance civilians or aid agencies (that may be termed “white”) may be involved. Additionally, coalition forces may become prevalent with different shades of “blue” present in the play-box.

6.2.4 Identify causal relationships

Obviously the C2 process is the core in any combat. It is perhaps pertinent to note that this may not be the same as formal structure (“wiring diagram”) but the process model is the preferred approach. It determines which agents can and do communicate with each other. In WISDOM-II two agents from the same team can communicate if there is a connection between them in the C2 network and they are within the communication range. Both vision and communication construct the situation awareness. Based on their situation awareness and the objective of their missions, the agents decide on what to do in terms of movement, information collection, resupply or engagement.

6.2.4.1 The C2 network

It defines the command and control hierarchy within one force. Since the commands can only be sent from the agents at the higher level to the agents at the lower level, the C2 network is a directed graph.

6.2.4.2 The vision network

If agent A can see agent B, then there is a link from Agent A to Agent B. The vision network is also a directed graph.

6.2.4.3 The communication network

These communication networks could carry two types of information: situation information and commands.

In a traditional force, the situation information typically flows from an agent to other agents in the communication channel, from the swarm leader to the team leader and from the team leader to the commander. In a networked force, the situation information flows directly from the agent to its commander. In both the traditional force and the networked force, a common operating picture (COP) is developed at the headquarter based on a fusion of the collected information. In a traditional force, based on the COP the commander makes decisions for each group in the battlefield and sends commands to the team leader, then the team leader sends commands to the swarm leaders. However, in a networked force, all agents in the battlefield can access the COP through the communication channel. Therefore each agent has a global view of the battlefield while in a traditional force, each agent only has its own local view of the battlefield.

Since a network is employed to model communication, it is easy for WISDOM-II to support various types of communications: Point to Point directly (P2Pdirect), Point to Point indirectly (P2Pindirect) and Broadcast (BC). Because the information flows from source to sink, the communication network is obviously a directed graph.

6.2.4.4 The situation awareness network

This defines current knowledge about friends and enemies through vision and communication. The information collected by vision and communication is fused and then this network is developed. Since both vision and communication are direction dependent, the situation awareness network is a directed graph too.

6.2.4.5 The engagement network

It defines the agents being fired at based on the firing agent's current knowledge about its enemies and friends. This network is also a directed graph. Agents may die through firing, therefore the engagement network cause changes in the C2 network in the next time step.

6.2.5 Build causal network

From the above analysis, five concepts (relationships) have been extracted: command and control, vision, communication, situation awareness and engagement. Based on their causal relationships, the causal network, the top layer in the NCMAA is depicted in Figure 6.3.

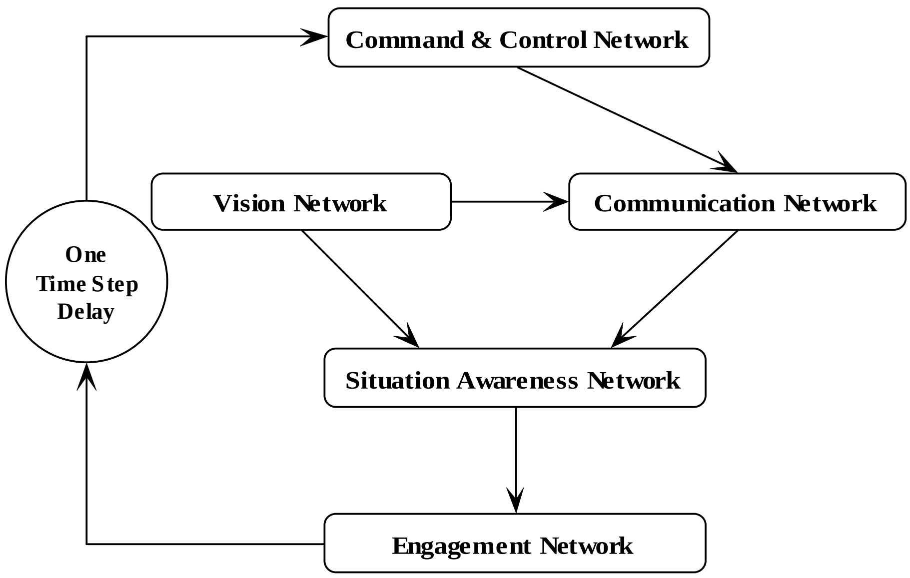


Figure 6.3: The causal network in WISDOM-II

6.2.6 Model agents and environment

6.2.6.1 Agent characteristics

Four types of agents are supported in WISDOM-II: combatant agent, swarm leader, team leader and commander. Both team leader and commander are virtual agents which exist in the force headquarters. They only have one capability: communication. Basically each combatant agent and swarm leader has five characteristic groups: health, vision, communication, movement and engagement.

The health defines the level of energy for an agent. With a user defined wounded threshold and immobile threshold, each agent may be in one of four health states: healthy, wounded, immobile, and dead (Figure 6.4). If an agent is wounded, it can go back to its team's hospital for treatment. This feature is first introduced in WISDOM-II to model recovery or limited resources, which will be described in the Section 6.2.10.

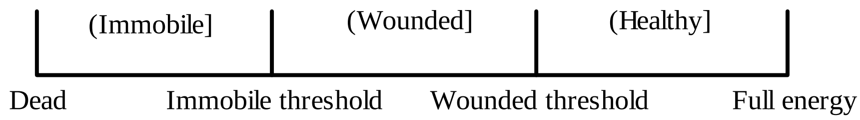


Figure 6.4: Agent health status

Each combatant agent has its own sensor which is defined by the sensor range and detection. The detection defines what kind of agents can be detected by using this sensor. If the detection of agent A is equal to or larger than the invisibility of agent B and Agent B is within Agent A's sensor range, then agent A will detect agent B. The value of detection should be between $(0, 1]$, with 1 representing the ability to detect all.

Combatant agents can communicate with other agents linked directly to them through the communication network. This communication occurs through a communication channel, which is modeled by the noise level, reliability, latency and communication range. The agent may only communicate with the agents within the range of that communication channel. A probabilistic model is adopted to implement the noise level and reliability of a communication channel. At each time step the message can only be transferred from one agent to another agent. The message will permanently be lost if it is older than a number of time steps predefined by the user. If there is a latency associated with a communication channel, any message through this channel has to wait a predefined number of time steps in order to be sent out.

The movement of an agent is determined by its speed and personality. WISDOM-II supports four kinds of speeds: still, low speed, medium speed and high speed. Agents with high speed can move one cell per time step. The low speed is one third of the high speed while the medium speed is half of the high speed. The movement algorithm is based on the tactical decision making and strategic decision making mechanism. The strategic decision making mechanism provides a guide to each group at the macro level while the tactical decision making mechanism is based on the agent personality to determine which location the agent should move to. The details of the movement algorithm are described in the Section 6.2.6.3 about the tactical decision making mechanism.

Engagement in WISDOM-II is determined by what kind of weapon the agent uses. The weapon is defined by the weapon power, fire range and damage radius. Based on the damage radius, two types of weapons are supported in WISDOM: point weapon, the damage radius of which is one, and explosive weapon, the damage radius of which is larger than one. WISDOM-II also supports direct and indirect fire. The indirect fire can fly over obstacles.

Status of each combatant agent and swarm leader are defined by their health level and position. There are four actions available to each combatant and swarm leader:

1. scanning the environment, which may change the status of vision, communication and situation awareness network;

2. communicating, which may change the status of the communication and situation awareness network;

3. movement, which may change the status of the vision network, the communication network, the situation awareness network, and the position status of agents;

4. firing, which may change the status of the engagement network, the C2 network, and the health status of agents.

6.2.6.2 Personalities of agents

Existing ABDs use the personality to define the tendency of an agent to move close to or far from certain types of agents while WISDOM-II uses it to define the influence of other agents on this agent. The influence of other agents may attract the agent to move closer to them or repulse the agent far from them. The personality in WISDOM-II is a vector quantity specified by its magnitude and its direction. The magnitude is the strength of the influence which is between 0 (no influence) and 1 (strongest influence). The direction defines at which direction the influence occurs. There are eight directions, defined by a value between 0 and 1 (figure 6.5).

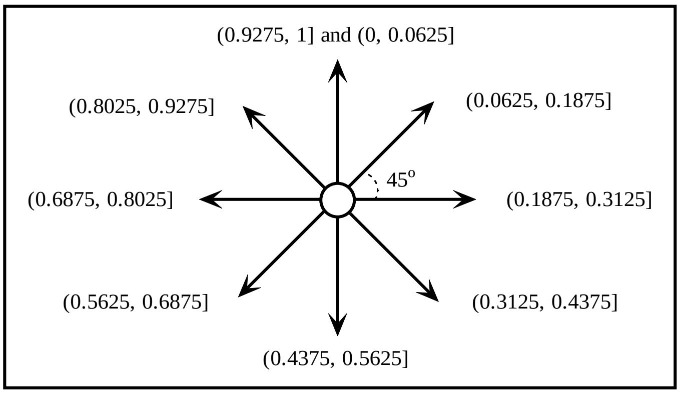


Figure 6.5: Force directions

The influence of agent A on agent B may be different from that of agent B on agent A. This difference can be reflected in either the value of the influence or the direction of the influence or both. Similar to existing ABDs, the personality is used to determine the new location the agent should move to.

For an agent, its movement is influenced by all the agents which it can detect or communicate with. The algorithm to aggregate and resolve movement is described later. The personalities for each type of movement influence are:

1. Influence vector of healthy friend agent via sensor;
2. Influence vector of wounded friend agent via sensor;
3. Influence vector of immobile friend agent via sensor;
4. Influence vector of healthy hostile agent via sensor;
5. Influence vector of wounded hostile agent via sensor;
6. Influence vector of immobile hostile agent via sensor;
7. Influence vector of healthy friend agent via communication;
8. Influence vector of wounded friend agent via communication;
9. Influence vector of immobile friend agent via communication;
10. Influence vector of healthy hostile agent via communication;
11. Influence vector of wounded hostile agent via communication;
12. Influence vector of immobile hostile agent via communication;
13. Influence vector of target flag.

The team leader and commander do not have personalities since they are virtual agents and do not move.

6.2.6.3 Decision making mechanism

There are two decision making mechanisms in WISDOM-II: tactical and strategic. The former is used by each agent to determine where it should move to and it is based on the agent's current knowledge and personalities while the later is used by each commander to determine the way point for each swarm and it is based on the swarm's mission type and the distribution of the firepower of each force in the battlefield.

Tactical decision making mechanism The movement of each agent is determined by its knowledge and personality. Only a healthy or wounded agent in the battlefield can move to a new location. In each time step, the agent can only move to its neighbor cell at the direction of the overall influence of all influencing agents. A movement function as in Equation 6.1 is constructed on the influence vectors and an agent moves in the direction of the resultant influence vector. The sum of the influence vector in the equation is a vector summation. Calculations are done synchronously with the moves. This process is repeated for each time step in the simulation. This movement algorithm is totally different from that implemented in any other ABDs:

$$RF_k = \sum_{i}^{n} \frac{\overrightarrow{F}_i^v}{D_{ki}} + \sum_{j}^{m} \frac{\overrightarrow{F}_j^c}{D_{kj}} + \frac{\overrightarrow{F}^t}{D_{kt}} \tag{6.1}$$

where:

RF denotes the resultant influence of an evaluated agent k;

n denotes the total number of agents perceived by agent k through vision in the information fusion network;

$\overrightarrow{F}_i^v$ denotes the influence vector from agent i, who is perceived by agent k through vision in the information fusion network;

m denotes the total number of agents perceived by agent k through communication in the information fusion network;

$\overrightarrow{F}_j^c$ denotes the influence vector from agent j, who is perceived by agent k through communication in the information fusion network;

$\overrightarrow{F}^t$ denotes the influence vector from agent k's target. If the evaluated agent is a combatant agent, then the target is its swarm leader. If the evaluated agent is the swarm leader, then the target is the swarm way point.

D_{ki}, D_{kj} and D_{kt} denotes the distance between agent k and the influencing agent i, j and target/waypoint.

In WISDOM-II, each cell can only accommodate one alive agent. If more than one agent would like to move to the same cell, a collision occurs, then the collision resolution mechanism is used to remove it. The collision resolution mechanism is defined by a set of rules.

- The agent which occupied this cell in the previous time step has the highest priority to stay in this cell;
- The swarm leader has the second highest priority to move to this cell;
- The wounded agent has the third highest priority to move to this cell, the wounded agent normally goes back to its hospital for recovery.
- If multiple agents with the same priority wish to occupy the cell, one is uniformly randomly chosen to move and the others stay in their original cells.

Strategic decision making mechanism A strategic decision is made for each swarm by the commander of each force based on the COP, which is the global view of the battlefield for that force, and the mission type. Each team is assigned one mission in the scenario definition input XML file by the user. Then all the swarms in that team have the same mission as the team mission. Four types of missions are supported in WISDOM-II: defend, occupy, attack, surveillance. The mission of defend means the swarm is trying to protect a certain area; the mission of occupy means the swarm is trying to occupy certain area; the mission of surveillance means the swarm is trying to collect all possible information about the battlefield; the mission of attack means that the swarm is trying to attack its enemy if its enemy can be detected. Otherwise the swarm approaches to its goal.

Three type of decisions can be made for each swarm: advance, defend and withdraw. The command of advance means the swarm need to go forward; the command of defend means the swarm need to protect the current location; the command of withdraw means the swarm need to escape from the current location. After making decisions, the high level commanders send the decision/command/intention to the lower level agents. To simplify the implementation of the command, the command sent to the swarm leader is the location of the way point.

The commander abstracts the whole environment into $n \times n$ (n is predefined by the user) hyper cells and calculates the total fire power of the hostile and own force for each hyper cell. The total fire power is a function of individual agent weapon power, fire range, damage radius and health level (see Equation 6.2).

$$P = \sum_{i}^{n} (f_i^P * f_i^R * r_i * h_i) \tag{6.2}$$

where:

P denotes the total fire power of one force in this hyper cell;

n denotes the total number of agents of one force in this hyper cell;

f_i^P denotes the weapon power of agent i in this hyper cell;

f_i^R denotes the fire range of agent i in this hyper cell;

r_i denotes the damage radius of agent i in this hyper cell;

h_i denotes the health of agent i in this hyper cell.

For each swarm with the mission of occupy, the commander calculates the force power ratio defined in Equation 6.3 for each of the surrounding hyper cells.

$$R_p = \frac{P_h}{P_o} \tag{6.3}$$

where R_p is the fire power ratio, P_o is the total fire power of own force in that hyper cell and P_h is the total fire power of hostile force in that hyper cell. The way point for each swarm is generated based on Equation 6.4.

$$WP = \begin{cases} C_i, & \text{if } \exists\ R_p^i \leq \theta_a,\ C_i \in C^b; \\ C_0, & \text{else if } \forall\ R_p^i > \theta_a,\ R_p^i \leq \theta_d,\ C_i \in C^b; \\ C_j, & \text{else } \forall\ C_k \in C^s,\ R_p^j = min(R_p^k). \end{cases} \tag{6.4}$$

where:

θ_a is the advance threshold while θ_d is the defend threshold. Both of them are defined by the user in the input configuration XML file;

C^b is the set of hyper cells which is between the swarm and its goal. For example, if the center of the swarm is in the cell of G in Figure 6.6, the set of hyper cells for various placement of the goal is defined in Table 6.1. The maximum number of the hyper cells in this set is three;

C^s is the set of hyper cells which is surrounding the hyper cell of that swarm. The maximum number of the hyper cells in this set is eight;

C_0 is the hyper cell which is occupied by that swarm;

C_i is a hyper cell in C^b;

C_j is a hyper cell in C^S;

$R_p^i,\ R_p^j,\ R_p^k$ is the force power ratio in the hyper cell i, j or k respectively.

If the firepower ratio of one hyper cell in C^b is less than or equal to the advance threshold θ_a, the center of this hyper cell will be assigned to the swarm as the way

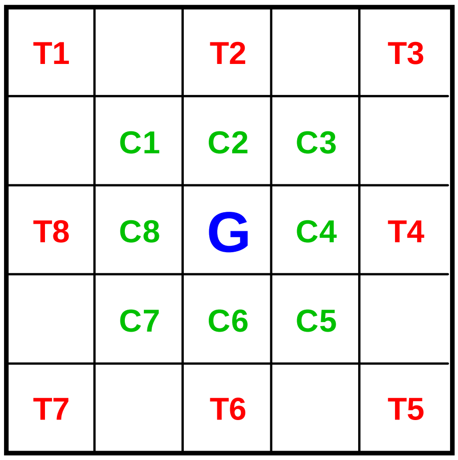
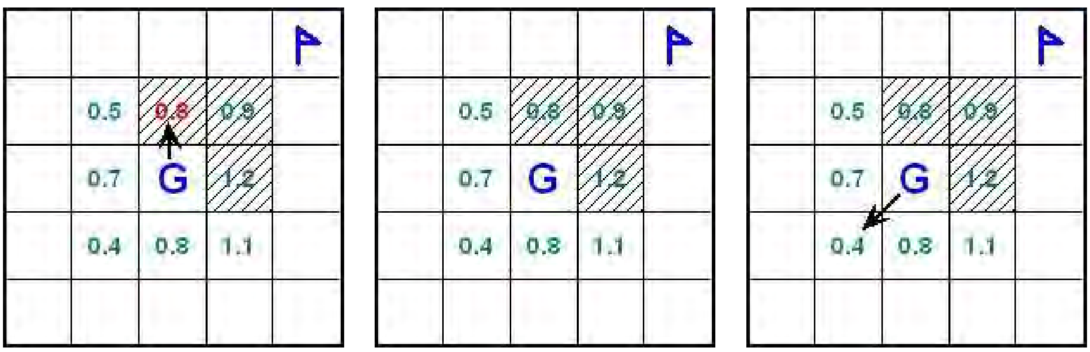


Figure 6.6: Hyper cells in between

Table 6.1: The set of hyper cells in between

Relative position of goal	Hyper cells in between (C^b)
top-left, e.g. $T1$	$C1$, $C2$ and $C8$
top, e.g. $T2$	$C1$, $C2$ and $C3$
top-right, e.g. $T3$	$C2$, $C3$ and $C4$
right, e.g. $T4$	$C3$, $C4$ and $C5$
bottom-right, e.g. $T5$	$C4$, $C5$ and $C6$
bottom, e.g. $T6$	$C5$, $C6$ and $C7$
bottom-left, e.g. $T7$	$C6$, $C7$ and $C8$
left, e.g. $T8$	$C7$, $C8$ and $C1$



(a) $\theta_a = 1.3$ (b) $\theta_a = 0.7, \theta_d = 1.3$ (c) $\theta_d = 1.3$

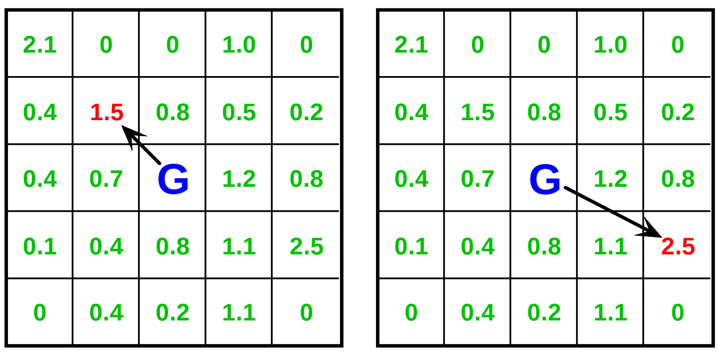
Figure 6.7: Example of waypoint for mission of occupy. The value in the square is the force power ratio. The shadow cells are the set of hyper cells in-between. The flag is the goal of the swarm.

point (see Figure 6.7(a)). If the firepower ratios of all hyper cells in C^b are between the advance threshold θ_a and the defend threshold θ_d, the swarm will stay in the original hyper cell (see Figure 6.7(b)). Otherwise the commander selects the center of the hyper cell which has the minimal firepower ratio among all surrounding hyper cells as the way point for the swarm (see Figure 6.7(c)).

If more than one hyper cell meet the condition, the commander will randomly select one. When the command is sent to the swarm, the swarm leader may misunderstand it. Such misunderstanding is modelled by the probability of misunderstanding and the variance of misunderstanding. The variance defines the degree of a received way point deviating from its correct location. After receiving the command, the swarm leader may or may not follow it based on the probability of following the command.

For a swarm with the mission of attack, the waypoint is the centre of the surrounding hyper cell with highest firepower of the hostile force (see Figure 6.8(a)). If there is no enemy detected in the surrounding hyper cells, the way-point is the centre of the hyper cell randomly chosen from the hyper cells between the swarm and its goal.

For a swarm with the mission of surveillance, the way point is the hyper cell with highest hostile force power (see Figure 6.8(b)). In order to maximize the information collected, no more than one swarm will be assigned to the same hyper cell.



(a) Mission of attack (b) Mission of surveillance

Figure 6.8: Example of waypoint for mission of attack and surveillance. The value in the square is the force power ratio.

Since the area the team needs to defend is predefined by the user in the scenario definition input XML file, no command will be sent to the swarm with the mission of defend.

The frequency of the commander sending the command is also defined by user in the scenario definition input XML file.

6.2.6.4 Recovery

One of the most important aspects in military operations is the logistics, where the medical treatment system is one of the key components. The model of the artificial hospital is first introduced in WISDOM-II. Each team may have a hospital in the team base, which is defined by the number of doctors and the recovery rate. If the team has a hospital, the wounded agent will move back to the hospital for treatment. Each doctor can treat only one wounded soldier at each time step and the health of that treated soldier will be increased by the recovery rate at each time step. If all doctors are already treating, the wounded soldier will be put in the queue to wait for treatment. When the agent is fully recovered, it will move back to the battlefield nearby its swarm leader. If its swarm leader is in the hospital or it is the swarm leader, it will be positioned in the cell around the team base.

The waiting queue and the number of doctors in the hospital are used to model the limited resource available and the recovery capability for a force. The capability to make the wounded soldier fully recovered is a key issue in warfare. With such kind of recovery model the analysts may search for the minimal resource needed for maintaining a minimal level of the recovery capability for a force.

6.2.6.5 Terrain feature

WISDOM-II adopts 2-dimension environment and supports impassable objects. Agents cannot see or travel through impassable blocks. With the indirect weapon, the agent can shoot its enemies over them.

6.2.7 Develop finite state machine

Each of the conceptual networks may have one or more instances defined for the blue and/or red agents. There are seven instances of networks defined in WISDOM-II (see Figure 6.9). Since different forces have different C2 structures and communication can only occur within the same force, there are two instances of the C2 network and two instances of the communication network, one for the blue force and one for the red force, while there is only one instance for each other network in WISDOM-II.

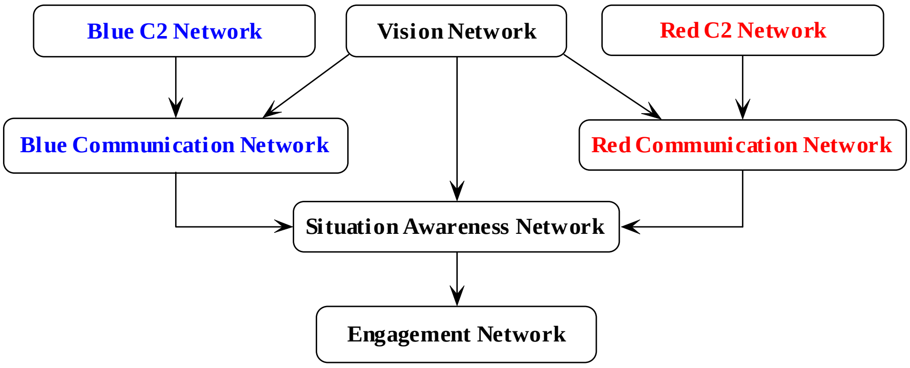


Figure 6.9: The bottom layer of the NCMAA in WISDOM-II

The state of each network is determine by the state of agents within the network and the network structure. The structure of both the blue and red C2 network is changed only when some agents die and are removed from the network. The movement action may influence the state of vision and communication network. In turn it will change the state of the situation awareness network. Obviously, the firing action influences the state of the engagement network.

6.2.8 Identify interaction between networks

Based on the causal network, these instance networks with their interactions make up the bottom layer of the NCMAA (Figure 6.9). This is the basis of Granger causality for conducting the network based causal reasoning using path analysis and root cause analysis in WISDOM-II.

6.2.9 Develop network based analysis and reasoning engine

Recently defence analysts have realized that network analysis (Dekker 2005a; Dekker 2005b; Xu and Chen 2005b) and causal model (influence diagram) (Curtis and Dortmans 2004) are two valuable tools in military analysis. The properties of the networks, such as communication networks, and command and control networks, may largely affect the outcome of military operations. For example, Dekker analysed the effect of network topology on the military performance (Dekker 2005a), classified the architecture of network centric warfare based on network measures (Dekker 2005c) and evaluated the robustness of military critical infrastructure networks (Dekker 2005b). Xu (Xu and Chen 2005b; Xu and Chen 2005a) applied network theory into analysis of terrorism. The real-time reasoning engine in WISDOM-II is based on the properties of the networks in military operations, through a series of causal reasoning, time series analysis on these network and military performance measures to allow defence analysts to understand the dynamics and capture the embedded patterns during the simulation.

The architecture of the NCMAA facilitates real-time reasoning during the simulation as follows:

- The first step is to establish a causal network as the basis for cause-and-effect relationships. This causal network (see Figure 6.3) is established by domain experts.
- Second, some performance measures, such as casualty or loss exchange ratio, and all meaningful network measures which can be interpreted in the military domain, e.g. the average shortest path length of the C2 network can be interpreted as a measure of the speed of command, are calculated at each simulation time step, and streamed and fed into a set of moving windows with different user predefined size.
- Third, correlation analysis among these windowed time series data are conducted based on the causal network.

- Finally a natural language processing centre interprets these results and presents them in a natural language format which may be easily understood by the user.

If we imagine that the simulation is at time step t, a window of length w is established to extract the time series between $t - w$ and t. The network and performance measures are calculated. Examples of these measures include: the damage (casualty) of the blue and red force, and average degree, average shortest path length and the clustering coefficient of each C2 network, communication network, vision network and engagement network, over time. Within a window W, the correlation coefficient is calculated between all possible influenced time series according to the cause-and-effect relationships defined by the causal network (Figure 6.3), e.g. calculate the correlation coefficient between the average degree of the communication network and the force casualty. The information is then filtered using a thresholding mechanism. Those measures above a certain threshold predefined by the user are sent to the natural language engine. The natural language engine is responsible for forming sentences written in plain English to explain the dynamics of the simulation to the user.

Figures 6.10 and 6.11 are two examples of the correlation analysis between the force damage and the structure of the network. Figure 6.10 is the correlation coefficient between the red damage and the blue communication network measures over time while Figure 6.11 is the correlation coefficient between the blue damage and the red communication network measures. Both examples use the window size of 5 and the threshold of 0.5. At points above 0.5, we may think that the communication activity is probably the key reason leading to the damage of their enemy during the last 5 time steps. Therefore based on the correlation coefficient, the nature language processing center may present its interpretation to the user as: "The damage of the red force during time steps 53 to 58 is probably caused by the communication activities within the blue team".

Following are the templates used in the system:

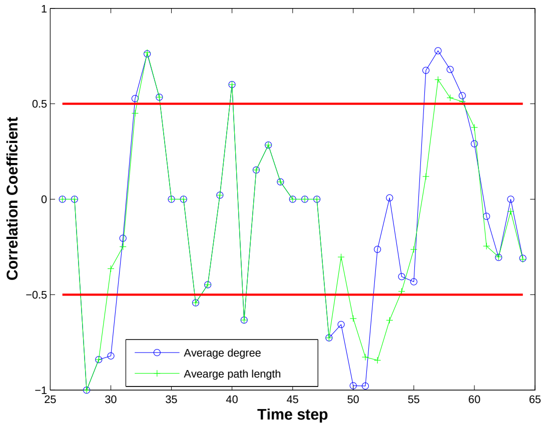


Figure 6.10: The correlation coefficient between the red damage and the blue communication network measures

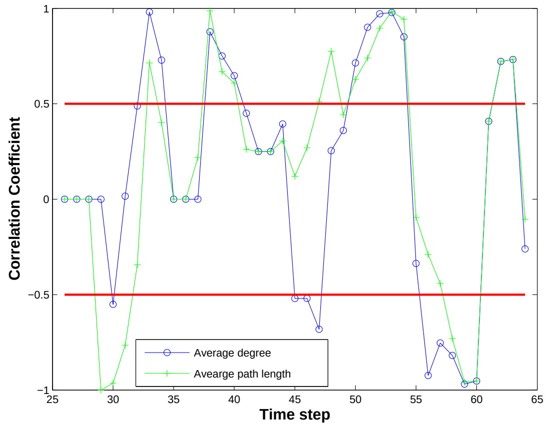


Figure 6.11: The correlation coefficient between the blue damage and the red communication network measures

1. The ? (blue/red) team caused more damage to the ? (blue/red) team. The damage ratio is ? .

2. Without any damage, the ? (blue/red) team causes damage of ? health points to the ? (blue/red) force.

3. An order has been sent to the leader of group ? in the ? (blue/red) team to move toward ? .

4. The group ? in the ? (blue/red) team is ? (advancing to/defending in/withdrawing to) ? .

5. The situation awareness of the ? (blue/red) team is gained mainly through ? (vision/communication).

6. The agents in the ? (blue/red) team are coordinating their firing to achieve maximum damage in the ? (blue/red) team. Or the agents in the ? (blue/red) team are maximizing their fire range to achieve maximum damage in the ? (blue/red) team. Or the agents in the ? (blue/red) team are spreading their fire to achieve maximum damage in the ? (blue/red) team.

7. An average damage of ? occurred in the ? (blue/red) team over the last ? time steps is properly caused by the activities in the ? network of the ?

(blue/red) team.

8. An average damage of _?_ occurred in the _?_ (blue/red) team over the last _?_ time steps is properly caused by the situation awareness of the _?_ (blue/red) team on _?_ (agent/force) level.

9. An average damage of _?_ occurred in the _?_ (blue/red) team over the last _?_ time steps is properly caused by the degree of order in the _?_ (blue/red) team.

The first two templates explain what is happening on the battlefield. For example the reasoning interface may show that "*The red team caused more damage to the blue team. The damage ratio is 1:2*", or "*Without any damage, the blue team causes damage of 16 health points to the red team*".

The template 3 shows the command sent by the force commander to the swarm. For example, it may be "*An order has been sent to the leader of group 2 in the blue team to move toward (35, 25)*". The template 4 presents what the group is doing. For example, it may be "*The group 2 in the blue team is advancing to (35, 25)*". Based on the template 3 and 4, the user may predict the system behaviour during the simulation.

The template 5 and 6 gives explanation on how it happens. The template 5 presents how a team gains its situation awareness. If more than half situation awareness of the blue team is from communication, the reasoning interface shows that "*the situation awareness of the blue team is gained mainly through communication*". Regarding the template 6, every time step the system calculates the variance of the in-degree for each team in the firing network. If the variance of the in-degree of one team is high, it means the agents in the other spread their fire to different enemies. Therefore the damage of that team is high. If so many agents in one team fire at same enemy, the maximum damage is the total health of one single agent. So when the variance of the in-degree of the red team exceeds the threshold defined in the XML configuration file, the system explains this situation to the user that "*the agents in the blue team are coordinating their firing to achieve maximum damage in the red team*".

The template 7, 8 and 9 present why it happens. Within the window size, the system calculates the correlation coefficient between network measures, such as the average shortest path length, average degree and clustering coefficient, and the damages of each team. If the correlation coefficient exceeds the threshold defined in the XML configuration file, the reasoning interface will show that, for example, "*an average damage of 4 occurred in the red team over the last 5 time steps is properly caused by the activities in the communication network of the blue team*". The system also calculates the correlation coefficient between the accuracy of the situation awareness and the damage of each team. The accuracy of the situation awareness is measured by the percentage of the correct information collected. The accuracy of the situation awareness on agent level is the average percentage of the correct information of all combatants while the accuracy of the situation awareness on force level is the percentage of the correct information in COP. If the correlation coefficient exceeds the threshold defined in the XML configuration file, then the template 8 is used as, for example, "*An average damage of 4 occurred in the red team over the last 5 time steps is properly caused by the situation awareness of the blue team*". Regarding the template 9, if the correlation coefficient between the spatial entropy and the damage exceeds the threshold defined in the XML configuration file, the reasoning interface shows that, for example, "*An average damage of 4 occurred in the red team over the last 5 time steps is properly caused by the degree of order in the blue team*".

6.2.10 Design other features required in WISDOM-II

6.2.10.1 Visualization

One drawback of existing agent based warfare simulation systems is that they only provide limited information to the analysts during the simulations (Yang et al. 2005b). WISDOM-II fills this gap. Beside visualizing the information of each single agent, WISDOM-II also visualizes each network and all possible network measures spatially and temporally. By spatially visualizing each network, WISDOM-II provides a graphic view for each type of interactions between agents. It offers a window for the user to know what kind of interaction occurs between agents and to identify the role of each interaction during the simulations. By temporally visualizing the network measures, WISDOM-II presents a time series dynamics of each interaction with the system. It may help analysts to predict and analyze the outcome of the simulated war.

Temporally and spatially visualizing each network and its measures makes it easier to validate and verify the system itself. If something happens, it will immediately be reflected on the visualization. The developer or the user can then identify and capture it quickly. Figure 6.12 presents two examples of spatial visualization of the communication and vision network. Assume the communication network is initialized as a complete network, i.e. every agent connects to each of the others. If there is no link between some agents, the user may easily identify there is something wrong. Figure 6.13 are some examples of temporal visualizations of network measures, force performance measures and the spatial entropy, which is a measure of the degree of clustering of a force on the battlefield (Ilachinski 1999). With these graphs, the analysts may easily capture the dynamics of the communication and vision network, and the force performance during the simulation. Before the time step of 43, the changing trend of both blue and red communication network is similar. The changing trend of the force damage for both teams is also similar. However, the average degree of the blue communication network is higher than that of the red communication network. It may be the reason that the red damage is slightly higher than the blue damage. After that, the average degree of the red communication network decreases faster than that of the blue communication network while the red damage increases faster than the blue damage. Combining with the dynamics of the vision network, one may conclude that the high damage of the red force is because of no effective communication. With the average degree of the vision network decreasing, the blue team takes advantage of its communication network to retrieve more information about its environment and enemy. Consequently it has low damage. If inspecting the time series of the spatial entropy, one may find that the spatial entropy of the blue team is higher than that of the red team from time step 18. It suggests that spreading a force is better than clustering it in this scenario. Table 6.2 presents all the information temporally and spatially visualized in WISDOM-II.

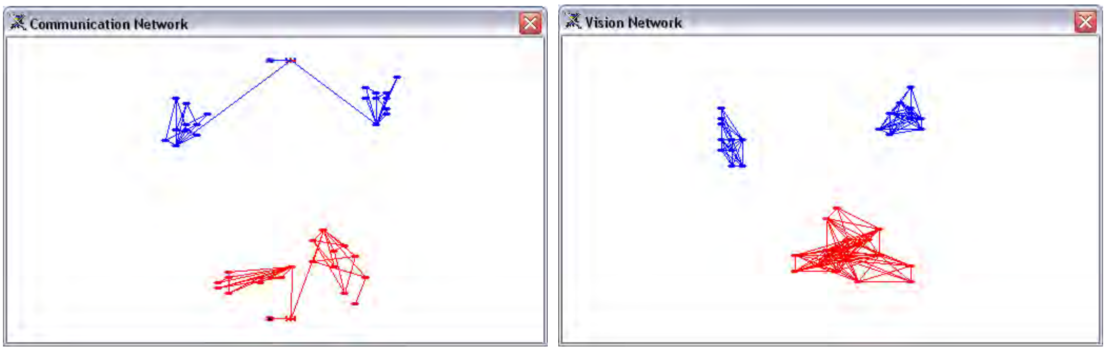


Figure 6.12: Examples of spatial visualizations

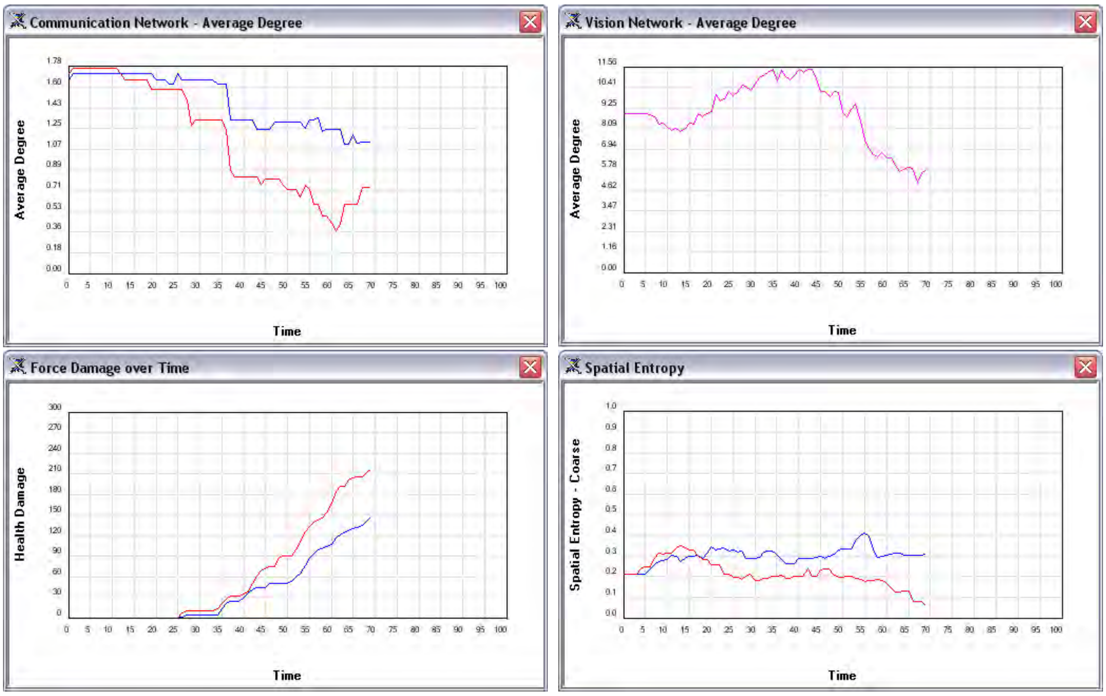


Figure 6.13: Examples of temporal visualizations

Combat entropy (E_c) can be used as a predictor of combat outcomes (Ilachinski 1999). It is defined by:

$$E_c = \frac{c}{N_a} \log \frac{c}{N_a}$$

Table 6.2: Visualized information in WISDOM-II

Temporal Visualization	Force damage Fraction of remaining healthy agents for each force Fraction of remaining wounded agents for each force Fraction of remaining immobile agents for each force Fraction of treating agents for each force Fraction of remaining alive agents for each force Fraction of dead agents for each force Combat entropy (E_c) Spatial entropy ($E_s(b)$) Average shortest path length of C2 network Average shortest path length of vision network Average shortest path length of communication network Average shortest path length of engagement network Average degree of C2 network Average degree of vision network Average degree of communication network Average degree of engagement network Clustering coefficient of C2 network Clustering coefficient of vision network Clustering coefficient of communication network Clustering coefficient of engagement network
Spatial Visualization	Common operation picture for each force C2 network Vision network Communication network for each force Situation awareness network Engagement network

where c represents the casualty (in absolute numbers) and N_a represents the force strength of the adversary (either red or blue).

Spatial entropy provides a measure of the degree of disorder of a battlefield state (Ilachinski 1999). It is defined by:

$$E_s(b) = (2 \log b)^{-1} \sum_{i=1}^{b^2} p_i(b) \log p_i(b)$$

$$p_i(b) = \frac{N_i(b)}{N}$$

where b is the size of the sub-block of the (D/b)-by-(D/b) array of sub-blocks into which the battlefield is partitioned, and D is the dimension of the battlefield. N is the number of agents on the battlefield, $p_i(b)$ represents the probability of finding an agent in the ith sub-block. log is the logarithm base-2, b^2 is the total number of sub-blocks on the battlefield, and $(2\log b)^{-1}$ is a normalization constant.

Finally combining with the real-time reasoning component, the analysts may gain deeper understanding of what happens in each time step during the simulation and how the simulation is progressing.

6.2.11 System implementation and testing

WISDOM-II is implemented in C++ with two run modes: command line and window version.

6.2.11.1 Validation

The validation and verification approach proposed by Sargent (Sargent 1999) was followed and integrated it to the whole development procedure. The following are some examples of validation and verification techniques (Sargent 1999) used in WISDOM-II:

- Animation: Through the visualization component, the system operational behaviors is graphically displayed over time, e.g. the structure of networks and the position of agents (Figure 6.12). It may help us to verify that the programming is correct and the system does what is expected.
- Comparison to other models: a comparison study is conducted between WISDOM-II and MANA. Table 6.3 shows the loss exchange ratio over 500 runs between MANA and WISDOM-II in two simple scenarios, coded in a similar way (Yang et al. 2005a). Through comparison, it appears that the model is consistent with an other model.

Table 6.3: The LER over 500 runs between MANA and WISDOM-II

Traditional Force		Networked Force	
MANA	WISDOM-II	MANA	WISDOM-II
0.95 ± 0.33	1.07 ± 0.47	0.93 ± 0.30	1.07 ± 0.40

- Event validity: Compare occurrences of events between simulation system and real system. Following are examples used for WISDOM-II:
  - Health status: a number of scenarios were created where there are three agents with three different status: immobile, wounded and healthy. It shows that the healthy agent always move to its flag, the wounded agent always move back to its hospital and the immobile agent always stay where it is. Another set of scenarios were created where there is a healthy agent, a wounded agent and immobile agent, but there is no hospital. The simulations show that the healthy and wounded agent always move to their flag while the immobile agent stays where it is.
  - Vision: a number of scenarios were created where agent A is within agent C's vision range while agent B is beyond agent C's vision range. The results show agent C can always detect agent A and never detects agent B.
  - Communication: a number of scenarios were created where agent A is within agent B's communication range while agent B is outside agent A's communication range. The results show that the information is always transferred from agent B to agent A and no information is transferred from agent A to agent B.
  - Firing: a number of scenarios were created where agent A is within its enemy agent B's firing range while agent B is outside its enemy agent A's fire range. The simulated results show that agent B can always fire at agent A while agent A cannot fire at agent B.
  - Movement: to test the movement algorithm, a number of scenarios were created with agents with different personality: tend to move close friend

or enemy, and avoid moving close to friend or enemy. The simulations show that the movement of agents are exactly what are expected. Another set of scenarios are created where one agent is set with high speed, one agent with medium speed and another one with low speed. The results show that the agent with the speed moves fastest, the agent with low speed moves slowest, and the agent with the medium speed is in between.

  - Environment: a set of scenarios were created where agent B is within agent A's vision range and agent B is behind a impassible block. The simulation results show that agent A cannot detect agent B because of the block although it is with its vision range. Another set of scenarios are created where agent B is with agent A's firing range and agent A is equipped with either a direct weapon or indirect weapon. The results show that the direct weapon can never shoot enemies behind an impassable block while the indirect weapon can.

- Extreme condition tests: The system should be able to deal with any extreme and unlikely combination of levels of factors in the system. The followings are examples for WISDOM-II:

  - Vision: a set of scenarios were created where some agents are invisible. The results show that these agents cannot be detected by any other agents.

  - Communication: a set of scenarios were created where the communication range of some agents is zero. The results show that there is no link between these agents with other agents in the communication network. That means no communication occurs between these agents and other agents.

  - Firing: a set of scenarios were created where the firing range of some agents is zero. The results show that these agents can never shoot their enemy.

  - Movement: a set of scenarios were created where there are some agents with speed of zero. The results show that agents with speed of zero always stay in the original place.
  - Environment: a set of scenarios were created where one agent is surrounded by an impassible block. The results show that the situation awareness is empty if the communication range of its friendly agents is zero. This is because it cannot get information about other agents through both vision and communication. Therefore it cannot fire its enemy even if it has an indirect weapon with a long firing range. If the communication range of its friends is not zero, its friend can then send a message to it through communication. It can fire at its enemy with its firing range only if it is equipped with an indirect weapon.

- Operational graphics: Graphically display the values of various performance measures during the simulation, e.g. the spatial entropy, force damage and casualty (Figure 6.13).

6.3 Summary

WISDOM-II is re-designed and re-developed on the new architecture of NCMAA. It is much different from other existing ABDs in the sense of architecture, functionality and capability. It has a number of unique features. WISDOM-II is

- The first system to have a built-in network analysis tool;
- The first system that is able to provide group-level real-time reasoning and interpret the outcomes in a natural language during the simulation;
- The first system that combines the flexibility and scalability of agent-based distillation (tactical level) and the rationality of cognitive agents (strategic level);

- The first system to introduce the concept of artificial hospital for recovery and modelling simple resource technical constraints;
- Flexible to switch between different force structures to investigate future concepts;
- Flexible to test different command, control, and communication (C3) structures;
- Flexible to test different communication problems such as network structure, noise, latency, etc;
- Having a four level C2 structure and heterogeneous agent at the swarm (group) level;
- Using modern visualization techniques to capture the dynamics and patterns during the simulation.

Since it is developed on the NCMAA architecture, it can overcome limitations of existing ABDs to some degree. With help of the network measures, WISDOM-II may easily verify and validate the underlying structure. If a certain network collapses, the user can immediately detect it through the network measures and visualizations. WISDOM-II conducts the real-time reasoning at the network (group) level. Such reasoning captures the domain specific interaction between networks (interactions) and an interface provides a real-time interpretation of the simulation process in nature language for the analyst. This may overcome the limitations of the reasoning at the individual agent level. This feature allows WISDOM-II to be able to deal with problems with high complexity. Since everything in WISDOM-II is a network or a part of a network, the new concept, e.g. NCW, can be easily modelled, analyzed and verified by using WISDOM-II. The use of a network as the representation unit in WISDOM-II also facilitates efficient parallelism based on network structure, and grounded modelling. In WISDOM-II, a rule based algorithm is used to make strategic decisions, which guides a semi-reactive agent to make tactical decisions. Therefore, the interaction between tactics and strategies is easily captured and studied. Concepts such as information misunderstanding, communication, level of information fusion, etc can now be studied in a unified framework. Overall, WISDOM-II is a promising ABD system which creates a new approach for analysts to understand the dynamics of and gain insight into warfare.

Chapter 7

Fitness Landscape Analysis with WISDOM-II

7.1 Introduction

Chapter 4 characterizes the fitness landscape based on WISDOM-I by using the information analysis approach. In this chapter, the fitness landscape generated by WISDOM-II is analysed using the same approach as in chapter 4 and is compared with that from WISDOM-I.

7.2 Experimental setup

In chapter 4, six scenarios (strategies) are defined, analysed, and can be classified into three classes: GOL and BAL, DEF and COW, and AGG and VAG based on the characteristics of the fitness landscape. In this chapter, three strategies, one from each group, (see Table 7.1) are chosen for the red team while the strategy (a vector of personalities) of the blue team is allowed to vary.

In order to facilitate the comparison of the fitness landscapes generated by WISDOM-

Table 7.1: Different strategies for the red team used in the experiments

Scenario	**Friend**	**Enemy**	**Goal**
Goal Oriented (GOL)	Neutral	Neutral	Target
Aggressive (AGG)	Neutral	Attack	Neutral
Defensive (DEF)	Cluster	Neutral	Neutral

I and WISDOM-II, most unique features of WISDOM-II are turned off except two decision making mechanisms: strategic which guides the movement of the whole group, and tactical which guides the movement of each individual. WISDOM-I does not have a strategic decision making mechanism. Further, the tactical decision making mechanism is different between WISDOM-I and WISDOM-II. In WISDOM-II, the decision variables are represented with a vector of 18 real numbers representing different characteristics of personalities as follows:

1. $P_1 - P_2$: influence (including magnitude and direction) of a healthy friend within the vision range;
2. $P_3 - P_4$: influence (including magnitude and direction) of a healthy friend within the communication range;
3. $P_5 - P_6$: influence (including magnitude and direction) of an injured friend within the vision range;
4. $P_7 - P_8$: influence (including magnitude and direction) of an injured friend within the communication range;
5. $P_9 - P_{10}$: influence (including magnitude and direction) of a healthy enemy within the vision range;
6. $P_{11} - P_{12}$: influence (including magnitude and direction) of a healthy enemy within the communication range;
7. $P_{13} - P_{14}$: influence (including magnitude and direction) of an injured enemy within the vision range;

8. $P_{15} - P_{16}$: influence (including magnitude and direction) of an injured enemy within the communication range;

9. $P_{17} - P_{18}$: influence (including magnitude and direction) of the target (way-point/flag).

All personalities (decision variables) are real numbers in the range of $[-1, 1]$. The corresponding direction is defined by Figure 6.5 in chapter 6. The configurations of the environment, initial position and the number of agents are the same as in chapter 4 (see Figure 4.1). A single evaluation of the game involves repeating the simulation 100 repeats, each for 500 time steps.

The same objective function and fitness function are adopted as in chapter 4 (see Equations 4.14 and 4.15).

7.3 Random walk

As in chapter 4, ten different random walks are taken, each of length 10,000 solutions using two fitness functions (average and normalized). Each stochastic neighbourhood in the search space was obtained by adding a random number drawn from a Gaussian distribution with zero mean and 0.1 standard deviation to each variable in the genotype. If the value of any personality is out of the range $[-1, 1]$, the value is truncated.

Figure 7.1 depicts the time series of the best solution found so far for random walk by using the average fitness and the normalized average fitness. According to the average fitness, one may see that the best solution found in the AGG scenario is higher than that in the GOL scenario or DEF scenario. As discussed in chapter 4, in the scenario when the red team would always like to attack its enemy, it may lose coordination among the red agents, for example, in the AGG scenario. Therefore the blue team may easily damage the red team. However, the result from WISDOM-II shows the normalized average fitness value of the best solution found in the DEF scenario is the highest among these three scenarios. The very low value of the normalized average fitness in the GOL and AGG scenario suggests that the searching process in both scenarios involves a large amount of variations. The stochasticity plays a critical role in both GOL and AGG scenarios.

Figure 7.1 also shows that the improvement mostly occurs at the start of the search stage for both average fitness and normalized average fitness. The improvement almost stops for both AGG and GOL scenarios after 2000 steps. However better solutions can still be found in the DEF scenario after 2000 steps, especially when using the normalized average fitness.

When compared with Table 4.2 in chapter 4, similar patterns can be observed except that the normalized average fitness is lower in WISDOM-II than in WISDOM-I for the GOL scenario. This implies that the influence of the stochasticity is higher in WISDOM-II than in WISDOM-I.

Figure 7.2 presents some representative examples of the time series of fitness value for random walk. Obviously the fitness landscape is quite rugged for both average fitness and normalized average fitness. In this chapter, a good solution is defined as that the blue damage is less than the red damage. That is, the fitness value of the average fitness is larger than 200. It is very hard to find a good solution for both GOL and DEF scenario. Only few good solutions can be found and are highly separated by a number of bad solutions. For the AGG scenario, lots of solutions found are good solutions.

The low fitness value of the normalized average fitness in all three scenarios in Figure 7.2 suggests that all the solutions found in all three scenarios, especially in the GOL and DEF scenario, are unstable. This is consistent with the above findings about the role of the stochasticity.

When compared with Figure 4.2 in chapter 4, one may easily see the difference. The signal-worst solutions, as defined in chapter 4, are lower (and therefore bet-

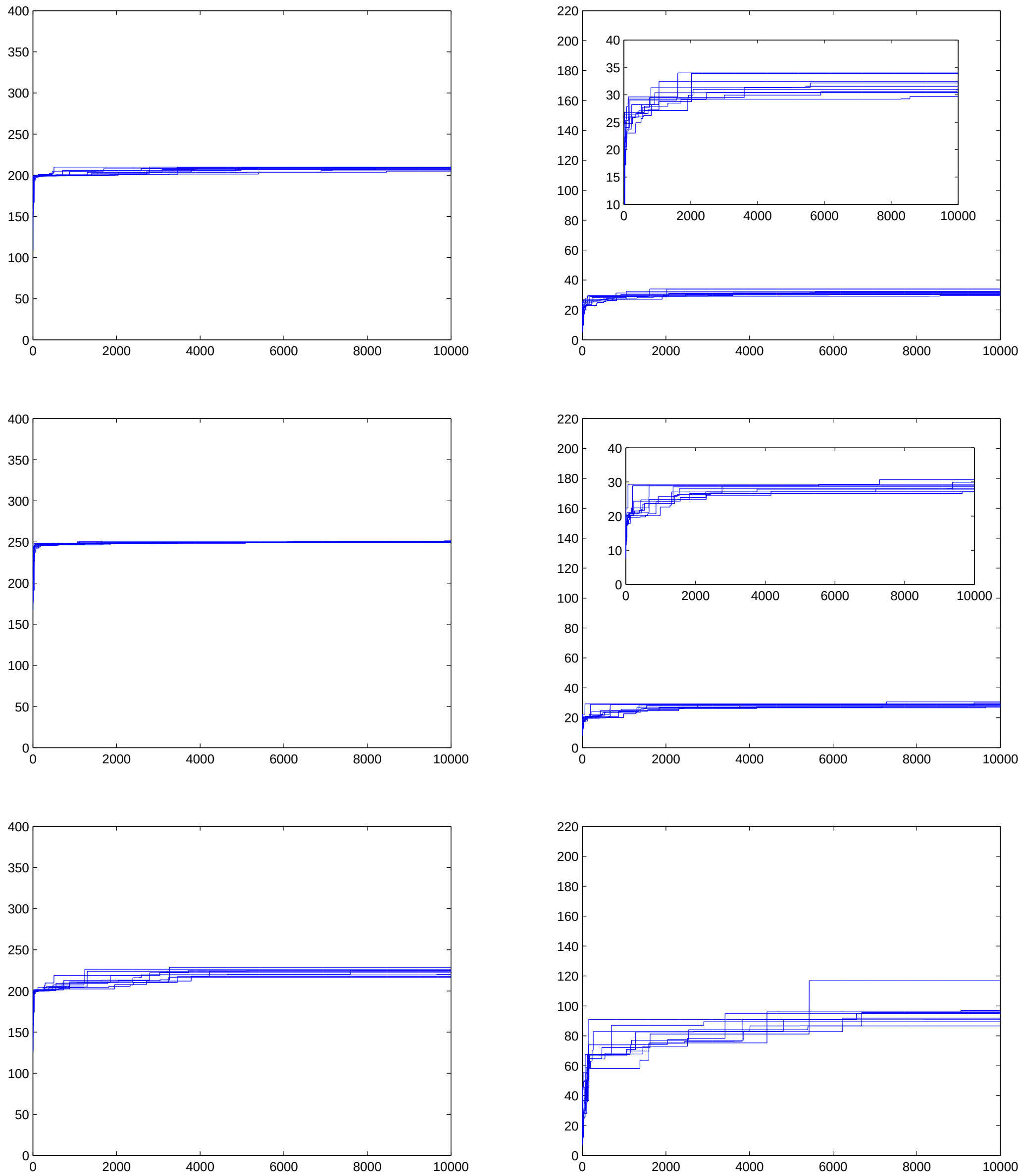


Figure 7.1: The best solution found over time by using the average fitness (on left) and normalized fitness (on right) for random walk. The order for top to bottom is: GOL, AGG, and DEF

ter) in both average fitness and normalized average fitness for all three scenarios in WISDOM-II than in WISDOM-I. This is because WISDOM-II has a strategic decision making mechanism to coordinate the behaviours of the agents. Since WISDOM-I does not have this kind of coordination mechanism, the fitness value of the worst solution searched is higher in WISDOM-II than in WISDOM-I.

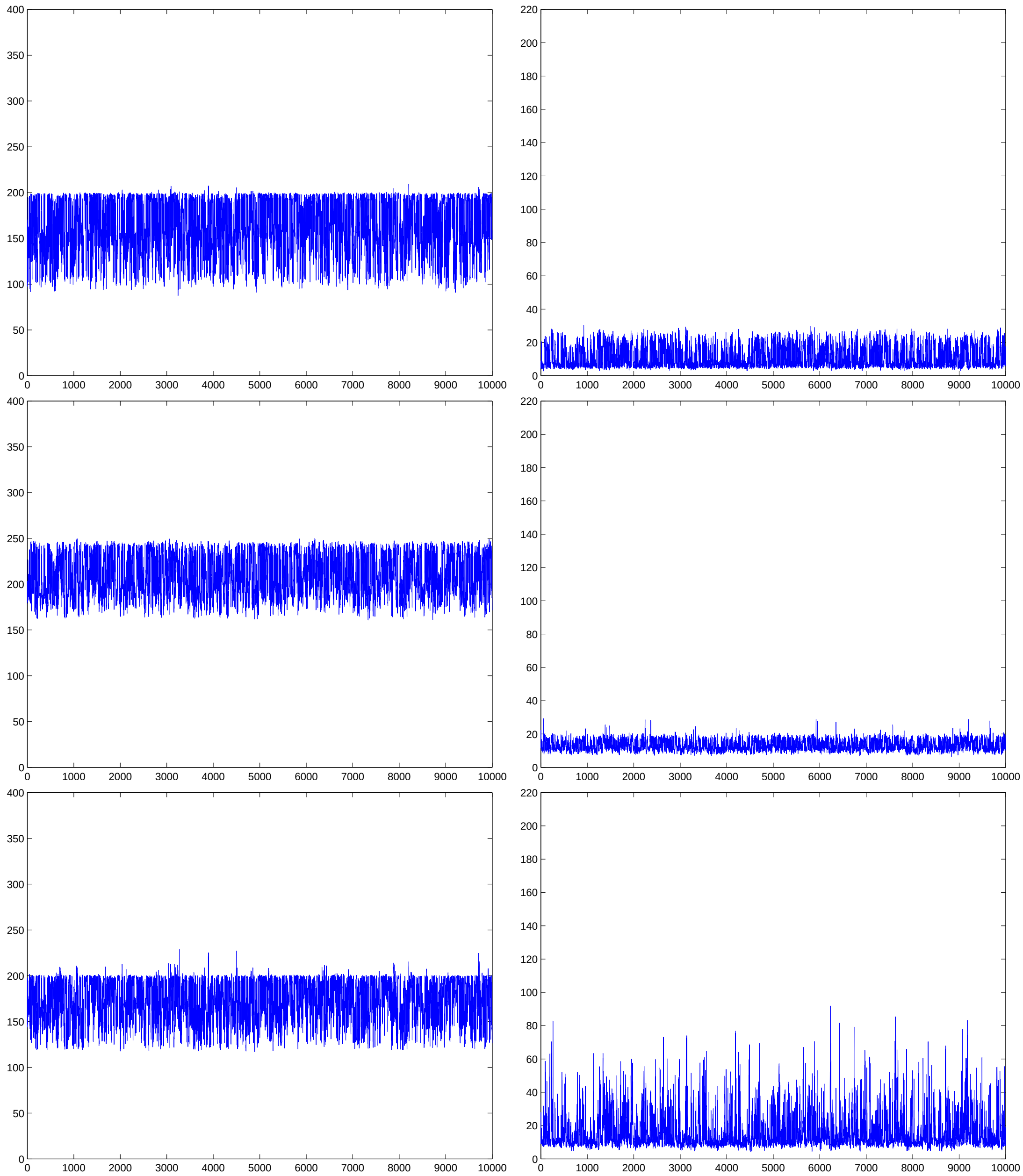


Figure 7.2: The fitness value over time for random walk using average fitness (on left) and normalized average fitness (on right). The order from top down is: GOL, AGG, DEF, respectively

The tit-for-tat situation also does not appear in WISDOM-II. As discussed in chapter 4, the tit-for-tat behaviour is common when the game is symmetric. However, the game is no longer symmetric in these scenarios. The red team can not take advantage of the strategic decision making mechanism while the blue team can.

Figure 7.3 is the histogram of the fitness value by two fitness functions for the random walk. In order to facilitate the comparison between WISDOM-I and WISDOM-II, the figures use the same scale as in chapter 4.

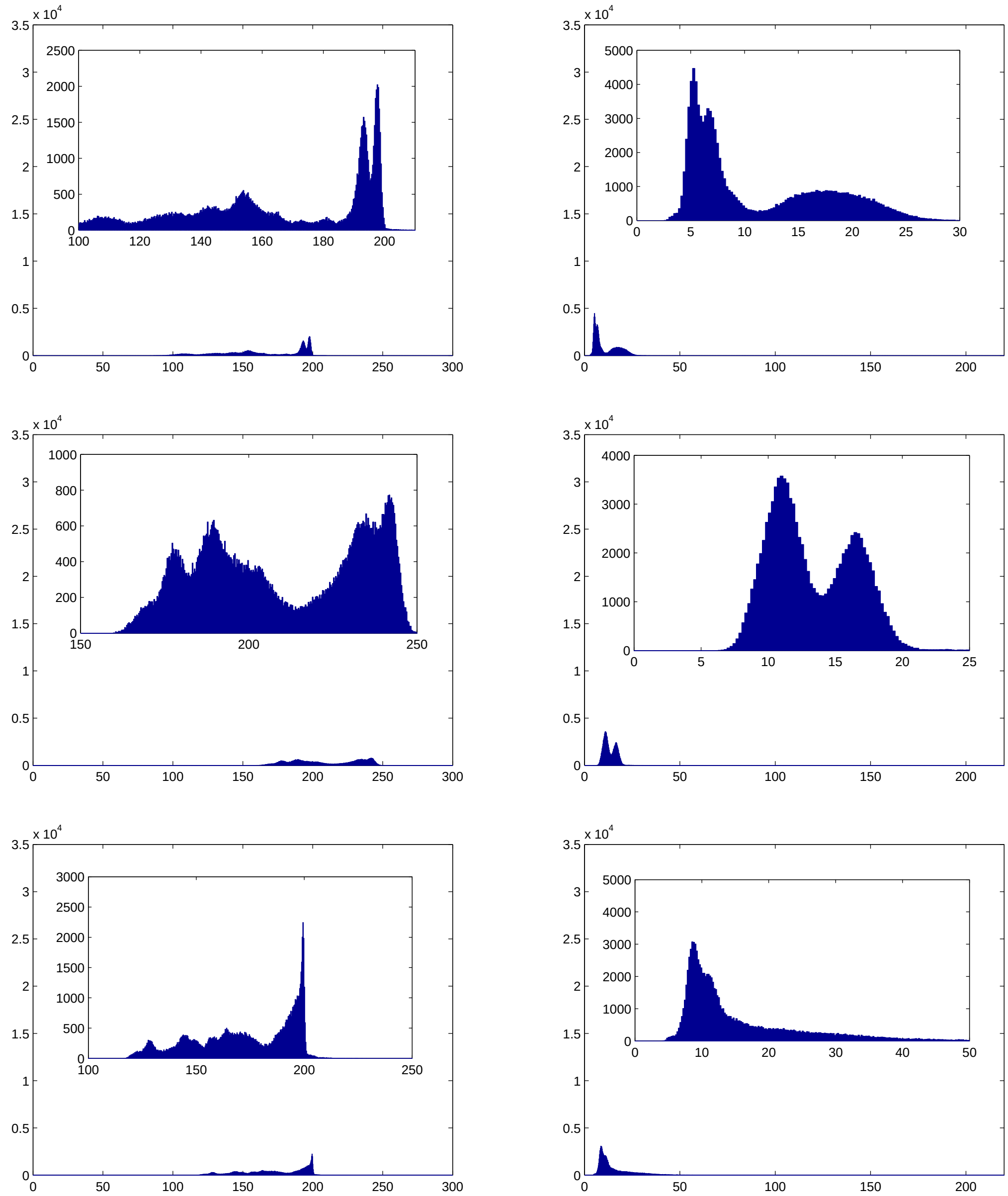


Figure 7.3: Histogram for random walk by using the average fitness (on left) and normalized fitness (on right). The order for top to bottom is: GOL, AGG, and DEF

For the average fitness function, the fitness value of most solutions found in the GOL scenario is between 100 and 200, in the DEF scenario it is between 120 and 200, and in the AGG scenario it is between 170 and 245. Only few solutions are GOOD solutions in either GOL or DEF scenario. For the GOL and DEF scenario, peaks around the point with the fitness value of 200 shows that there is a high probability to find a solution with the fitness value of 200. However in the AGG scenario, there are two small peaks with the fitness value of 190 and 240 respectively and with similar height. It suggests that it is more likely to find a solution with the fitness value of either 190 or 240 than any other values in the AGG scenario.

For the normalized average fitness function, the fitness value of most solutions found in all three scenarios is less than 20. In the GOL scenario, there is a big peak at the point with the fitness value of 10 and a small peak around the point with the fitness value of 20. It suggests that the fitness value of the solution found in the GOL scenario is more likely to be around 10. In the AGG scenario, almost all solutions fall into the fitness range of 10 to 20. That means it is almost impossible to find a solution with fitness value above 20. However for the DEF scenario, although there is a big peak between 10 and 20, there are a lot of solutions found between 20 and 40. There are two peaks in both GOL and AGG scenarios while there is only one peak in the DEF scenario. This implies that the effect of stochasticity is higher in both GOL and AGG scenario than that in the DEF scenario.

When compared with Figure 4.3 in chapter 4, one can see that only in the AGG scenario can a good solution be easily found for both WISDOM-I and WISDOM-II. For both GOL and DEF scenarios, there is an attractor at the point with the fitness value of 200 in WISDOM-I while there is no such attractor in WISDOM-II.

Looking at the normalized average fitness, for both WISDOM-I and WISDOM-II, there are many solutions found with fitness value less than 20. This implies that the solutions in all scenarios are very unstable for both systems.

Table 7.2 lists the results of the fitness landscape analysis using the information content approach. It is clear that the landscapes using both average fitness and normalized average fitness are very similar, and the landscapes of all three scenarios are also very similar in terms of information content ($H(\epsilon = 0)$), partial information content ($M(\epsilon = 0)$) and expected number of optima. This means the degree of ruggedness and modality of the landscape in these three landscapes is almost the same.

Table 7.2: The information theoretic measures using both fitness functions for random walk

		ϵ^*	$H(\epsilon = 0)$	$M(\epsilon = 0)$	Exp. # of Optima
Average Fitness	GOL	105.00 ± 5.27	0.41 ± 0.00	0.59 ± 0.00	2969.40 ± 13.44
	AGG	80.00 ± 0.00	0.41 ± 0.00	0.58 ± 0.01	2887.00 ± 29.40
	DEF	89.00 ± 9.94	0.42 ± 0.00	0.59 ± 0.01	2949.60 ± 27.10
Normalized Average Fitness	GOL	23.70 ± 1.49	0.41 ± 0.00	0.61 ± 0.01	3048.00 ± 32.19
	AGG	16.80 ± 1.32	0.41 ± 0.00	0.62 ± 0.00	3081.30 ± 18.99
	DEF	73.00 ± 3.50	0.40 ± 0.00	0.60 ± 0.00	3012.10 ± 15.27

However, in terms of information stability (ϵ^*), the landscapes are different. The highest information stability is obtained in the GOL scenario when using average fitness while the highest information stability is observed in the DEF scenario when using normalized average fitness. That is, the highest difference between two neighbouring peaks is observed in the GOL scenario using average fitness while the highest difference between two neighbour peaks is observed in the DEF scenario using normalized average fitness. One may also notice that the information stability is similar between the landscapes using average fitness and normalized fitness in the DEF scenario. This suggests that there are higher peaks in the DEF scenario than that in the GOL and AGG scenario.

When compared with the table 4.3 in chapter 4, the information content is similar between WISDOM-I and WISDOM-II while the partial information content is slightly higher in WISDOM-II than that in WISDOM-I. That is, the ruggedness is similar while the modality is higher in WISDOM-II. It can also be reflected by the number of expected optima.

The information stability in WISDOM-I is much higher than that in WISDOM-II according to the average fitness. This is caused by the same reason as the lower value of the fitness signal-worst in WISDOM-II. Since the fitness value of the worst solution found is higher in WISDOM-II than that in WISDOM-I, the difference between two neighbour solutions is obviously lower in WISDOM-II than in WISDOM-I. In terms of the normalized average fitness, it is consistent with the previous finding that there is no attractor in the GOL and DEF scenarios in WISDOM-II.

7.4 $(1+1)$ Evolution strategy

$(1+1)$ Evolution strategy (ES) adopts the same setup as in chapter 4. Firstly, it generates a solution at random which is considered the best solution found so far. The new solution is obtained by adding a random number drawn from a Gaussian distribution with zero mean and 0.1 standard deviation to each personality of the best solution found so far. If the new solution is better than or equal to the best solution found so far, the former replaces the latter. If not, a new solution is generated and the process continues until the maximum number of objective evaluations allowed is reached; after which, the algorithm terminates. The experiments are repeated ten times and each run is stopped after a total of 10,000 solutions have been generated. The experiments are performed for both fitness functions: the average fitness and the normalized average fitness.

Figure 7.4 shows the progression of the best solution found over time for each of the ten runs. The similar patterns can be observed in $(1+1)ES$ as in random walk. According to the average fitness, the overall best solution is found in the AGG scenario while it is found in the DEF scenario in terms of the normalized average fitness. The stochasticity is very critical in both GOL and AGG scenario since the fitness value of the normalized average fitness is pretty low in both scenarios. The improvement occurs only at the beginning stage in both GOL and AGG scenario while the improvement can still occur at the middle of the searching process in the DEF scenario. These findings are consistent with the findings based on the above fitness landscape analysis by random walk.

The fitness values of the overall best solutions found in all three scenarios using $(1+1)ES$ are almost the same as those using random walk. This implies the highly

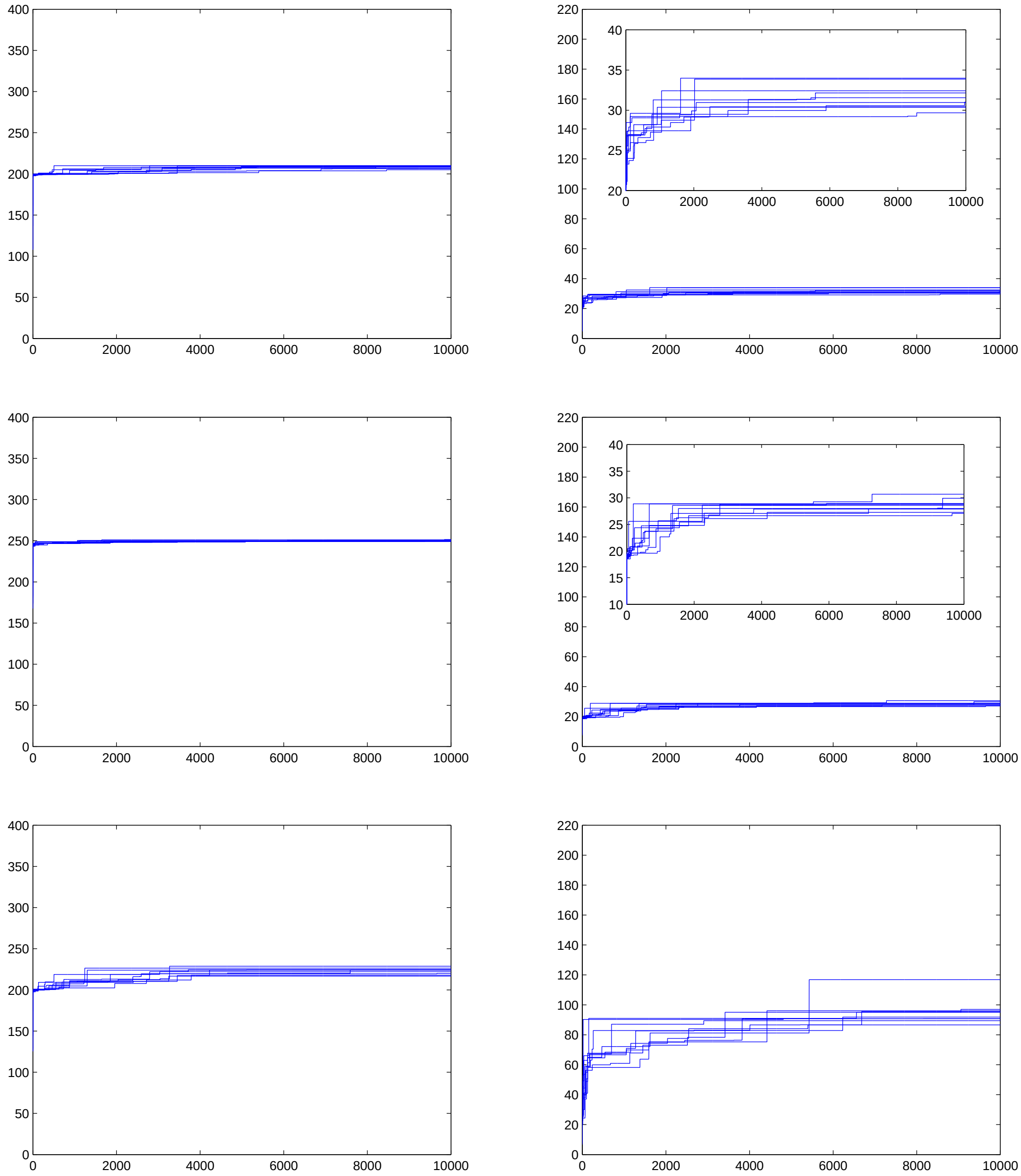


Figure 7.4: Best solution found over time by using the average fitness (on left) and normalized fitness (on right) for $(1+1)ES$. The order for top to bottom is: GOL, AGG, and DEF

exploitative search, such as $(1+1)ES$, does not improve the overall performance of the search. Note that as $(1+1)ES$ only keeps the best solution found so far, the histogram for $(1+1)ES$ (Figure 7.5) is not equivalent to that in random walk.

This finding is not consistent with that for WISDOM-I. In WISDOM-I, the fitness

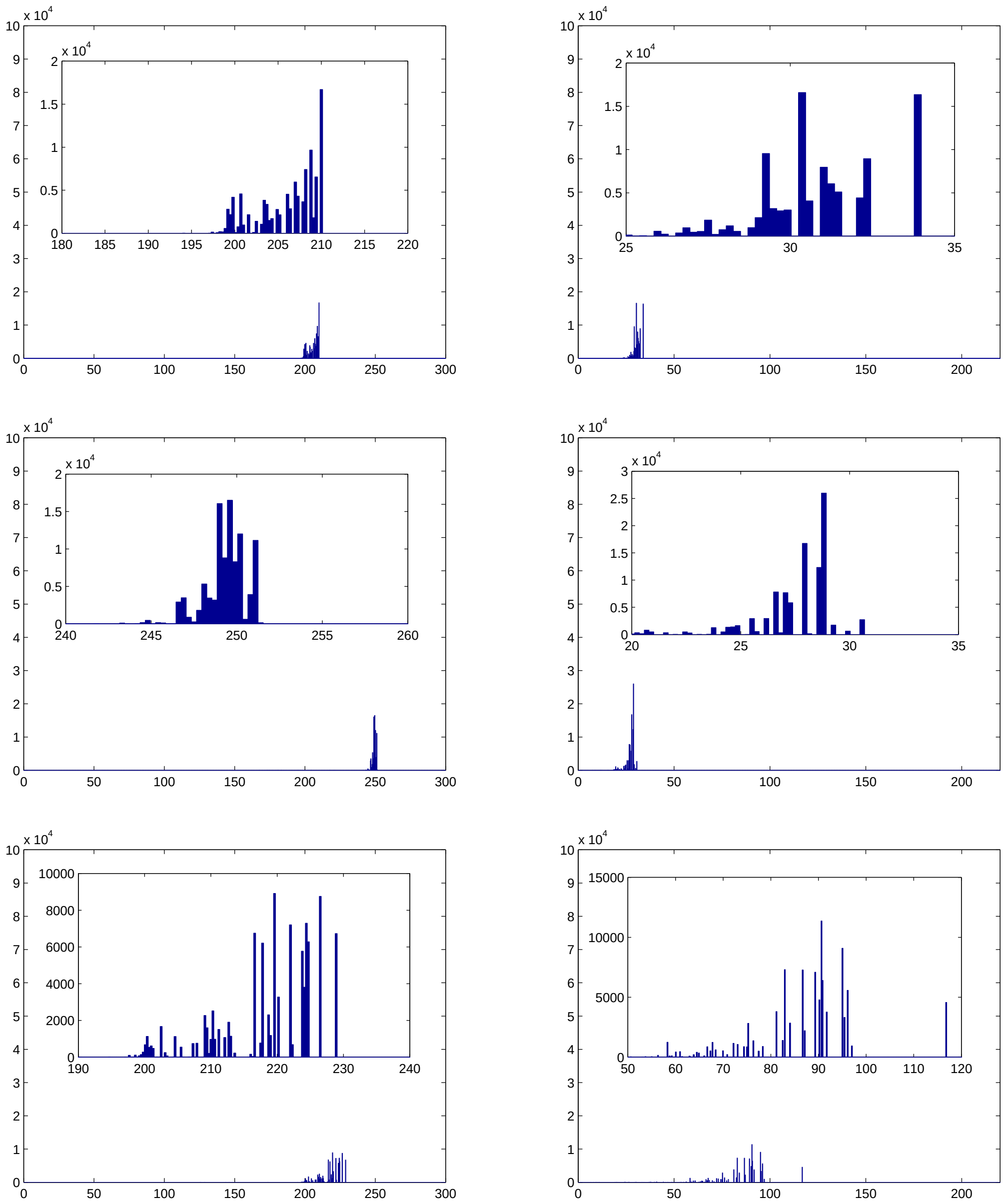


Figure 7.5: Histogram for $(1+1)ES$ by using the average fitness (on left) and normalized fitness (on right). The order for top to bottom is: GOL, AGG, and DEF

value of the overall best solution found in $(1+1)ES$ is higher than that in random walk. It seems $(1+1)ES$ can take advantage of the neighbouring information in WISDOM-I while it cannot in WISDOM-II. This implies that the peaks are clustered in WISDOM-I while they are not in WISDOM-II under the definition of the neighbourhood in this study. When looking at the normalized average fitness, the attractor at the value of 200 in the GOL and DEF scenario in WISDOM-I does not appear in WISDOM-II.

7.5 Multi-objective analysis

As discussed in chapter 4, one single fitness function, e.g. the average fitness or the normalized average fitness, could hide much information during the search for a good solution. A Pareto-based multi-objective evolutionary approach is adopted for further analysis as in chapter 4.

The experiments with similar settings as before are conducted for both random walk and $(1+1)ES$ with two objectives: minimizing the damage of the blue team (Equation 4.16) and maximizing the damage of the red team (minimizing the remaining health of red team) (Equation 4.17).

Figure 7.6 is the scatter diagram and pareto-dominance diagram for all three scenarios. The left column is drawn from random walk. The pareto-optimal is far below the diagonal in the AGG scenario for both random walk and $(1+1)ES$. This means that blue may cause more damage to red in this scenario. However, in the GOL scenario, almost all pareto-optimal solutions found are above the diagonal. That is, it is almost unlikely for blue to find a solution to win the game, which means the red damage is larger than the blue damage. In the DEF scenario, most pareto optimal solutions found are around the diagonal and some are below the diagonal.

If drawing the diagonal line to split all the solutions found in each scenario, one can see that there are more than half of the solutions found in the AGG scenario where blue can win the game. In the GOL scenario, blue almost cannot find a solution to win the game while in the DEF scenario, although it is hard, blue can still find solutions to win the game.

The figures at the right side are based on $(1+1)ES$. In order to faciliate analysis, the solution space has been partitioned into six areas (Figure 7.7). When compared

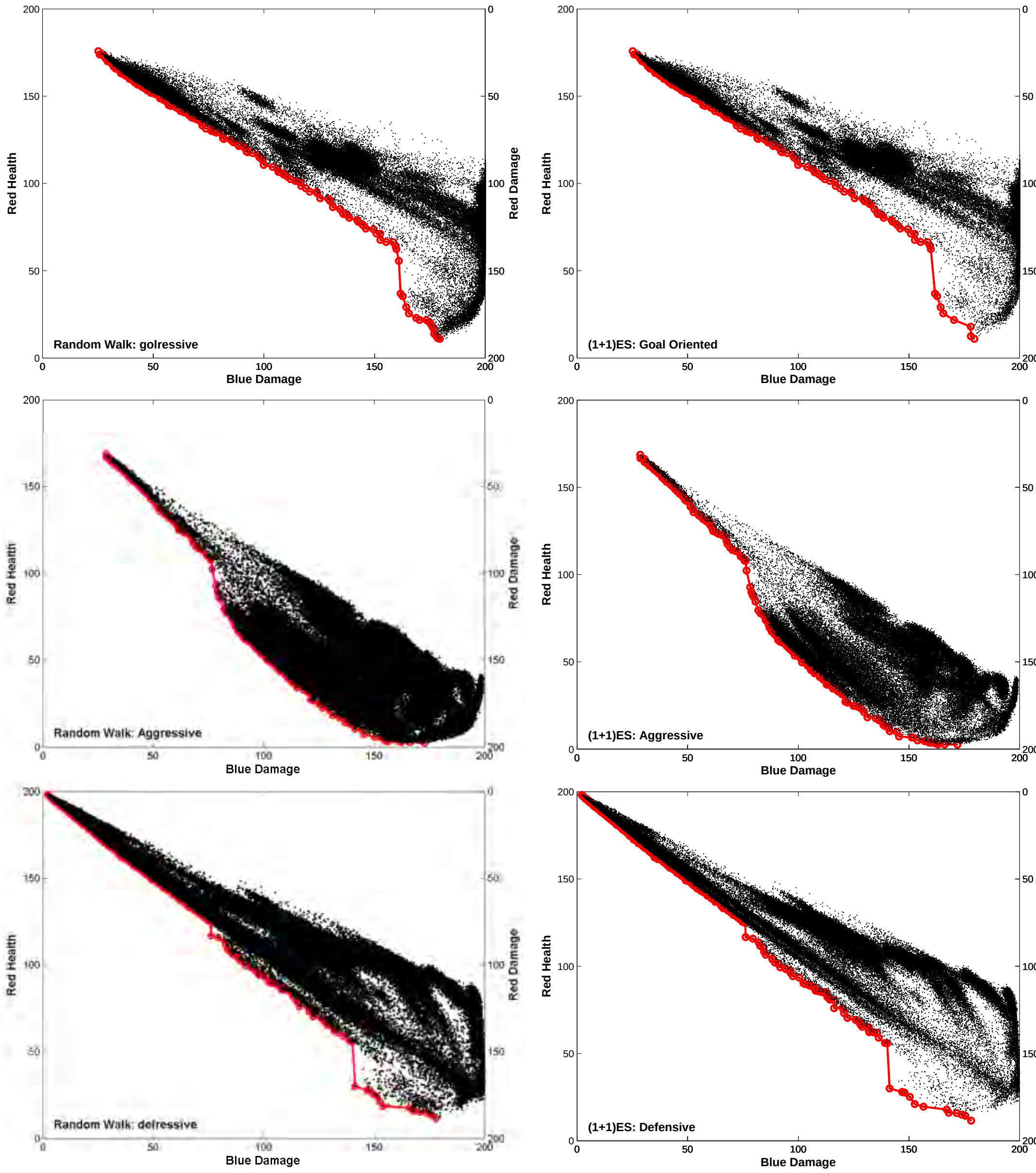


Figure 7.6: Scatter and Pareto-front diagram for random walk (on left) and (1+1)ES (on right). The order for top to bottom is: GOL, AGG, and DEF

with the left figures, one may find that they are very similar. Thus, the highly exploitative search, $(1 + 1)ES$, does not show any advantages over the highly explorative search, random walk. This is consistent with the findings in section 7.4. One can also see that only in the AGG scenario, can blue damage all of the red for both random walk and $(1 + 1)ES$. This finding is the same as that in WISDOM-I. However, in WISDOM-I, the performance of $(1 + 1)ES$ is slightly better than that

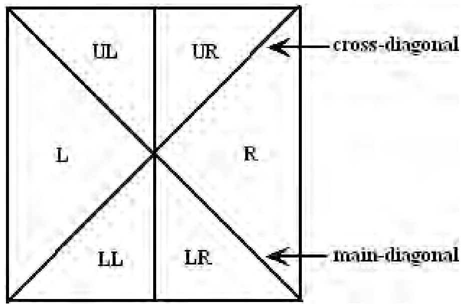


Figure 7.7: Topology of solution space

of random walk. The main difference between WISDOM-I and WISDOM-II is that, in WISDOM-II for all three scenarios, the explored solution space is a strip zone around the diagonal where the damage of blue and red team is very similar. However in WISDOM-I, for the GOL and DEF scenario, the explored space is the area of UL, UR and R, and more solutions are fallen into the area of UL according to the density. For the AGG scenario, most solutions are fallen into the area of LL, LR and R for random walk while the solutions cover the area of L, LL, LR and R for $(1+1)ES$. It shows that in WISDOM-I, for both random walk and $(1+1)ES$ the blue may be completely damaged with a very little damage of the red, or even no damage of the red. However in WISDOM-II, such situations do not appear. The worst case in WISDOM-II happens in the GOL scenario that the blue is totally damaged by the red while the red is at least damaged half. It confirms the previous finding that the signal-worst in WISDOM-II is smaller than that in WISDOM-I.

When comparing the pareto-optimal sets between two systems, one may see that the pareto-optimal set from WISDOM-I for all three scenarios are better than those from WISDOM-II. This may be because the modality of the fitness landscape of WISDOM-II is higher than that of WISDOM-I. The density of the solutions found within the solution space can be considered as a measure of effectiveness of search. Combining with above analysis of solution topology, it seems that not very bad and also not very good solutions can be easily and effectively found by using WISDOM-II while both very bad and very good solutions may be found by using WISDOM-I.

7.6 Summary

In this chapter, a fitness landscape based on WISDOM-II is analysed by using same approach as in the chapter 4. Three scenarios (strategies) are chosen for the red team: Goal-oriented (GOL), Aggressive (AGG) and Defensive (DEF) while the strategy (a vector of personalities) of the blue team is evolved.

In order to facilitate the comparison of the fitness landscapes generated by WISDOM-I and WISDOM-II, most unique features of WISDOM-II are turned off except both tactical and strategic decision making mechanisms. In WISDOM-II, the decision variables are represented with a vector of 18 real numbers representing different characteristics of personalities. Same objective function and the fitness function are adopted as in the chapter 4. Each configuration is evaluated 100 times, each for 500 time steps.

Table 7.3: Comparison of the fitness landscape generated by WISDOM-I and WISDOM-II

	WISDOM-I	**WISDOM-II**
Influence of stochasticity	high	
Signal-worst	high	low
Tit-for-Tat	common	not common
Attractor	at the fitness of 200	No
Information content	similar in both systems	
Partial information content	low	high
Information stability	high	low
# of expected optima	low	high
Performance of search algorithm	exploitation is better	similar
Progress of search	most improvements occurs at the beginning	
Solution clustering	yes	no
Topology of solution space	UL for GOL LL, LR, R for AGG UL for DEF	above main-diagonal for GOL centred on main-diagonal for AGG above main-diagonal for DEF

For all three scenarios, the fitness landscapes are rugged and multi-modal. The difficulty of the blue team in finding a good solution (a combination of the personalities for the blue agents) to win the game is largely dependent on the strategy the red team takes. The characteristics of the fitness landscape change when the strategy of the red team changes. The degree of difficulty for the blue team to find a good solution increases in the order of: AGG, DEF and GOL. All these findings are consistent with those in WISDOM-I. However, there are also many differences between the fitness landscapes generated by WISDOM-I and WISDOM-II (see Table 7.3). As discussed above, the strategic decision making mechanism has been identified as the major cause leading to some differences between landscapes.

In the next chapter, some military analyses are conducted to exemplify the usage of WISDOM-II.

Chapter 8

Analysis of Military Operations in Urban Terrains

8.1 Introduction

One of the most significant changes in the last century is the urbanization of the world's population. Such global urbanization has largely influenced military operations and led to shifting from operations in open country to operations in urban centres. The urban environment represents a set of unique challenges to soldiers and leaders (Aragon 2001; Phillips et al. 2001).

First, urban environments, e.g. streets and buildings, provide a three-dimensional ground threat which does not appear in the traditional open country conflict. In an urban conflict, to control an area means to control a volume instead of controlling the surface of the ground in an open terrain conflict. Multi-level buildings, sewer and subway systems, all provide the third dimension.

Second, the technological advantage of the weapon system may be mitigated. One common advantage of a weapon system is its capability to have a long target range including both minimum and maximum target distance. However in an urban warfare, the military force may not have sufficient distance between itself and its enemy to take advantage of these long-range weapons.

Third, the close-quarter conditions of urban warfare may increase vulnerabilities. The close-quarter may reduce the ability of a force to disperse itself. When area-target weapons, such as grenades and mortar rounds, are used, they may lead to high casualty rates.

Fourth, civilians are intermixed with hostile units. This makes urban warfare quite different from traditional battles (Grau and Kipp 1999). A military force needs to be able to distinguish between civilians and hostile units and then to destroy them.

Finally, the tactics of a military unit may be affected by hazardous materials, such as power lines and generators, natural gas lines and stations, chemicals, etc. Research on real wars shows that MOUT exhibits different behaviours from operations in a non-urban environment (Desch 2001). All these differences require one to develop new and innovative tactics and strategies for MOUT.

Recent research shows that simulation, especially agent-based simulation, is a robust and valuable tool to study MOUT (Aragon 2001; Brown et al. 2003). However, through the study of current simulation model capabilities in six aspects: direct fire, indirect fire, mobility, search and target acquisition, tactical communications and wide area surveillance, Crino (2001) argued that the current simulation models are inadequate to fully investigate MOUT and that all these six aspects need to be enhanced.

In this chapter, the impacts of force size, firepower (direct fire), communication and strategic planning on attrition in a series of simulated urban combats are investigated and analysed through a parametric study using WISDOM-II. First, a set of baseline scenarios are developed to study the effect of force size and firepower on attrition and are compared with the findings of Davies et al. (2004) which used EISTein to see if similar patterns can be captured by WISDOM-II and EINSTEin. After that, three sets of extended scenarios are created where the capability of the blue force is enhanced with communication, a strategic planning mechanism, or both. Finally, a single simulation is used to exemplify how to understand the simulation through the reasoning log.

8.2 Scenario configuration

8.2.1 Baseline scenarios

In order to study the role of force size, firepower (direct fire), communication and strategic planning, a set of baseline scenarios is created where all unique features in WISDOM-II are turned off, such as the artificial hospital and strategic decision making mechanism (strategic planning). Both blue and red forces are made up of homogenous agents which are only equipped with a point-to-point direct weapon system. There is no communication for either blue and red forces. The behaviour of agents are completely based on their personalities. This configuration is consistent with that in Davies et al. (2004).

8.2.1.1 Force size and firepower

For a Lanchester equation based combat model with direct weapon only as discussed in chapter 2, the attrition should follow the coupled differential equations known as the Lanchester Square Law (LSL):

$$\begin{cases} \frac{dB_s}{dt} &= -\alpha_r R_s(t) \\ \frac{dR_s}{dt} &= -\alpha_b B_s(t) \end{cases} \tag{8.1}$$

where $B(t)$ and $R(t)$ represent the force size of the blue and red respectively, and α_b and α_r are analogous to the firepower for the blue and red force respectively. The firepower is represented by the single shot kill probability P_{kill} in this chapter as in the study of Davies et al. (2004). The solution to Equation 8.1 should satisfy:

$$\int \alpha_r R(t) dR = \int \alpha_b B(t) dB \tag{8.2}$$

and one may see that:

$$\alpha_b B(t)^2 - \alpha_r R(t)^2 = k \tag{8.3}$$

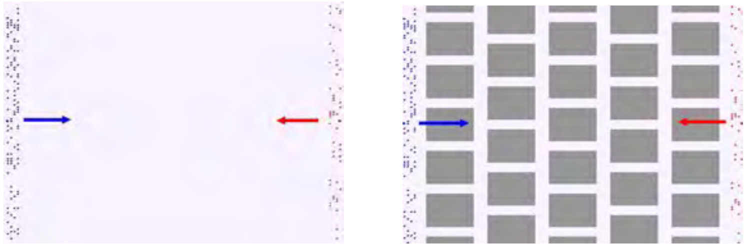
where k is a constant. When $k = 0$, it can be said that the two forces are evenly matched (Davies et al. 2004). In this study, five evenly matched forces are defined as shown in Table 8.1. The role of firepower and force size can then be studied by comparing the experimental results with those expected from the LSL.

Table 8.1: Configurations of the force size ratio and P_{kill}

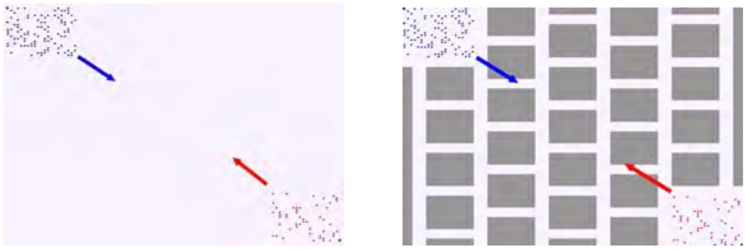
Blue force size ($B_s(0)$)	Blue P_{kill} (α_b)	Red force size ($R_s(0)$)	Red P_{kill} (α_r)
50	0.01	50	0.01
100	0.01	50	0.04
150	0.01	50	0.09
200	0.01	50	0.16
250	0.01	50	0.25

8.2.1.2 Movement direction

Davies et al. (2004) found that some force units modelled in EINSTein cannot navigate around obstacles when the movement direction is orthogonal to obstacle faces (Figure 8.1(a)), and diagonal simulation may mitigate this side effect. Therefore, diagonal simulation is adopted in this study as shown in Figure 8.1(b). The results for orthogonal simulation are attached in Appendix A.



(a) Orthogonal simulation



(b) Diagonal simulation

Figure 8.1: Initial starting position and direction of force movement

8.2.1.3 Terrain

To avoid undesired terrain advantages for both blue and red teams, the urban terrain is designed symmetrically. The density of building blocks is varied in the environment by varying the size of each building block, while the coordinates of the centre of each block remained unchanged. The distance between neighbouring building blocks decreases when the size of the building block increases. To avoid long line of sight at the beginning of the simulation, blocks are staggered. The resultant corridors are where the mini-battle is likely to occur. The width of the corridors, which is the number of agents able to stand abreast cross the corridor, is used as an indicator of the size of the battlefield and the type of the urban environments.

Six types of urban environments are developed as shown in Figure 8.2 and Table 8.2. In order to make the environment symmetric, the size of environment is 99×99 in both Gap3 and Gap1.

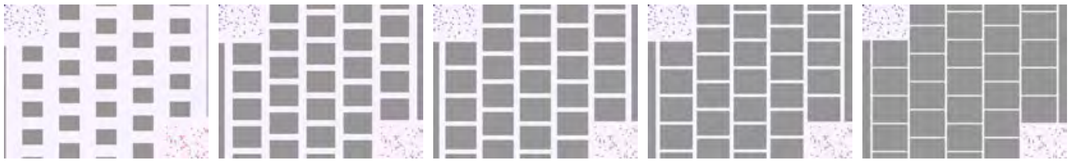


Figure 8.2: Urban terrains used in the study. From left to right, the corridor width: 8, 4, 3, 2 and 1.

Table 8.2: Configurations of terrains used in the study

Terrain	Size of Env.	Block size	Corridor width	Urban coverage
Open	100×100	-	-	0
Gap8	100×100	10	8	26.82%
Gap4	100×100	14	4	54.06%
Gap3	99×99	15	3	65.04%
Gap2	100×100	16	2	70.72%
Gap1	99×99	17	1	79.92%

8.2.1.4 Agent characteristics

Both blue and red force have the same type of agents except the single shot kill probability (P_{kill}), which is defined with the force ratio in Table 8.1. Four types of agents are developed as shown in Table 8.3. All types of agents try to attack their enemy and reach their goal. Agents can either avoid (dispersed) or ignore (non-dispersed) their own force. The vision and firing range of agents can be short (30) or long (99). All agents behaviour is completely determined by their personalities.

Table 8.3: Agent characteristics for each force size configuration and terrain

Type	Vision and firing range	Preference to own force
Non-dispersed short range	30	Neutral
Non-dispersed long range	99	Neutral
Dispersed short range	30	Avoid
Dispersed long range	99	Avoid

To ensure that forces become engaged approximately in the middle of the battlefield, the agents are set to always try to reach the goal, which is the headquarter of the opposite force. The agents also try to attack their enemy if any enemy is visible within their vision range. These settings ensure that engagement does not occur at the edge of the urban environment (Davies et al. 2004). Therefore, discontinuities of the urban terrain are unlikely to influence the outcome of the simulation.

8.2.2 Extended scenarios

With future warfighting concepts tending to focus on asymmetric warfare in urban environments, it was vital to test the role of networked forces and strategic planning in these environments. Based on the baseline scenarios, three sets of extended scenarios are created to study the role of communication and strategic planning. All configurations in the baseline scenario are maintained except the capability of the blue force. For each of these three sets of scenarios, the capability of the blue force is supplemented with communication and/or strategic planning as depicted in Table 8.4.

Table 8.4: Extended scenarios

Extended scenario	**Features**
A	Baseline scenarios + communication in the blue force
B	Baseline scenarios + strategic planning in the blue force
C	Baseline scenarios + communication and strategic planning in the blue force

The strategic planning mechanism used in this study is defined by five parameters: mission type, resolution of the battlefield, frequency to send out commands, advance threshold and defend threshold. The mission type for both forces in all scenarios is “occupy”, which means the force tries to occupy the goal predefined in the scenario. The commander develops the strategic plan based on a low resolution view of the battlefield. In this study, the overall battlefield is abstracted into 5 by 5 hyper cells, based on which the commander makes plans. The frequency to send out commands is set to 5. This means the commander sends out commands to the groups every 5 time steps. The advance threshold is set to 1 while the defend threshold is set to 1.2. For the details of how the strategic planning mechanism works, refer to section 6.2.6.3.

8.3 Experimental setup

According to the above configuration, there are 120 individual simulations in total for both baseline and extended scenarios, which are: 6 terrain types x 4 agent types x 5 force size ratios. For each simulation, 200 runs are conducted, each of which consists of 200 time steps. The outcome of combat is measured by the normalized loss exchange ratio (NLER) as shown in equation 8.4.

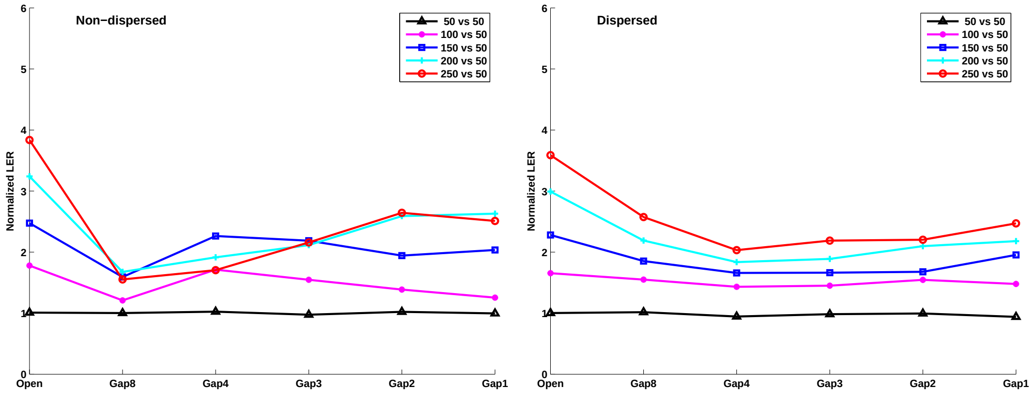
$$NLER = \frac{\frac{BlueCasualty}{RedCasualty}}{\frac{B_s(0)}{R_s(0)}} \tag{8.4}$$

From the perspective of the blue team, the lower, the better.

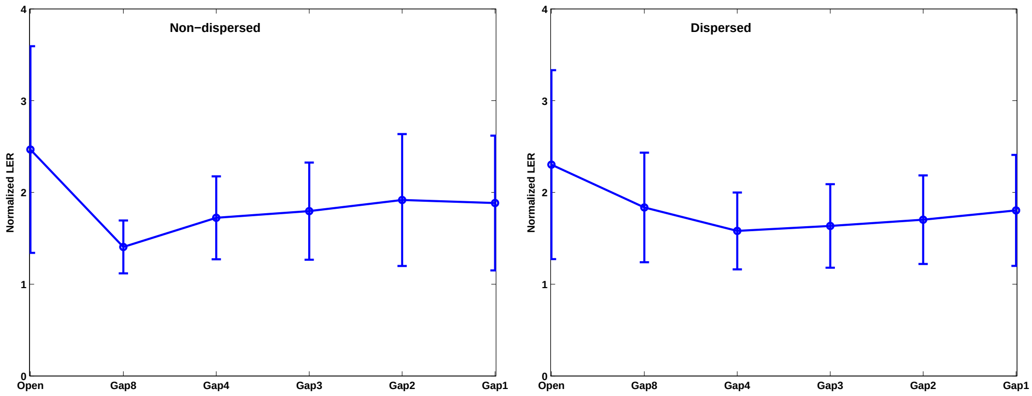
8.4 Results and analysis

8.4.1 Baseline scenarios

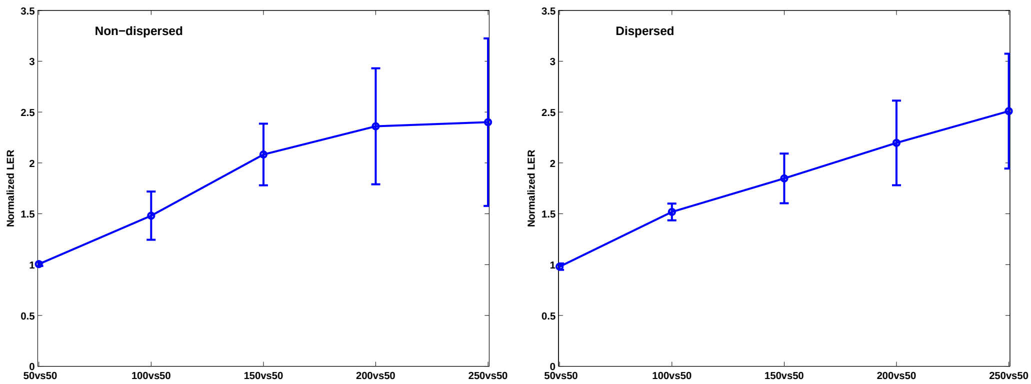
Figures 8.3 and 8.4 are the outcomes for both dispersed and non-dispersed force in the baseline scenarios for the short and long vision and firing range respectively. All figures show that the experimental outcomes are very different from what is expected from LSL. According to LSL, the NLER should be unity for a combat of two evenly matched forces. However the results suggest that a small force with higher P_{kill} is better than a large force with low P_{kill}, as shown in Figures 8.3(c) and 8.4(c). The difference between large force size configuration and the smallest force size configuration is increased as the coverage of blocks increases except in the cases when the force has a range of 30 in the open terrain and the dispersed force with the range of 99 in the terrain of Gap8 (see Figures 8.3(a) and 8.4(a)). One interpretation is that since the larger force has low P_{kill}, it needs to highly coordinate the behaviours of its combatants in order to maximize the casualty of its enemy. In general, even in an open terrain, it is very hard for a force to be fully coordinated. Therefore, as the coverage of blocks increases, blocks could largely hinder the coordination in the larger force. The performance of that force in turn decreases.



(a) NLER for different urban terrains and different force size configurations

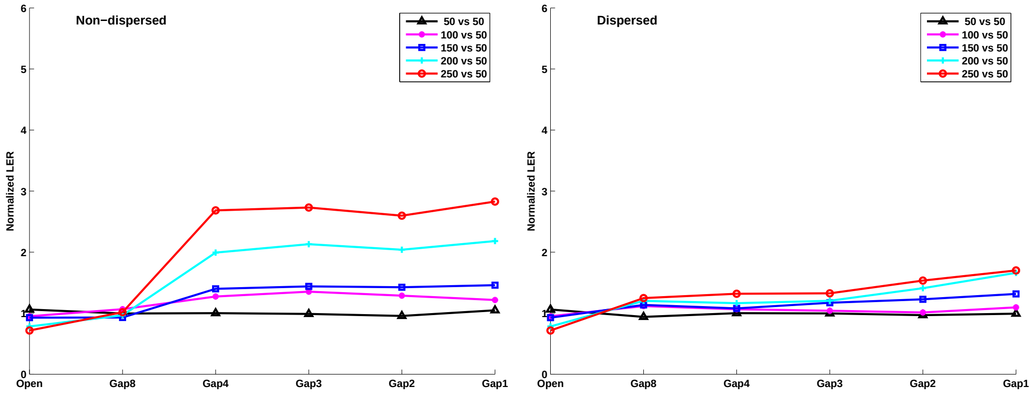


(b) Average NLER across different force size configurations for different urban terrains

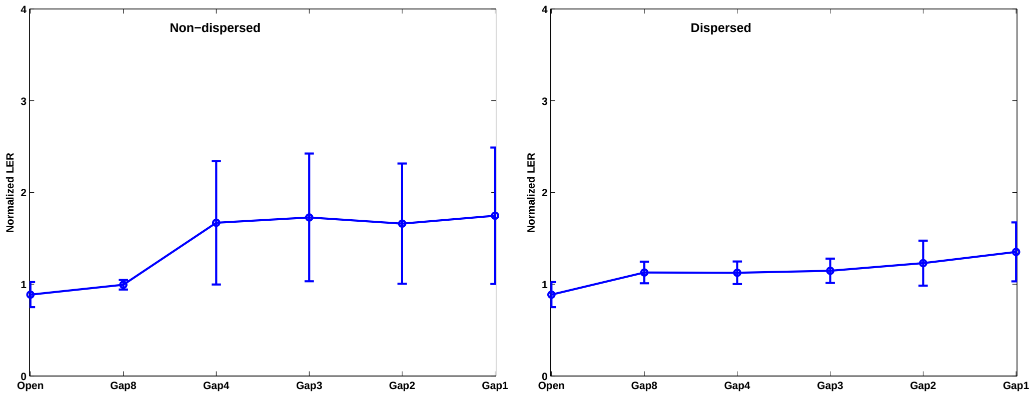


(c) Average NLER accross different urban terrains for different force size configurations

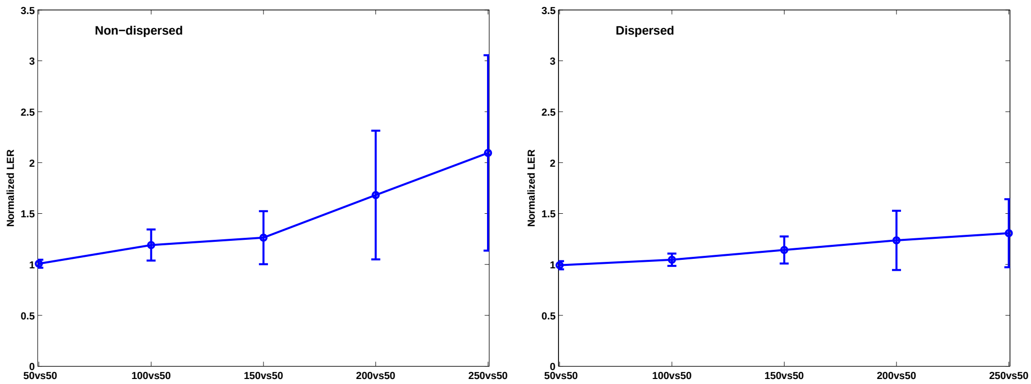
Figure 8.3: Outcomes for both non-dispersed (left) and dispersed force (right) with a short range (30) vision and firing in the baseline scenarios



(a) NLER for different terrains and different force size configurations



(b) Average NLER across different force size configurations for different terrains



(c) Average NLER across different terrains for different force size configurations

Figure 8.4: Outcomes for both non-dispersed (left) and dispersed force (right) with a long range (99) vision and firing in the baseline scenarios

However, this interpretation is not supported by the results (Figures 8.3(a) and 8.3(b)). The performance of the larger force is even worse in an open terrain than that in an urban terrain. This may be because of the firing algorithm adopted in

WISDOM-II, where the agents always try to fire at their closest enemy. If more than one enemy exists in a close proximity, the agents will choose one to shoot at random. And if an agent shoots its enemy, then it will not move at this time step. Therefore engagement actually occurs between the agents at the front of each force. For example, for the case of 250 blue agents against 50 red agents, the NLER is above 3.5 regardless of whether the force is dispersed or not. For the diagonal simulation, the initial position of the blue force is within the upper-left square as shown in Figure 8.1(b). When both forces approach their goals, the force ratio between agents at the front line of the blue force and the red force is less than $5 : 1$. However, the firepower ratio between blue and red team still keeps at $1 : 5$. Therefore, it leads to that the NLER is larger than 1, which is expected by LSL. This is also the reason why the NLER is a little bit less for a dispersed force than that for a non-dispersed force. For a dispersed force, the right graph in Figure 8.3(a) shows that the NLER is higher for Gap8 than that for Gap4. The wider the corridor, the more agents being accommodated in the corridor. The more visible the blue agents are, the more dispersed the blue force. Therefore the ratio of the agents at the front line between the blue force and the red force is less for Gap8 than that for Gap4.

When inspecting Figure 8.4(a) more closely, one may find that the outcome of the simulation is different from that predicted by LSL even in an open terrain with long range vision and firing, which means that an agent may see everywhere and fire at anyone. LSL requires fire to be uniformly distributed over the surviving units (Przemieniecki 2000). However, agents in WISDOM-II always fire at the closest enemy. To investigate what causes the difference, a number of experiments are conducted by modifying the firing algorithm so that agents uniformly choose among the visible enemies to shoot at instead of firing at the closest enemy. The results with the modified firing algorithm is presented in Table 8.5. It shows that with the modified firing algorithm, the NLER is around 1 which is as expected by LSL. With the firing algorithm adopted by WISDOM-II, the NLER varies from 0.72 to 1.06. However, we will continue with the original firing algorithm of WISDOM-II since it is more logical. The rationality behind the firing algorithm in WISDOM-II is as follows:

1. The closer the enemy, the more threat from it;
2. It is not reasonable to shoot an enemy that is just behind another enemy using a direct weapon.

Table 8.5: The NLER of different force size configuration by different firing algorithms with long firing range (99)

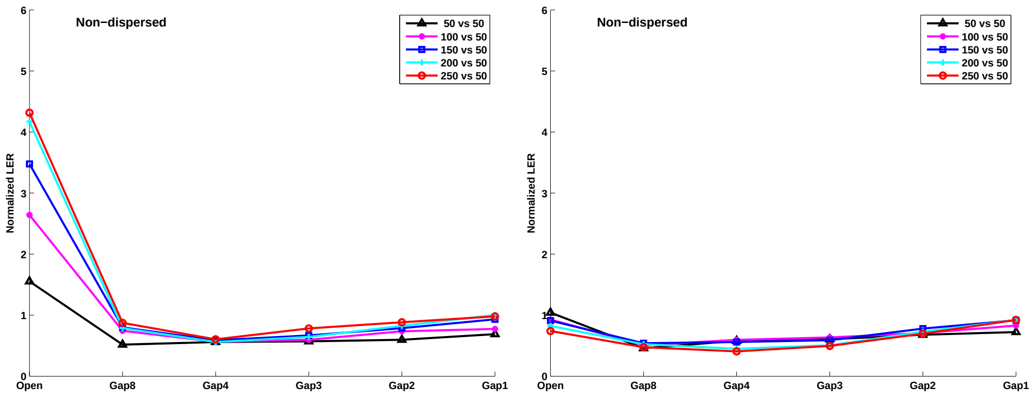
Firing algorithm	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50
Modified	0.98	1.03	1.05	1.03	1.02
Original	1.06	0.95	0.93	0.78	0.72

8.4.2 Extended scenario A - the blue force with strategic planning and without communication

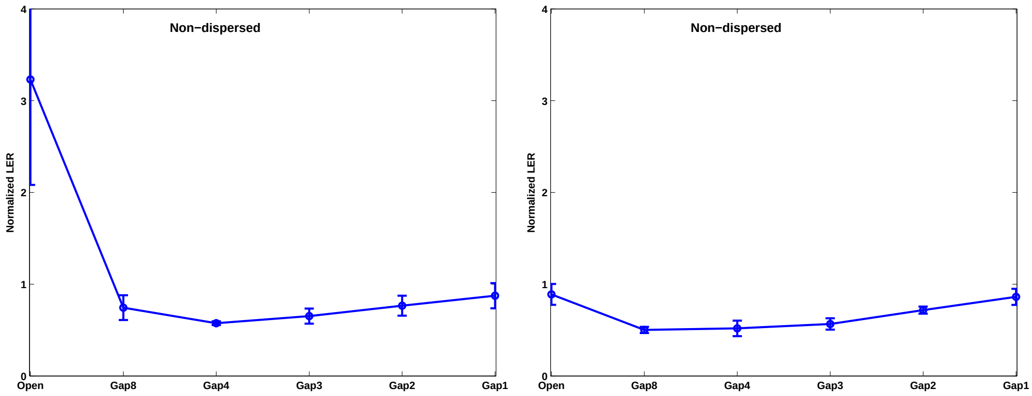
In the extended scenario A, the blue force adopts a strategic planning mechanism to guide the behaviours of its agents. All other configurations are maintained similar to the baseline scenario.

Figure 8.5 represents the outcomes of the non-dispersed force for both short range (left) and long range (right) vision and firing in the extended scenario A. When comparing Figures 8.5(a) and 8.5(b) with the corresponding graphs in the left side of Figures 8.3(a), 8.3(b), 8.4(a) and 8.4(b), one may find that the performance of the blue force is largely improved with the strategic planning mechanism for all urban terrains except for the open terrain. The blue force wins the game for all urban terrains (NLER is less than 1) while it was defeated by the red force in the baseline scenarios. This implies that high coordination between a large number of agents may overcome the disadvantages of low firepower.

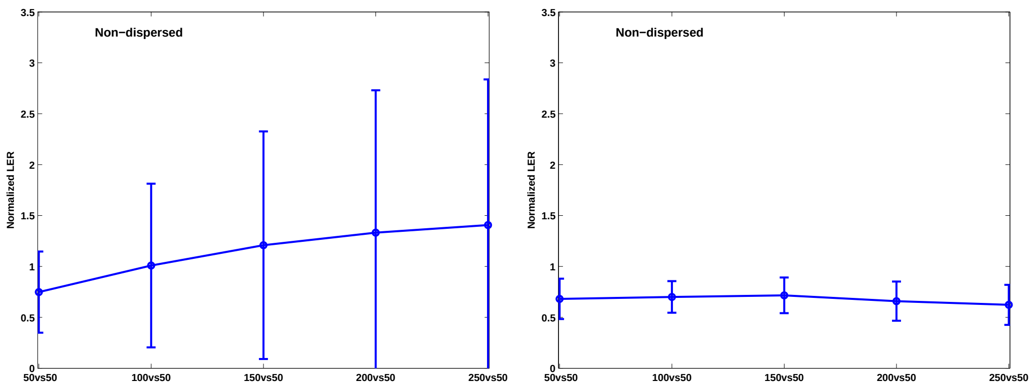
For the open terrain, agents can detect and fire at any agent in the battlefield when the range is long (99). Coordination is not crucial to the outcome of combat in this scenario. Therefore, there is not much difference between the cases with and without



(a) NLER for different terrains and force size configurations



(b) Average NLER across different force size configurations for different urban terrains



(c) Average NLER across different terrains for different force size configurations

Figure 8.5: Outcomes of the non-dispersed force for both short range (left) and long range (right) vision and firing in the extended scenario A.

a strategic planning mechanism. For the case of short range (30), the performance of the blue force with the strategic planning is even worse in open terrain. This is because the more highly clustered the blue agents are, the less the number of agents at the front line, which can be engaged with the red agents. Figure 8.6 is two screen dumps of the simulation where the blue force uses (left) or does not use (right) the strategic planning mechanism in the extended scenario A. The blue force is at the upper-left corner while the red force is at the bottom-right corner. It is clear that the blue force without the strategic planning (right) is more scattered than that with the strategic planning (left). Therefore, the number of blue agents at the front line in the left side graph is less than that in the right side graph. This suggests that this type of strategic planning, which may be suitable when agents posses long range vision, may not be suitable for when agents posses short range vision.

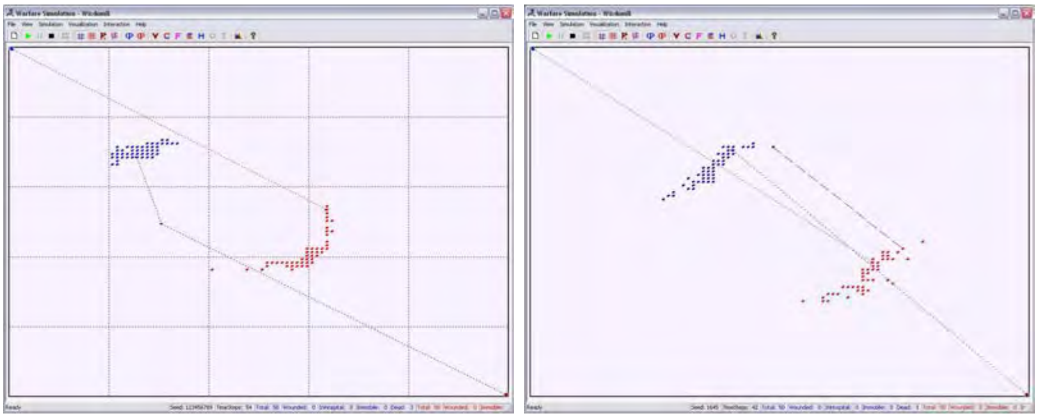


Figure 8.6: Two screen dumps of the simulation for the force with (left) and without (right) the strategic planning in the extended scenario A. The blue force is at the upper-left corner while the red force is at the bottom-right corner.

When comparing Figure 8.5(b) with Figures 8.3(b) and 8.4(b) correspondingly, one may note that the difference between highly dense terrain and light dense terrain is much less when the blue force adopts the strategic planning mechanism than that in the baseline scenarios. This implies that the scattering effect of blocks is offset by the clustering effect of a strategic plan. Figure 8.5 also shows that the NLER is almost constant when the range is long (99). The longer the range, the more information about the battlefield is obtained. Based on this global view of the battlefield, the strategic plan can effectively and efficiently coordinate the behaviours of the blue agents even if there is a large number of agents. Therefore, there is not much difference among different urban terrains.

Figure 8.7 represents the outcomes of the dispersed force for both short range (left) and long range (right) vision and firing in the extended scenario A. When compared with Figure 8.3 and 8.4 respectively, one may find that they share similar patterns and the corresponding NLER is only a little bit less with the strategic planning mechanism than that without a strategic planning mechanism. The reason why the performance of the blue force is not largely improved as observed for the non-dispersed force above is because the clustering effect of the strategic planning mechanism is largely offset by the dispersed personality of the agents in this scenario.

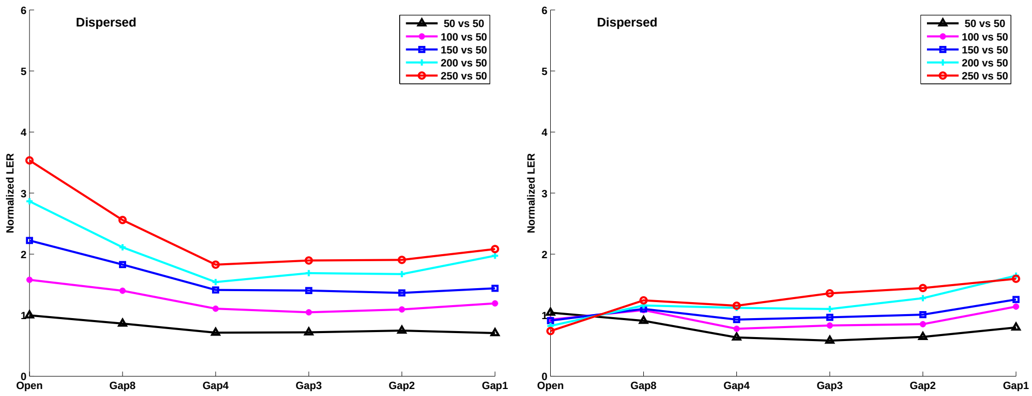
8.4.3 Extended scenario B - the blue force with communication and without strategic planning

In the extended scenario B, there is no strategic planning mechanism for the blue force. But, the blue agents may or may not communicate with other blue agents while there is no communication within the red force. The communication range varies from 0 to 20. In order to study the effect of communication on the outcome of combat, for both blue and red forces, the vision range is fixed to 30 while the firing range is fixed to 99. Other settings are similar to those in the baseline scenario.

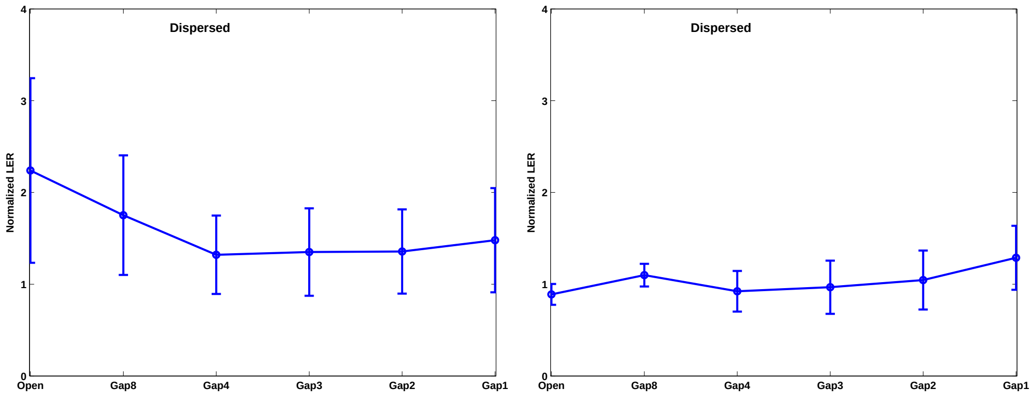
Figure 8.8 depicts the average NLER of both non-dispersed (left) and dispersed force (right) in the extended scenario B for different terrains, force size configurations and communication ranges. One may observe the same trends as in the baseline scenario and the extended scenario A. The performance of the small size force is better than that of the large size force. The performance decreases when the urban coverage of blocks increases.

It is also not surprising that the NLER decreases when the communication range increases. In this scenario, the longer the communication range, the better.

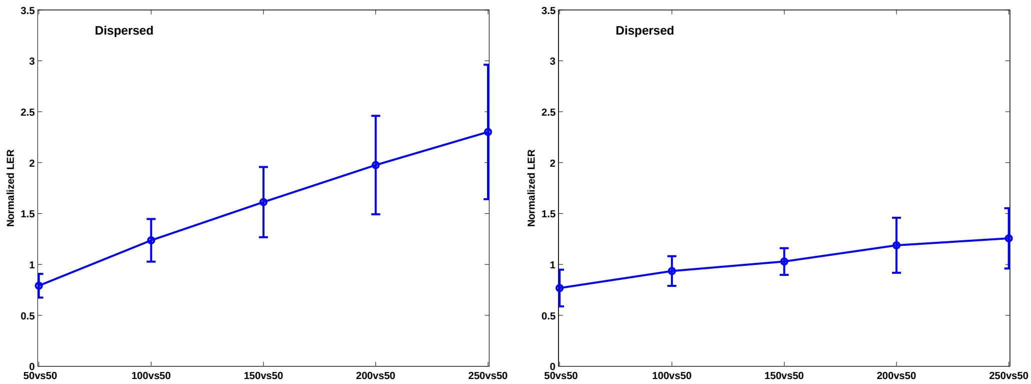
When comparing the results between dispersed and non-dispersed forces, one may



(a) NLER for different terrains and force size configurations



(b) Average NLER across different force size configurations for different urban terrains



(c) Average NLER across different terrains for different force size configurations

Figure 8.7: Outcomes of the dispersed force for both short range (left) and long range (right) vision and firing in the extended scenario A.

find that the NLER is always higher for the dispersed force than that for the non-dispersed force. This is because the density of the agents is lower for a dispersed force than that for a non-dispersed force. Therefore, the total number of agents able

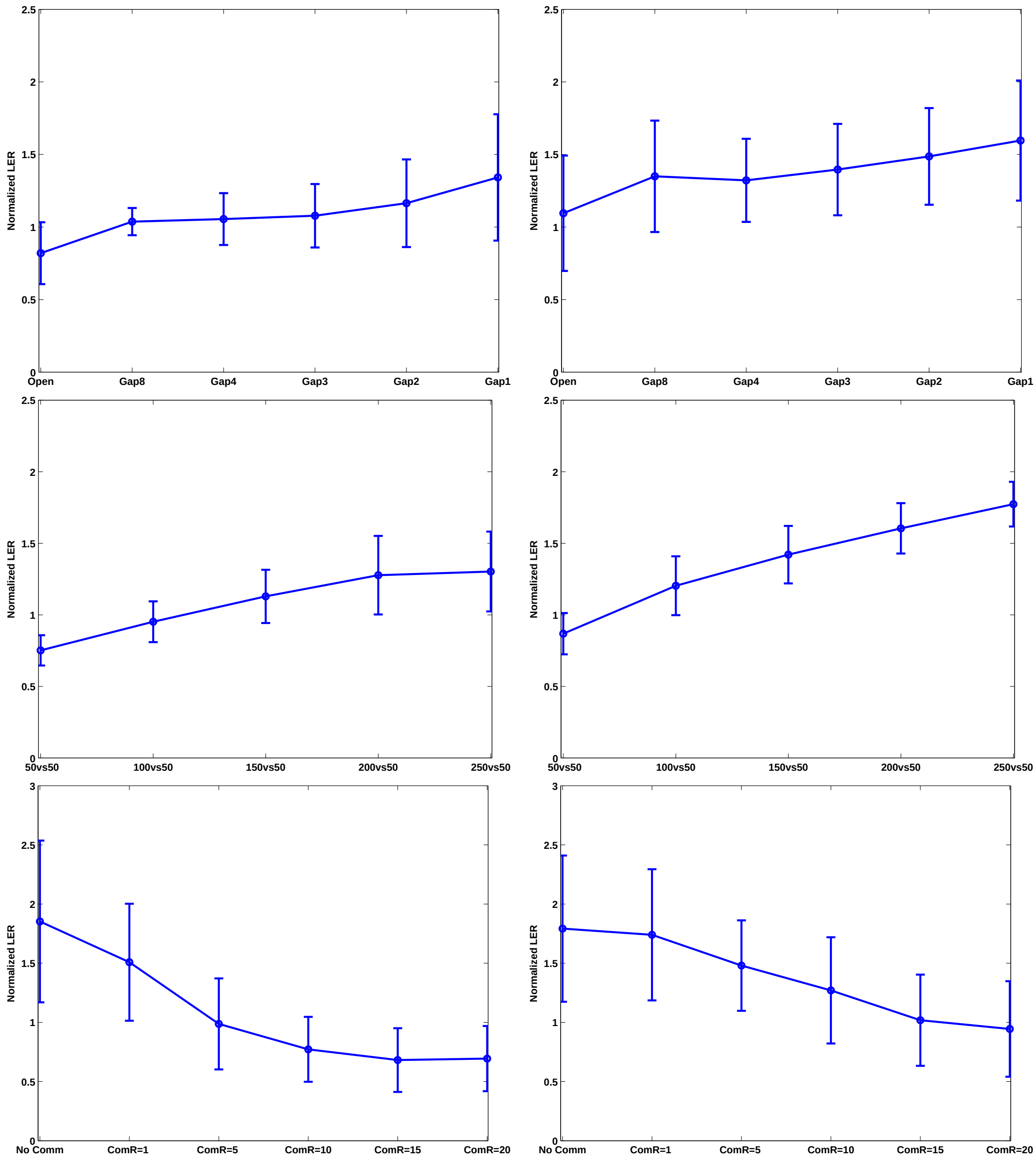


Figure 8.8: Average NLER of both non-dispersed (left) and dispersed force (right) in the extended scenario B for different terrains, force size configurations and communication ranges from top to bottom respectively.

to communicate within a certain communication range is lower in a dispersed force than that in a non-dispersed force. In turn, the NLER is higher.

Figure 8.9 presents the average NLER across different force size configurations of both non-dispersed (left) and dispersed (right) forces for different terrains and communication ranges. For both non-dispersed and dispersed blue forces, the performance is improved when the communication range increases for all terrains.

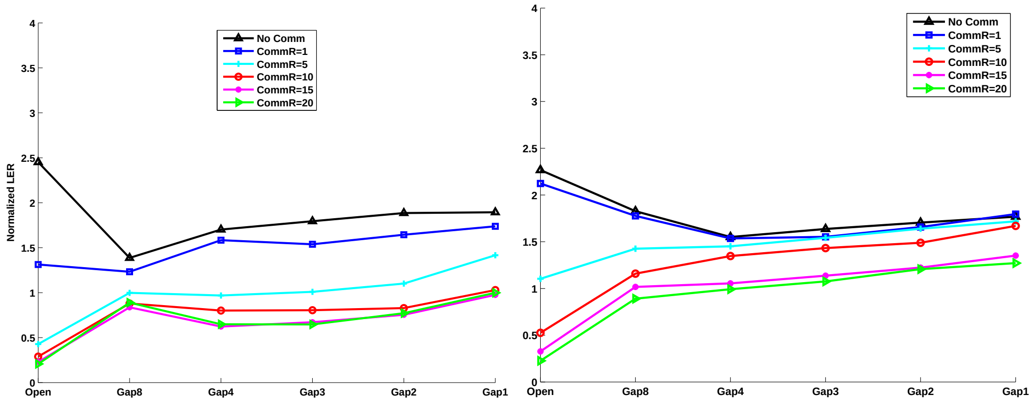


Figure 8.9: Average NLER across different force size configurations of both non-dispersed (left) and dispersed (right) forces for different terrains and communication ranges

Figure 8.10 represents the improvement made by communication (the NLER difference between the case with communication and the case without communication) for both non-dispersed and dispersed forces for different terrains. For the non-dispersed force (the left graph of Figure 8.10), the largest improvement occurs in the open terrain. The improvement then decreases in Gap8, increases again in Gap4 and then keeps almost constant to Gap2. Finally it decreases again in Gap1. This pattern appears for all communication ranges. In the open terrain, the lack of blocks facilitates coordination among the blue agents. Therefore the blue force may fully take advantages of communication and the NLER then largely decreases. When the terrain changes from no obstacles to Gap8, there are two effects of the blocks. One is to prevent the agent from shooting at its enemy if the enemy is behind a block. The other is to spread the agents in the terrain. Therefore, the number of agents within the communication range decreases. That is, the degree of coordination decreases. With these two effects, the improvement decreases from the open terrain to Gap8. Another reason might be that the NLER is already low in Gap8 when there is no communication (see the left graph in Figure 8.9). It is very hard to improve the performance of the force further, which may already be taking full advantage of its capability. This is also the reason why the improvement in Gap4 is larger than that in Gap8. If agent A sends information about its enemy to agent B and agent B can fire at that enemy, we call it "effective communication". From Gap4 to Gap2, although the size of the block increases, the effective communication may not be reduced. It leads to that the improvement within these three environments are similar. The reason why the improvement decreases in Gap1 is because there is no space to accommodate more agents in the corridor in Gap1 even if the agents can communicate with many of their friends.

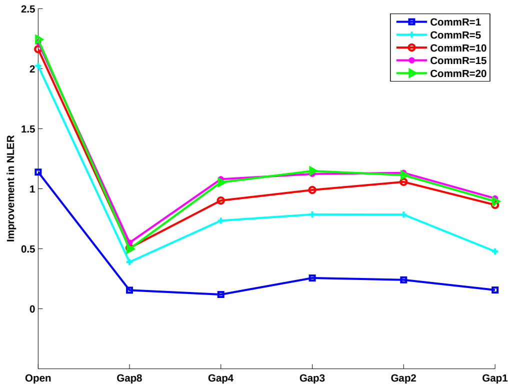


Figure 8.10: Improvement made by communication of both non-dispersed (left) and dispersed (right) forces for different terrains and communication ranges in the extended scenario B

One may also find that the improvement made by each unit of communication range decreases when the communication range increases from 0 to 20. Even when the communication range is 20, there is no improvement when compared to that in the communication range of 15. This is because the effective communication does not increase when the communication range increases above a certain point, e.g. 15 in this scenario. For a non-dispersed force, the big jump occurs between communication range of 1 and 5 since the agents are closer to each other within a limited space, e.g. a corridor, as all agents try to move to their goal and enemy. Therefore, the number of agents within the communication range suddenly increases when the communication range increases from 1 to 5.

For the dispersed force (the right graph of Figure 8.10), the largest improvement occurs when the communication range increases from 5 to 10. Since a dispersed force is spread over the battlefield, the diameter of the force cluster is larger than that of a non-dispersed force. Therefore, a big jump happens between the communication ranges of 5 and 10.

Figure 8.10 also shows the improvement decreases for the dispersed force as the terrain moves from open terrain to Gap4, which is different from that for the non-dispersed force. When the urban coverage of blocks is not too high, the effective communication largely reduces when the coverage of blocks increase. Therefore the improvement decreases. However, when the coverage of the blocks continues to increase, e.g. from Gap4 to Gap1, the effectiveness of communication does not change. This makes the improvement unchanged across these urban terrains.

Figure 8.11 depicts the average NLER across different terrains of both non-dispersed (left) and dispersed (right) forces for different force size configurations and communication ranges in the extended scenario B. For both non-dispersed and dispersed forces, the performance becomes worse when the force size increases or the communication range decreases. This is consistent with our previous findings.

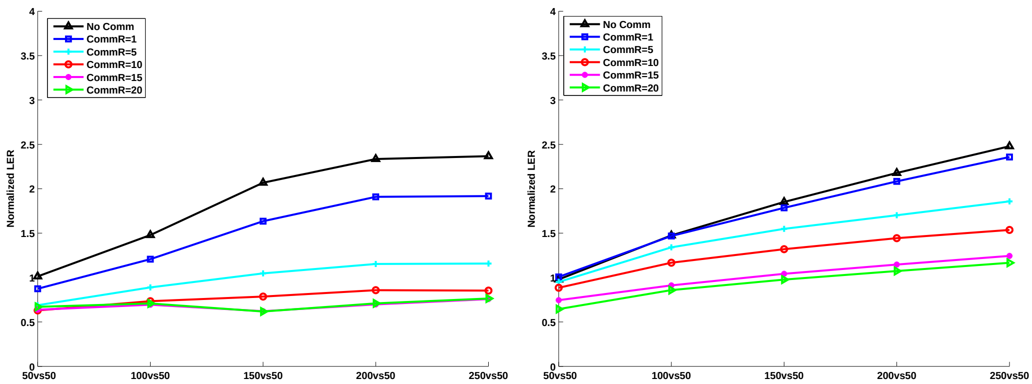


Figure 8.11: Average NLER across different terrains of both non-dispersed (left) and dispersed (right) forces for different force size configurations and communication ranges in the extended scenario B

Figures 8.12 represent the improvement made by communication of both non-dispersed (left) and dispersed (right) forces for different force size configurations and communication ranges in the extended scenario B. For the same communication range, the improvement increases when the force size increases. This is because the number of agents within the communication range increases as the force size increases.

For the same force size, the improvement made by each unit of communication

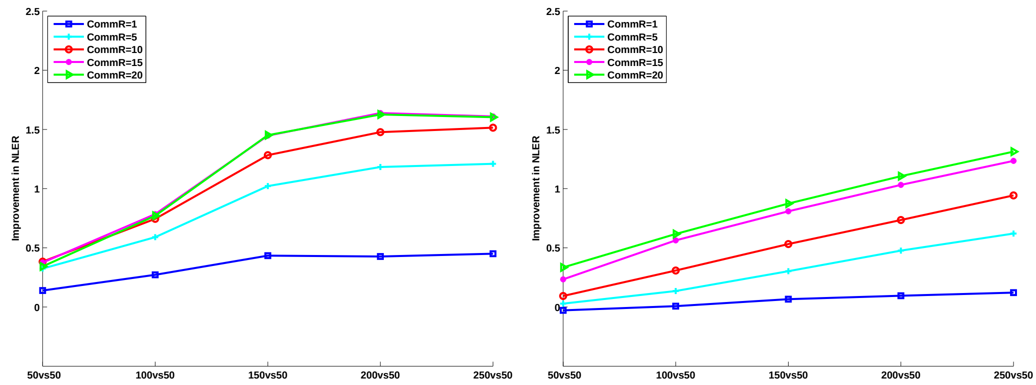


Figure 8.12: Improvement made by communication of both non-dispersed (left) and dispersed (right) forces for different force size configurations and communication ranges in the extended scenario B

range decreases when the communication range increases. The reason is that with the spreading effect of the blocks, the number of agents able to communicate does not increase at a constant rate when the communication range increases. Therefore, the improvement made by each unit of communication range decreases when the communication range is already long.

For the dispersed force (the right graph in Figure 8.12), the improvement made by communication is linear to the force size. This is because for a dispersed force, the agents approximately uniformly spread over the battlefield. Therefore, as the force size increases from 50 to 250, the number of agents within the communication range linearly increases.

8.4.4 Extended scenario C - the blue force with communication and strategic planning

In the extended scenario C, the blue force adopts a strategic planning mechanism (defined in subsection 8.2.2) to guide the behaviours of the blue agents. All other configurations is maintained similar to the extended scenario B.

Figure 8.13 depicts the average NLER of both non-dispersed (left) and dispersed (right) forces for different terrains, force size configurations and communication ranges in the extended scenario C. When compared to Figure 8.8, the performance of the non-dispersed blue force is even worse with a strategic planning than that without it in the open terrain. This is because of the same reason as discussed in subsection 8.4.2.

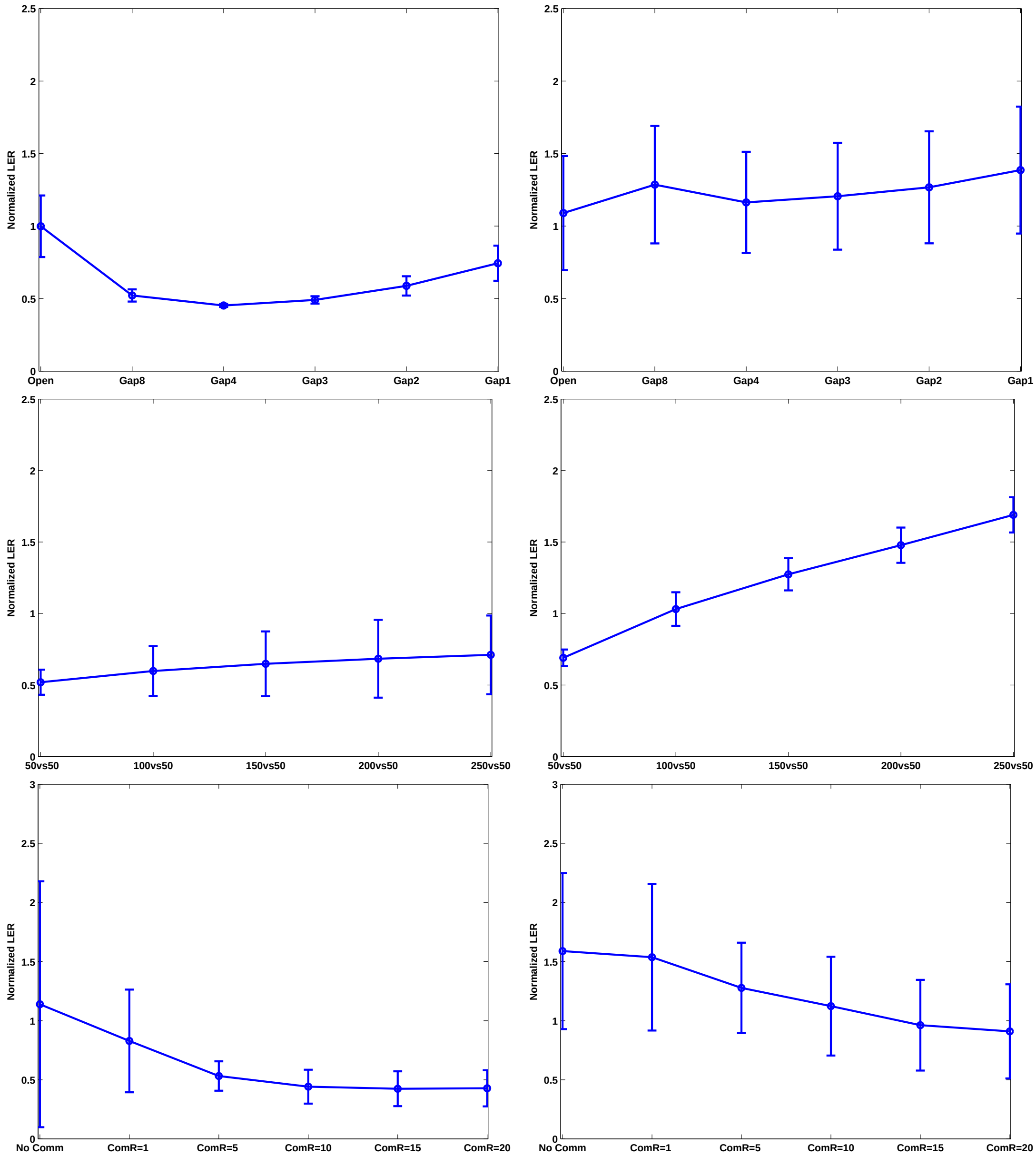


Figure 8.13: Average NLER of both non-dispersed (left) and dispersed (right) forces in the extended scenario C for different terrains, force size configurations and communication ranges from top to bottom respectively

The improvement (difference of NLER between the extended scenario C and B) increases when the size of the non-dispersed force increases while it is almost constant for the dispersed force. For a dispersed force, the more the agents know each other, the more scattered the force. For a larger size force, this spreading effect is stronger than a smaller size force. Therefore, the clustering effect of the strategic planning is largely offset by the spreading effect for a larger size dispersed force. For a non-dispersed force, this spreading effect does not occur. The larger the force, the more effective the strategic planning.

The improvement decreases when the communication range increases for both non-dispersed and dispersed forces. The strategic planning and communication are two forms of coordination. If the performance of the blue force is already improved through communication, it might be very hard to improve it further through another means of coordination, e.g. strategic planning.

Figure 8.14 represents the average NLER across different force size configurations of the non-dispersed force for different terrains while Figure 8.15 represents the average NLER across different terrains of the non-dispersed force for different force size configurations in the extended scenario C. When comparing Figure 8.14 with the left side of Figure 8.9 and Figure 8.15 with the left side of Figure 8.11 correspondingly, one may find that the performance difference between short range and long range communication becomes less when the blue force adopts a strategic planning mechanism for all terrains and force size configurations. For the short range communication, strategic planning may largely increase the degree of coordination among agents. However, for the long range communication, the degree of coordination is already very high. It is very difficult to improve it further by strategic planning. Therefore the performance difference decreases.

Figures 8.14 and 8.15 also show that the performance of the blue force is similar for different terrains (except open terrain) and force size configurations. It implies that the higher coordination, the less influence of terrain and force size configuration on the outcome of combat.

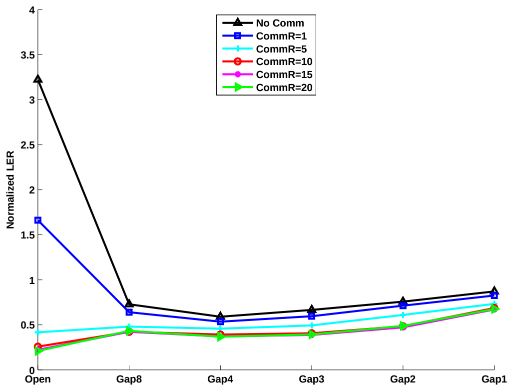


Figure 8.14: Average NLER across different force size configurations of the non-dispersed force for different terrains in the extended scenario C

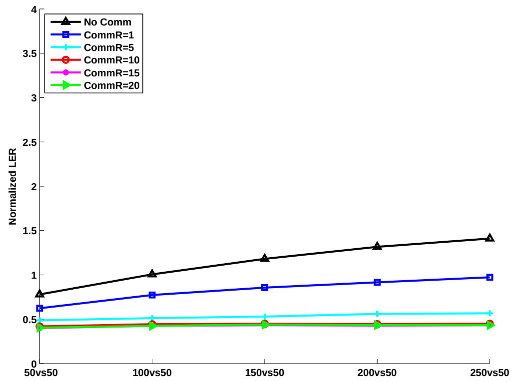


Figure 8.15: Average NLER across different terrains of the non-dispersed force for different force size configurations in the extended scenario C

Similar patterns can be observed when comparing Figure 8.14 with the left side of Figure 8.9 and Figure 8.15 with the left side of Figure 8.11. The force performance is improved when strategic planning is adopted for all cases except the cases without communication or with a short range communication in the open terrain. This is caused by the same reason as discussed in subsection 8.4.2.

Figure 8.16 represents the average NLER across different force size configurations of the dispersed force for different terrains while Figure 8.17 represents the average NLER across different terrains of the dispersed force for different force size configurations in the extended scenario C. When compared with Figure 8.9 and 8.11 correspondingly, one may note that the improvement is smaller for the dispersed force than that for the non-dispersed force. This is because the spreading effect of dispersed personality of the agents offsets the clustering effect of strategic planning.

8.4.5 Analysis on the reasoning log

A single simulation run is used to demonstrate how WISDOM-II interprets the simulation through the reasoning log. In this scenario, the vision range and firing range for both forces is 30 and 99 respectively. The second force size configuration

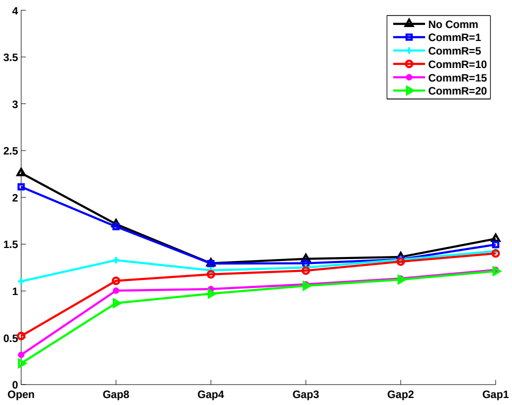


Figure 8.16: Average NLER across different force size configurations of the dispersed force for different terrains in the extended scenario C

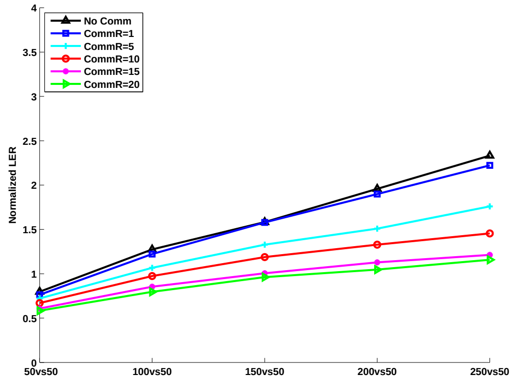


Figure 8.17: Average NLER across different terrains of the dispersed force for different force size configurations in the extended scenario C

in Table 8.1 is selected. The blue force has 100 agents with the P_{kill} of 0.01 while the red force has 50 agents with the P_{kill} of 0.04. The communication range of the blue force is 10 and the strategic planning defined in subsection 8.2.2 is adopted by the blue force while there is no communication and strategic planning mechanism in the red force.

The reasoning log is attached in Appendix B. From the reasoning log, one may see that there are three stages in combat. The first stage is the pre-engagement stage for both forces up to the time step 37. Normally in this stage, each force organizes its agents and approaches the goal. So the reasoning log shows how both blue and red forces approach their goals. For example, the statement that “19: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.” in the reasoning log shows that both blue group 1 and red group 2 approach to its goal at the time step 19. Since only blue force adopts a strategic planning mechanism, there is no order sent out to the red force. It only occurs for the blue force. For example, in the reasoning log there is a statement that “20: An order has been sent to the leader of group 1 in the blue team to move toward (30, 50)” at the time step 20. With the strategic planning mechanism, the blue force commander makes plans for all groups in the blue force and guide them to reach their goal. At the time step 20, for example, the commander asks the group 1 to move to the position of (30, 50). From Figure 8.18, one may see that the average degree of the blue communication increases and the average shortest path length of the blue communication network decreases at the first stage. The commands and information can then be quickly sent to the agents. This implies that after the first stage the blue force is ready to fight against the red force.

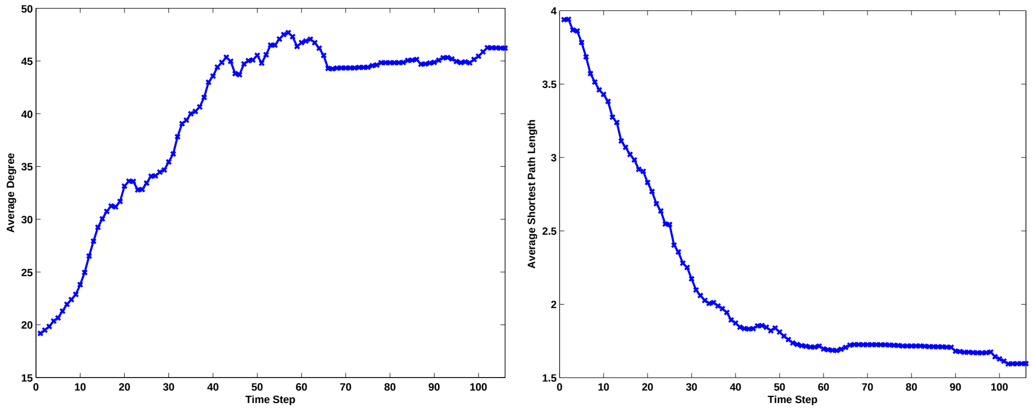


Figure 8.18: Average degree (left) and average shortest path length (right) of the blue communication network

The second stage is the engagement stage. This stage is the main course of combat from the time step 38 to 64. In this stage, the forces fire at each other. The combat is explained to the user in the reasoning log through three aspects as follows:

- What is happening? The user may know what is happening at each time step through the reason log. For example, “38: The engagement starts”, “50: An order has been sent to the leader of group 1 in the blue team to move toward (50, 50)” and “51: Without any loss, the red team causes damage of 10 to the blue force”.
- How does it happen? The reasoning log can interpret how it happens. For example, ”46: The agents in the red team are spreading their fire to achieve maximum damage in the blue team”, “47: The situation awareness of blue team is gained mainly through its sensor component. The situation awareness

of red team is gained mainly through its sensor component", and "52: The agents in the blue team are spreading their fire to achieve maximum damage in the red team". To have the statement at the time step 46 and 52, the standard deviation of the in-degree of the the blue or red agents in the firing network must be less than the predefined threshold, e.g. 0.5, in the scenario (Figure 8.19).

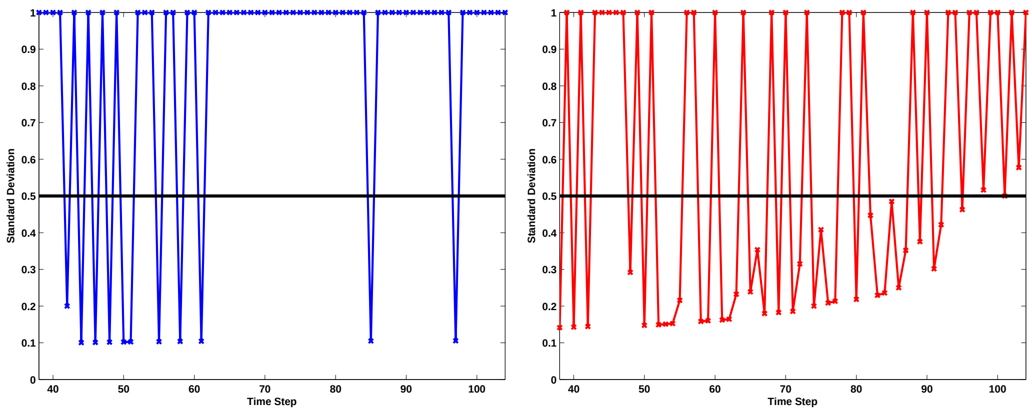


Figure 8.19: The standard deviation of the in-degree of blue agents (left) and red agents (right) in the engagement network

- Why does it happen? Based on the real-time reasoning engine, the system presents the reason why it happens to the user. For example, "55: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of friend on force level" and "55: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team". From Figures 8.18, 8.20 and 8.21, one may see that the damage of red force increases while the average degree of the blue communication network increases and the average shortest path length decreases for the time step 50 to 55.

The last stage is the epilogue of combat. Since at the end of the second stage one force normally has totally controlled the whole battlefield, the controlling force keeps attacking the remaining opposing agents with very small damage to its own (Figures

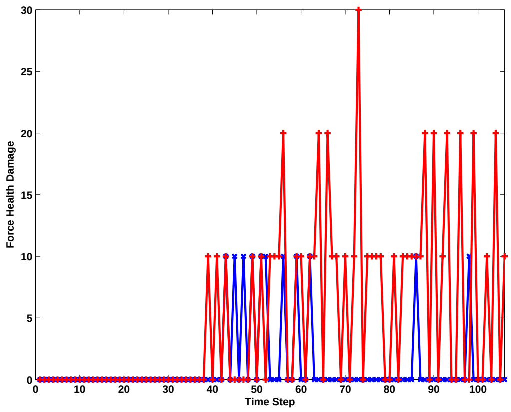


Figure 8.20: Force damage in each time step

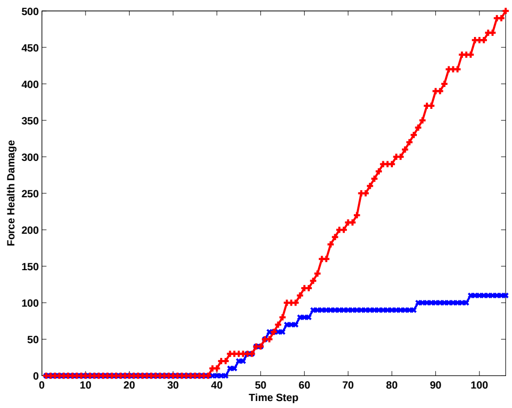


Figure 8.21: Total force damage over time

8.20 and 8.21). During this stage, one may note that the average degree of the blue communication network keeps at a high level and the average shortest path length at a low level (Figure 8.18), while the red damage keeps at a high level (Figures 8.20 and 8.21). It seems the damage of the red force is mainly because of communication in the blue force, which can be confirmed by the reasoning log. In this stage, the reasoning log is full of such statements as "An average damage of 4 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team" at the time step 68, 69, 70, 72, 77, 78, 79, 82, 83, 84, 85, 87, 88, 89, 92, 93, 94, 98 and 103. In this stage, the reasoning log still interprets the simulation through three aspects: what is happening, how it happens, and why it happens.

8.5 Summary

In this chapter, a series of experiments were conducted to investigate the effect of force size, firepower (direct fire), communication, strategic planning and terrain characteristics in urban warfare. A number of findings have been obtained. The urban terrain seems to favour the force with high firepower. The skill of the combatants is more important than the force size. Based on LSL, this should not happen between two evenly matched forces. It is because the blocks in the urban terrain largely reduce the ability of a large force to coordinate its fire while the engagement occurring within a narrow corridor allow the small size force to fully take advantages of its high firepower. However, sometimes the numerically superior force still shows its advantages, especially with communication. In summary the overall impact of the block in the urban terrain is a combination of two effects. One is that the block may fragment the force and reduce the degree of coordination of the force. The other is that the block may protect the agents from being shot.

With communication, the agents may have more information about their own force and enemy. For a non-dispersed force, the information about its own force does not have much effect on the outcome because the agents are neutral to their own force. However, the more information about the enemy, the more agents move to and fire at their enemy. Therefore in general the longer the communication range, the better. For a dispersed force, the more information about its own force, the more dispersed the force, and the harder to coordinate agents' behaviours. The overall effect of communication for a dispersed force is a combination of both effects of information about own force and enemy. Therefore a long communication range is not always good for a dispersed force. Moreover, the effect of communication is also influenced by the terrain type. The block not only makes the force scattered, but also prevents the agents from firing at their enemy. With the direct weapon, the effective communication may not increase when the communication range increases above a certain level. This is the reason why no improvement occurs in some terrains when the communication range increases.

With strategic planning, the force may largely improve its performance in most cases. However, in some cases the performance is worse than without a strategic planning mechanism. It suggests that different terrain types or force configurations require different strategic planning mechanisms. Since the agents try to move far from its own force members for a dispersed force, this dispersing effect largely offsets the clustering effect of a strategic planning. Therefore, the improvement made by a non-dispersed force is larger than that by a dispersed force. Communication and strategic planning are two means of coordination. Therefore, if a high degree of coordination has been achieved by using either communication or strategic planning, it is hard to make further improvement by adopting another one.

The patterns captured in the scenarios without coordination are consistent with those by traditional ABD (Davies et al. 2004). The large force almost cannot beat the small force. And the higher the density of the urban terrain, the worst the performance of the large force. However, if highly coordinating agents' behaviours, the large force can beat the small force in many cases. And a phase transition can be observed where highly dense urban terrains may lead to similar outcomes as open terrains, while medium to light dense urban terrains have different dynamics. Therefore the results obtained from the ABDs without rational strategic planning or coordination among agents can be misleading and might not be generalized.

At the end of this chapter, a single simulation was analysed to demonstrate how to understand the simulation through the reasoning log in WISDOM-II. Normally there are three stages in combat: pre-engagement stage, engagement stage and epilogue of combat. In the first stage, the reasoning log mainly shows what is going on in the battlefield and how the forces prepare for the battle. For the other two stages, the reasoning log interprets the simulation through three aspects: what is happening, how it happens, and why it happens.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

Simulation has been used to study combat for a very long time with both human-based and computer-based systems. In this thesis, a comprehensive literature review is first undertaken in models and simulations of combat. Although human-based simulation is more realistic, it is extremely expensive and does not allow defence analysts to investigate all aspects of combat. Most traditional computer-based simulations, such as such as ELAN, JANUS, CASTFORME, ModSAF and OneSAF, are built on equation based models, especially the Landchester Equations, which can be adequate for studying the role of weapon systems in combat. However, human factors and other aspects have not been included in the models. Therefore it is hard to study them by traditional computer-based models and simulations. Recently, CAS and MAS have widely been accepted as two valuable tools in military analysis. The idea that combat can be modelled as a CAS has widely been accepted and adopted in the field.

By its nature, MAS is a promising tool to study CAS. Current agent architecture includes reactive agent architecture, cognitive agent architecture and hybrid agent architecture. All of them have some advantages while facing some disadvantages.

Mainly based on reactive agent architecture, several ABDs have been developed and used to simulate comabt gain insight into it, such as ISAAC, EINSTein, MANA, CROCADILE and BactoWars. ISAAC and EINSTein are the first to ABDs which modelled combat as a CAS and used the theory of CAS to analyse combat. All latter ABDs were inspired by them. These systems are low resolution abstract models. However, the low fidelity in these models does not necessarily reduce their value when capturing and analysing the dynamics of combat. Any model is an abstraction of the real system to some degree. Adding more details to a model does not necessarily improve its ability to reproduce reality because details may complicate the model unnecessarily. Understanding the physics of each individual component does not always improve our understanding of the system as a whole. The key issue is that a good model should include and appropriately represent the factors which influence the behaviour of the real system. Validity of a model is not based on whether the model captures the detailed "mechanics" of each entity, instead it is based on whether the model can reproduce the behaviour (i.e. the statistical patterns) observed in the real system. Based on the research done by Ilachinski (1997, 1999, 2000), Barlow and Easton (2002) concluded that there is no doubt that existing agent-based simulation combat models can exhibit similar behavioural patterns that we would intuitively expect in a real battlefield.

Researchers (Nicholls and Tagarev 1994; Tailby et al. 2001; Perla and Loughran 2003; Bowley et al. 2003) also show that such low fidelity ABS/ABD may not only help people to better understand warfare, but also to guide the implementation of warfare models with higher fidelity. The applicability of ABS/ABD combat models is discussed in detail by Ilachinski (Ilachinski 2004). These ABS/ABDs have already been applied in a number of areas (i.e. education, policy analysis) by a number of organizations (i.e. universities, defence departments, consulting companies) (Lauren and Stephen 2002a; Galligan and Lauren 2003; Ilachinski 2004).

Using a similar architecture, WISDOM-I was developed and used as a simulation platform to establish an understanding of combat as a CAS. The main improvements of WISDOM-I are as follows:

1. Using agent based communication instead of squad based communication in existing ABDs, which increases the complexity of the system and makes the system more realistic;

2. Using a relational database (MySql) to store information. This facilitates post-analysis;

3. Improving the movement algorithm to avoid strange behaviours arising from lack of near/far discrimination;

4. Embedding an EC engine (single objective or multiple objectives) to search for optimal capability of a force for certain predefined scenarios.

Most of these ABDs, including WISDOM-I, are mainly built on reactive agent architecture. Its shortcomings limits the ability of these ABDs to answer such questions as: "which interaction plays a key role in combat?", "how does one relationship between agents affect others?", "how are these interaction evolving during simulation?", etc. Therefore, a novel agent architecture called NCMAA is proposed, which is based on network theory and CAS. NCMAA models each type of interaction as a network and agents as the nodes of one or more networks. Through such an explicit model of interactions within the system, the role of interactions can easily be identified and analysed. The emerging behaviours can then be interpreted through a powerful real-time reasoning engine, which is built on network theory, causal models and various analyses, such as Granger causality test, path analysis and root cause analysis. It helps users to understand the dynamics and outcomes of the simulation by conducting inductive reasoning during the simulation. Such reasoning at the group (network) level not only creates a new way to gain insight into a CAS, but also overcomes some problems, e.g. low scalability, arising from reasoning at the individual level in some agent architecture, e.g. BDI. The major advantages of NCMAA are concluded as follows:

- It is easy for users to analyse interactions between agents through a networked based model of interaction.
- It provides a chance to identify how one interaction influences another through a real-time network centric reasoning engine;
- It establishes for the first time a formal framework for reasoning at the group level in ABDs, which allows analysts to understand the results during the simulation.

This contribution provides supportive evidence for **hypothesis 2**.

Based on NCMAA, WISDOM-II is proposed and implemented in this thesis. With a real-time networked based reasoning engine, WISDOM-II is the first ABD be able to interpret the emerging behaviours online and present the interpretation in natural language to the users. Major unique features of WISDOM-II are summarized as follows:

- Built-in network analysis tool to conduct structural reasoning.
- Complex C3 (command, control and communication) model: WISDOM-II supports up to four level C2 hierarchy - commander, team leader, swarm leader and combatant. It is the first ABD which supports heterogeneous agents at the squad level.
- Integration of tactics with strategies: WISDOM-II supports decision making at two different levels: tactical and strategic. Almost all ABDs for combat have only tactical decision making mechanisms. This situation not only may lead to misunderstanding the process or dynamics of combat, but also may limit such ABDs to study combat at a high level. This contribution provides supportive evidence for **hypothesis 1**.
- Model of recovery: An explicit model of artificial hospital is first introduced in WISDOM-II. Each team may have a hospital defined by the number of

doctors and the recovery rate. A waiting list is used to queue the wounded combatant for treatment when all doctors are already treating. With such a model of recovery, defence analysts can easily and quickly study the dynamics of combat under limited resources and the relationship between the outcome of combat and the capability of recovery.

Because of global urbanization, the focus of military operations is shifting from open terrain to urban terrain. In this thesis, simulations are conducted on scenarios of MOUT. The results show the features of urban terrain largely influence the outcomes of combat. Recalling from chapter 8, the urban terrain seems to favour the small force with high firepower over the large force with low firepower. The skill of combatants is more important than the force size. Communication may compensate the disadvantages of the large force with low firepower to a certain degree. The force performance can be largely improved by adopting strategic planning. However, in some cases the performance is worse than without a strategic planning mechanism. It suggests that different terrain types or force configurations require different strategic planning mechanisms.

EC techniques have seldom been used in military analysis although they are very popular in optimization. In this thesis they are first adopted to conduct fitness landscape analysis, to characterise the solution space of combat simulations and to identify the degree of difficulty in searching for optimal solutions. The landscapes from both WISDOM-I and WISDOM-II are rugged and highly multimodal. The characteristics of the solution space largely depend on the strategy used by the red team. Multi-objective optimization seems a very promising tool in military analysis since the objective of each force in combat usually requires success in several conflicting aspects.

9.2 Future work

Several research directions for further investigation arise from this thesis. To facilitate the discussion, they are discussed in three categories:

9.2.1 Model of interaction

NCMAA takes the first step to explicitly model the interaction among the components within a CAS. With the aid of network theory, the interaction can be characterized with a number of network measures. The effect of one interaction on another is qualitatively analysed through time series analysis, correlation analysis, path analysis, etc.

How to quantify the influence of one interaction on another is still an open research question. This would involve identifying relationships between network measures (such as average degree and average path length) with behaviour. Some preliminary work has been undertaking using the Penetration Coefficient (PC), which combines four measures concerning the level of overlap between networks.

9.2.2 ABD for combat

Several ABDs for combat have been developed to simulate and study combat since Project Albert launched in 1991. Although these ABDs have been widely used and have let defence analysts study and understand combat, they still need to be enhanced in following aspects:

- **Movement**. Most existing ABDs adopt an attraction/replusion weighting system to make decisions for agents on the battlefield about which position they should move to. In the study of the movement algorithms used in ISAAC, EINSTein and MANA version 2, Gill and Grieger (Gill and Grieger 2003) argued that such a movement algorithm may lead to unexpected and weird behaviours under certain extreme situations. Based on their recommendations, later ABDs or versions improved their movement algorithms. However it is still hard to match them to reality. In reality, the combatants usually assess the risks for current position and all potential positions and then balance their own preference, the goal (order from commander) and the risk. Finally they make decisions. A better way is required to measure the risk instead of using personality based attraction/replusion weighting system, and a decision mechanism is also required that combines the risk, goal and own preference. WISDOM-II implements part of this (strategic decision making mechanism). But further extension should be made, e.g. applying risk assessment into tactical decision making.

- **Recovery**. Although WISDOM-II already has a model for recovery, it is very simple and not flexible enough for defence analyst to fully investigate the role of recovery in combat. Extensions could include modelling recovery on site and triage of different conditions.
- **Logistics**. No ABD has a model for logistics, a very important factor in combat.
- **Learning**. There is no learning mechanism for them to learn from their experience for most of current ABDs in combat.
- **Coordination**. At present communication is the only way to coordinate among agents in most existing ABDs. How about other types? For example, how to form an advantageous shape based on the shape of its opponent? How to adjust moving speed based on the speed of whole group in order to maintain certain shape of whole group? All these need to be modelled in the future.

9.2.3 EC techniques

In this thesis, EC techniques are used to study the characteristics of the solution space. Other EC techniques could also be employed in military analysis. In combat, usually two forces play against each other. Either force may adjust its strategy or tactics based on its opponent's tactics and strategy. Therefore co-evolution can be adopted to study the dynamics of combat when both forces are changing during the simulation. Within the same framework, cooperative co-evolution could be employed to study coordination among agents within the same force.

Agents currently make decisions based on their personalities. It may be possible to use neural network to make decisions for agents. Then the learning process can be simulated through evolving the parameters of the neural network.

Bibliography

Abbass, H. A. and R. Sarker (2002). The pareto differential evolution algorithm. *International Journal on Artificial Intelligence Tools 11*(4), 531–552.

Ackoff, R. L. (1979a). The future of operational research is past. *Journal of the Operational Research Society 30*(2), 93–104.

Ackoff, R. L. (1979b). Resurrecting the future of operational research. *Journal of the Operational Research Society 30*(3), 189–199.

Albert, R. and A. Barabási (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics 74*(1), 47–97.

Alberts, D. S. and J. J. Garstka (2001). Network centric warfare: Department of defense report to congress. Technical report, Department of Defense, U.S.A.

Alberts, D. S., J. J. Garstka, and F. P. Stein (1999). *Network Centric Warfare: Developing And Leveraging Information Superiority (2nd edition)*. CCRP Publication Series.

Aragon, A. R. (2001). Agent-based simulation of a marine infantry squad in an urban environment. Master thesis, Naval Postgradutate School, Monterey, California.

Axelrod, R. (1984). *The Evolution of Cooperation*. New York: Basic Books.

Baas, N. A. (1994). Emergence, hierarchies, and hyperstructures. In C. G. Langton (Ed.), *Alife III, Santa Fe Studies in the Sciences of Complexity, Proc. Volume XVII*, pp. 515–537. Redwood City: Addison-Wesley.

Balci, O. (1994). Validation, verification, and testing techniques throughout the life cycle of a simulation study. In J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila (Eds.), *Proceedings of the 1994 Winter Simulation Conference*, Lake Buena Vista, FL, USA, pp. 215–220.

Bar-Yam, Y. (2003). *Dynamics of Complex Systems*. Westview Press.

Bar-Yam, Y. (2004). A mathematical theory of strong emergence using multiscale variety. *Complexity 9*(6), 15–24.

Barlow, M. (2003). JBOTS, CROCADILE & TDSS: 3 conflict driven multi-agent systems for education, experimentation, & decision support. In H. A. Abbass and J. Wiles (Eds.), *The Australian Conference on Artificial Life ACAL 2003*, pp. 1–12.

Barlow, M. and A. Easton (2002). CROCADILE - an open, extensible agent-based distillation engine. *Information & Security 8*(1), 17–51.

Barringer, H., M. Fisher, D. Gabbay, G. Gough, and R. Owens (1989). METATEM: A framework for programming in temporal logic. In *REX Workshop on Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness (LNCS Volume 430)*, Berlin, Germany, pp. 94 – 129. Springer-Verlag.

Bates, D. and G. A. McIntyre (2004). Joint warfare system (JWARS) overview. http://www.mors.org/meetings/oa_nco/nco_pres/Bates_McIntyre3.pdf. Access on February 9, 2006.

Battilega, J. A. and J. K. Grauge (1984). *The Military Applications of Modeling*. Wright-Patterson AFB, OH: Air Force Institute of Technology Press.

Beckerman, L. P. (1999). The non-linear dynamics of war. http://www.belisarius.com/modern_business_strategy/beckerman/non_linear.htm. Access on February 14, 2006.

Bedau, M. A. (1997). Weak emergence. *Nos 31*, 375–399.

Bedau, M. A. (2002). Downward causation and autonomy in weak emergence. *Principia 6*, 5–50.

Bennington, R. W. (1995). Joint simulation system (JSIMS) - an overview. In *Proceedings of the IEEE 1995 National Aerospace and Electronics Conference (NAECON 1995)*, Volume 2, Dayton, OH, USA, pp. 804–809. IEEE.

Beyerchen, A. (1992). Clausewitz, nonlinearity, and the unpredictability of war. *International Security 17*(3), 59–90.

Bonasso, R. P., R. J. Firby, E. Gat, D. Kortenkamp, D. P. Miller, and M. G. Slack (1997). Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental & Theoretical Artificial Intelligence 9*(2), 237–256.

Borshchev, A., Y. Karpov, and V. Kharitonov (2002). Distributed simulation of hybrid systems with AnyLogic and HLA. *Future Generation Computer Systems 18*(6), 829–839.

Boswell, S., N. Curtis, P. Dortmans, and N. Tri (2003). A parametric study of factors leading to success in conflict: Potential warfighting concepts. In *Proceedings of the Land Warfare Conference*, Adelaide, Australia, pp. 369–381.

Bowley, D., T. Castles, and A. Ryan (2003). Constructing a SUITE of analytical tools: A case study of military experimentation. *ASOR BULLETIN 22*(4), 2–10.

Boyer, M. M. (2001). Root cause analysis in perinatal care: Health care professionals creating safer health care systems. *Journal of Perinatal & Neonatal Nursing 15*(1), 40–54.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation 2*(1), 14–23.

Brooks, R. A. (1990). Elephants don't play chess. *Robotics and Autonomous Systems 6*(1&2), 3–15.

Brooks, R. A. (1991a). Intelligence without reason. In M. J. and R. R. (Eds.), *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pp. 569–595. Sydney, Australia: Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

Brooks, R. A. (1991b). Intelligence without representation. *Artifical Intelligence 47*(1-3), 139–159.

Brown, L. P. (2000). Agent based simulation as an exploratory tool in the study of the human dimension of combat. Master thesis, Naval Postgraduate School, Monterey, California.

Brown, M. L., L. Jackson, D. Mueller, and J. Allison (2003). Military operations in urban terrain. In G. Horne and S. Johnson (Eds.), *Maneuver Warfare Science 2003*, pp. 39–46. USMC Project Albert.

Bryson, J. (2000). Cross-paradigm analysis of autonomous agent architecture. *Journal of Experimental and Theoretical Artificial Intelligence 12*(2), 165–189.

Burroughs, T. E., J. C. Cira, P. Chartock, A. R. Davies, and W. C. Dunagan (2000). Using root cause analysis to address patient satisfaction and other improvement opportunities. *Joint Commission Journal on Quality and Patient Safety 26*(8), 439–449.

CACI (2006a). Joint modeling and simulation system (JMASS). http://www.caci.com/business/systems/simulation/jmass.shtml. Access on February 9, 2006.

CACI (2006b). Joint warfare system (JWARS). http://www.caci.com/business/systems/simulation/jwars.shtml. Access on February 9, 2006.

CACI (2006c). JQUAD+/joint simulation system (JSIMS). http://www.caci.com/business/systems/simulation/jsims.shtml. Access on February 9, 2006.

Caldwell, W. J. and R. Wood (1995). Junk - janus fast movers. In C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman (Eds.), *Proceedings of the 1995 Winter Simulation Conference*, New York, NY, USA, pp. 1237–1243. ACM Press.

Campbell, A. P. and J. R. Hummel (1998). The dynamic information architecture system: An advanced simulation framework for military and civilian applications. http://www.dis.anl.gov/DIAS/papers/SCS/SCS.html. Access on Feburary 10, 2006.

Carlson, J. S. and R. Söderberg (2003). Assembly root cause analysis: A way to reduce dimensional variation in assembled products. *International Journal of Flexible Manufacturing Systems 15*(2), 113–150.

Castelfranchi, C. (2001). The theory of social functions: Challenges for computational social science and multi-agent learning. *Journal of Cognitive Systems Research 2*(1), 5–38.

Caughlin, D. (2000). An integrated approach to verification, validation, and accredition of models and simulation. In J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick (Eds.), *Proceedings of the 2000 Winter Simulation Conference*, Orlando, FL, USA, pp. 872–881.

Ceranowicz, A. (1994). Modular semi-automated forces. In J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila (Eds.), *Proceedings of the 1994 Winter Simulation Conference*, Orlando, Florida, United States,, pp. 755–761. ACM Press.

Chaitin, G. J. (1987). *Information, Randomness and Incompleteness : Papers on Algorithmic Information Theory*. Singapore: World Scientific.

Chapman, D. (1987). Planning for conjunctive goals. *Artificial Intelligence 32*(3), 333378.

Chen, P. S. and P. Chu (2001). Applying lanchester's linear law to model the ardennes campaign. *Naval Research Logistics 48*(8), 653–661.

Cioppa, T. M. (2002). *Efficient Nearly Orthogonal and Space-Filling Experimental for High-Dimensional Complex Models*. Ph.d. thesis, Naval Postgraduate School.

Coello, C., D. A. V. Veldhuizen, and G. B. Lamont (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers.

Collier, N., T. Howe, and M. North (2003). Onward and upward: The transition to repast 2.0. In *Proceedings of the First Annual North American Association for Computational Social and Organizational Science Conference*, Pittsburgh, PA USA.

Cooper, C. (1993). Complexity in C3I systems. In D. G. Green and T. Bossomaier (Eds.), *Complex Systems: From Biology to Computation*, pp. 223–231. Amsterdam: IOS Press.

Coram, R. (2002). *Boyd: The Fighter Pilot Who Changed the Art of War*. U.S.A.: Little, Brown.

Coylea, G. and D. Exelby (2000). The validation of commercial system dynamics models. *System Dynamics Review 16*(1), 27–41.

Crino, S. T. (2001). Representation of urban operations in military models and simulations. In B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer (Eds.), *Proceedings of the 2001 Winter Simulation Conference*, pp. 715–720.

Curtis, N. J. and P. J. Dortmans (2004). A dynamic conceptual model to explore technology-based perturbations to a complex system: the land force. *Asia-Pacific Journal of Operational Research 21*(4), 463–485.

Curtis, N. J., P. J. Dortmans, and J. Ciuk (2005). doing the right problem versus doing the problem right: Problem structuring within a land force environment. *Journal of the Operational Research Society 1*(13). advance online publication,.

Dastani, M. M. and L. Torre (2002). A classification of cognitive agents. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society (Cogsci'02)*, Washington DC, USA, pp. 256–261.

Davies, P. J., A. Stephens, and R. B. Connell (2004). Building urban combat theory: An investigation of the benefits of force size and firepower in simple urban environments. Technique Report DSTO-TR-1658, Land Operations Division, Defence Science and Technology Organisation, Australia. Limited Release.

De Wolf, T. and T. Holvoet (2005). Emergence versus self-organisation: different concepts but promising when combined. In *Engineering Self Organising Systems: Methodologies and Applications*, Volume 3464 of *Lecture Notes in Computer Science (LNCS)*, pp. 1–15. Springer Verlag.

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Kanpur, India: John Wiley & Sons, Inc.

Dekker, A. H. (2005a). Network topology and military performance. In A. Zerger and R. M. Argent (Eds.), *MODSIM 2005 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand*, Melborune, Australia, pp. 2174–2180.

Dekker, A. H. (2005b). Simulating network robustness for critical infrastructure networks. In V. Estivill-Castro (Ed.), *Twenty-Eighth Australasian Computer Science Conference (ACSC2005)*, Volume 38 of *CRPIT*, Newcastle, Australia, pp. 59–68. ACS.

Dekker, A. H. (2005c). A taxonomy of network centric warfare architectures. In *SETE 2005 Systems Engineering / Test and Evaluation Conference*, Brisbane, Australia.

Desch, M. C. (2001). *Soldiers in Cities: Military Operations on Urban Terrain* (Softcover ed.). Strategic Studies Institute, U.S. Army War College.

DIAS (1995). The dynamic information architecture system: A high level architecture for modeling and simulation. White paper, Advanced Computer Applications Center, Argonne National Laboratory, Argonne, IL. Access on Feburary 10, 2006.

d′Inverno, M., D. Kinny, M. Luck, and M. J. Wooldridge (1997). A formal specification of dmars. In M. P. Singh, A. S. Rao, and M. J. Wooldridge (Eds.), *Intelligent Agents IV: Proceedings of the Fourth International Workshop on Agent Theories, Architectures, and Languages, LNAI 1365*, pp. 155–176. Springer-Verlag.

DOD (2003). Defense science board task force on the role and status of DoD red teaming activities. Technical Report Unclassified Report 20301-3140, Office of the Under Secretary of Defense For Acquisition, Technology, and Logistics Washington, D.C.

DOD (2005). Department of defense dictionary of military and associated terms. http://www.dtic.mil/doctrine/jel/doddict/. Access on Sept. 6, 2005.

DOE (1992). Root cause analysis guidance document. DOE Guideline DOE-NE-STD-1004-92, Office of Nuclear Energy, U.S. Department of Energy, Washington, D.C.

Dolansky, L. (1964). Present state of the lanchester theory of combat. *Operations Research 12*(2), 344–358.

Dorogovtsev, S. N. and J. F. F. Mendes (2002). Evolution of networks. *Advances in Physics 51*(4), 1079–1187.

Dortmans, P. J., N. J. Curtis, and N. Tri (2005). An analytical approach for constructing and measuring concepts. *Journal of the Operational Research Society 1*(7). Advance online publication.

Dunne, P., E. Nikolaidou, and V. Vougas (2001). Defence spending and economic growth: A causal analysis for Greece and Turkey. *Defence and Peace Economics 12*, 5–26.

Dunne, P. and V. Vougas (1999). Military spending and economic growth in South Africa: A causal analysis. *Journal of Conflict Resolution 43*(4), 521–537.

Emmeche, C., S. Kppe, and F. Stjernfelt (1997). Explaining emergence: Towards an ontology of levels. *Journal Journal for General Philosophy of Science 28*(1), 83–117.

Epstein, J. M. (2002). Modeling civil violence: An agent-based computational approach. *PNAS 99*(90003), 7243–7250.

Esteva, M., J. A. Padget, and C. Sierra (2002). Formalizing a language for institutions and norms. In *ATAL '01: Revised Papers from the 8th International Workshop on Intelligent Agents VIII*, Volume 2333 of LNAI, London, UK, pp.

348–366. Springer-Verlag.

Ferber, J. (1999). *Multiagent Systems: An Introduction to Distributed Artificial Intelligence*. Edinburgh: Addison Wesley.

Ferber, J. and O. Gutknecht (1998). A meta-model for the analysis and design of organizations inmulti-agent systems. In *Proceedings of International Conference on Multi Agent Systems*, pp. 128–135.

Fernandes, C. M. B., R. Walker, A. Price, J. Marsden, and L. Haley (15). Root cause analysis of laboratory delays to an emergency department. *Journal of Emergency Medicine 5*(1997), 735–739.

Fields, M. (2002). Representing ground robotic systems in battlefield simulations. In E. R. Messina and A. M. Meystel (Eds.), *Proceedings of the 2002 PerMIS Workshop*, Gaithersburg, MD. NIST: NIST. NIST special publication 990.

Fisher, M. (1994). A survey of concurrent metatem - the language and its applications. In D. M. Gabbay and H. J. Ohlbach (Eds.), *Temporal Logic Proceedings of the First International Conference (LNAI 827)*, Berlin, Germany, pp. 480–505. Springer-Verlag.

Flake, G. W. (1998). *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation*. Cambridge, MA: The MIT Press.

Fricker, R. D. (1998). Attrition models of the ardennes campaign. *Naval Research Logistics 45*(1), 1–22.

Galligan, D. and M. Lauren (2003). Operational aspects of imaging radar system in maritime reconnaissance aircraft. *Journal of Battlefield Technology 6*(3), 29–32.

Galligan, D. P. (2004). Modelling shared siturational awareness using the MANA model. *Journal of Battlefield Technology 7*(3), 35–40.

Galligan, D. P., M. A. Anderson, and M. K. Lauren (2003). *MANA: Map Aware Non-uniform Automata Version 3.0 Users Manual*. Devonport, Aukland: Defence Technology Agency. Draft version.

Georgeff, M. P. and A. L. Lansky (1987). Reactive reasoning and planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, Seattle, WA, pp. 677682.

Gilbert, N. and S. Bankes (2002). Platforms and methods for agent-based modeling. *Proceedings of the National Academy of Sciences of the USA 99*(suppl. 3), 7197–7198.

Gill, A. W. (2004). Improvement to the movement algorithm in the mana agent-based distillation. *Journal of Battlefield Technology 7*(2), 19–22.

Gill, A. W., R. R. Egudo, P. J. Dortmans, and D. Grieger (2002). Using agent based distillations in support of the army capability development process - a case study. Technical report DSTO-TR-1378, DSTO, Australia.

Gill, A. W. and D. Grieger (2003). Validation of agent based distillation movement algorithms. Technical Report DSTO-TN-0476, Defence Science and technology organisation, Austrailia.

Gillett, C. (2002a). Strong emergence as a defense of non-reductive physicalism: A physicalist metaphysics for 'downward' determination. *Principia 6*(1), 89–120.

Gillett, C. (2002b). The varieties of emergence: Their purposes, obligations and importance. *Grazer Philosophische Studien 65*, 89–115.

Goldberg, D. E. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms.* Norwell, MA, USA: Kluwer Academic Publishers.

Goldstein, J. (1999). Emergence as a construct: History and issues. *Emergence 1*(1), 49–72.

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica 37*, 424–438.

Grau, L. W. and J. W. Kipp (1999). Confronting the specter. *Military Review 79*(4), 9–17.

Grieger, D. (2002). Using agent based distillations to support land warfare studies. General document DSTO-GD-0340, DSTO Systems Sciences Laboratory, Edinburgh, SA, Australia.

Grieger, D. and D. A. Gill (2001). Exploring the effect of the fog of war on the value of competitive edges in land warfare. In *the Proceedings of The 10th Biennial Computational Techniques and Applications Conference*.

Griffith, S. B. (1963). *Sun Tzu - The Art of War*. Oxford University Press.

Hamilton, J. D. (1994). *Time Series Analysis*. Princeton, NJ: Princeton University Press.

Handley, C. C. (2000). Quality improvement through root cause analysis. *Hospital Materiel Management Quarterly 21*(1), 74–8.

Harris, R. J. (2001). *A Primer of Multivariate Statistics* (3rd edition ed.). Mahwah, NJ: Lawrence Erlbaum Assoc Inc.

Hartley, D. S. (1995). A mathematical model of attrition data. In J. Bracken, M. Kress, and R. E. Rosenthal (Eds.), *Warfare Modeling*, pp. 443–465. John Wiley & Sons.

Hiemstra, C. and J. D. Jones (1994). Testing for linear and nonlinear granger causality in the stock price- volume relation. *The Journal of Finance 49*(5), 1639–1664.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA: The MIT Press.

Holland, J. H. (1996). *Hidden Order: How Adaptation Builds Complexity (Helix Books)*. New York: Addison Wesley Publishing Company.

Holland, J. H. (1998). *Emergence: From Chaos to Order*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Hordijk, W. (1996). A measure of landscapes. *Evolutionary Computation 4*(4), 335–360.

Horn, J. and D. E. Goldberg (1999). Genetic algorithm difficulty and the modality of fitness landscapes. In L. D. Whitley and M. D. Vose (Eds.), *Foundations of Genetic Algorithms 3*, pp. 243–269. San Francisco, CA: Morgan Kaufmann.

Huhns, M. N. and M. P. Singh (1998). Cognitive agents. *IEEE Internet Computing 2*(6), 87–89.

Ilachinski, A. (1997). Irreducible semi-autonomous adaptive combat (ISAAC): An artificial life approach to land combat. Research Memorandum CRM 97-61, Center for Naval Analyses, Alexandria.

Ilachinski, A. (1999). Enhanced isaac neural simulation toolkit (EINSTein), an artificial-life laboratory for exploring self-organized emergence in land combat. Beta-Test User's Guide CIM 610.10, Center for Naval Analyses.

Ilachinski, A. (2000). Irreducible semi-autonomous adaptive combat (ISAAC): An artificial life approach to land combat. *Military Operations Research 5*(3), 29–46.

Ilachinski, A. (2003). Exploring self-organized emergence in an agent-based synthetic warfare lab. *The International Journal Of Systems And Cybernetics 32*(1-2), 38–76.

Ilachinski, A. (2004). *Artificial War: Multiagent-Based Simulation of Combat.* Singapore: World Scientific Publishing Company.

Inchiosa, M. E. and M. T. Parker (2002). Overcoming design and development challenges in agentbased modeling using ASCAPE. In *Proceedings National Academy of Sciences*, Volume 99(3), pp. 73047308.

JACK (2005). *JACK Agent Manual* (Release 5.0 ed.). Carlton South, Victoria, Australia: Agent Oriented Software Pty. Ltd.

Johnson, S. (2001). *Emergence: The Connected Lives of Ants, Brains, Cities, and Software.* New York: Scribner.

Jones, T. and S. Forrest (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In L. Eshelman (Ed.), *Proceedings* *of the Sixth International Conference on Genetic Algorithms*, San Francisco, CA, pp. 184–192. Morgan Kaufmann.

Julisch, K. (2003). Clustering intrusion detection alarms to support root cause analysis. *ACM Transactions on Information and System Security 6*(4), 443–471.

Kaelbling, L. P. (1991). A situated-automata approach to the design of embedded agents. *SIGART Bulletin 2*(4), 85–88.

Kallel, L., B. Naudts, and C. R. Reeves (2001). Properties of fitness functions and search landscapes. In L. Kallel, B. Naudts, and A. Rogers (Eds.), *Theoretical Aspects of Evolutionary Computing*, pp. 175– 206. Berlin: Springer-Verlag.

Kollias, C., G. Manolas, and S. Paleologou (2004). Defence expenditure and economic growth in the European union: A causality analysis. *Journal of Policy Modeling 26*(5), 553569.

Kota, R. (2002). Survey report on JACK: A framework tool for building multi-agent software systems for CMIF projects. Report, Center for MultiSource Information Fusion, School of Engineering & Sciences, University at Buffalo, Buffalo, NY.

Lagnado, D. A., M. R. Waldmann, Y. Hagmayer, and S. A. Sloman (2006). Beyond covariation: Cues to causal structure. In A. Gopnik and L. Schulz (Eds.), *Causal learning: Psychology, Philosophy, and Computation.* Oxford: Oxford University Press. in press.

Lanchester, F. W. (1916). *Aircraft in Warfare, the Dawn of the Fourth Arm.* London: Constable.

Larkin, B. (1997). MASS, real-time, and Hardware-In-The-Loop. In *Proceedings of the IEEE 1997 National Aerospace and Electronics Conference (NAECON 1997)*, Volume 2, Dayton, OH, USA, pp. 525–529. IEEE.

Lauren, M. K. (2000). Modelling combat using fractals and the statistics of scaling systems. *Military Operations research 5*(3), 47–58.

Lauren, M. K. (2001a). Describing rates of interaction between multiple autonomous entities: An example using combat modelling. http://arxiv.org/ftp/nlin/papers/0109/0109024.pdf. Access on April 28, 2006.

Lauren, M. K. (2001b). Exporing the value of sensors to a recce unit using agent-based models. *Journal Of Battlefield Technology 4*(1), 44–49.

Lauren, M. K. (2001c). Fractal methods applied to describe cellular automaton combat models. *Fractals 9*(2), 177–184.

Lauren, M. K. (2002). Firepower concentration in cellular automaton combat modelsan alternative to lanchester. *Journal of the Operational Research Society 53*(6), 672679.

Lauren, M. K. and R. T. Stephen (2000). Modelling patrol survivability in a generic peacekeeping setting using ISAAC. Report DOTSE Report 177, Defence Operational Technology Support Establishment, New Zealand, Auckland.

Lauren, M. K. and R. T. Stephen (2002a). Fractals and combat modelling: Using MANA to explore the role of entropy in complexity science. *Fractals 10*(4), 481–490.

Lauren, M. K. and R. T. Stephen (2002b). MANA: Map-aware non-uniform automata. a new zealand approach to scenario modelling. *Journal of Battlefield Technology 5*(1), 27–31.

Lepingwell, J. W. R. (1987). The laws of combat? lanchester reexamined. *International Security 12*(1), 89–134.

Lesperance, Y., H. J. Levesque, F. Lin, D. Marcu, R. Reiter, and R. B. Scherl (1996). Foundations of a logical approach to agent programming. In M. Wooldridge, J. P. Muller, and M. Tambe (Eds.), *Intelligent Agents II (LNAI 1037)*, pp. 331–346. Berlin, Germany: Springer-Verlag.

Leszak, M., D. E. Perry, and D. Stoll (2000). A case study in root cause defect analysis. In *ICSE '00: Proceedings of the 22nd International Conference on Software Engineering*, New York, NY, USA, pp. 428–437. ACM Press.

Levin, S. A. (1998). Ecosystems and the biosphere as complex adaptive systems. *Ecosystems 1*(5), 431 – 436.

Lucas, T. W. and T. Turkes (2003). Fitting lanchester equations to the battles of kursk and ardennes. *Naval Research Logistics 51*(1), 95–116.

Luke, S., C. Cioffi-Revilla, L. Panait, and K. Sullivan (2004). Mason: A new multi-agent simulation toolkit. In *Proceedings of the 2004 SwarmFest Workshop*.

Maes, P. (1991). The agent network architecture (ANA). *SIGART Bulletin 2*(4), 115–120.

Mateski, M. (2004). Toward a red teaming taxonomy, 2.0. *Red Team Journal*.

Maxwell, D. T. (2000). An overview of the joint warfare system (JWARS). Technique report, The MITRE Corporation, McLean, Virginia.

Merz, P. and B. Freisleben (1999). Fitness landscapes and memetic algorithm design. In C. D., D. M., and G. F. (Eds.), *New Ideas in Optimization*, pp. 245–260. London: McGraw-Hill.

Milinski, M. (1987). Tit for tat in sticklebacks and the evolution of cooperation. *nature 325*, 433–435.

Minar, N., R. Burkhart, C. Langton, and M. Askenazi (1996). The swarm simulation system: A toolkit for building multi-agent simulations. Working Paper 96-06-042, Santa Fe Institute, Santa Fe.

Minsky, M. (1986). *The society of mind.* New York, NY, USA: Simon & Schuster, Inc.

Mitchell, M., S. Forrest, and J. H. Holland (1992). The royal road function for genetic algorithms: Fitness landscapes and GA performance. In F. J. Varela and P. Bourgine (Eds.), *Proceedings of the First European Conference on Artificial Life*, Cambridge, MA, pp. 245–254. MIT Press.

MSRR (2005). Combined arms and support task force evaluation model. http://www.msrr.army.mil/index.cfm?RID=MNS_A_1000185. Access on February 8, 2006.

Nelms, C. R. (2003). Root cause analysis - not what you might think. http://www.maintenanceworld.com/root-cause-analys.htm. Access on March 8, 2006.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review 45*(2), 167–256.

Nicholls, D. and T. D. Tagarev (1994). What does chaos theory mean for warfare? *Airpower Journal 8*(3), 48–57.

Nolfi, S. (2002). Power and the limits of reactive agents. *Neurocomputing 42*(1-4), 119–145.

North, M. J., N. T. Collier, and J. R. Vos (2006). Experiences creating three implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation 16*(1), 1–25.

Nowak, M. and K. Sigmund (1992). Tit for tat in heterogeneous populations. *Nature 355*, 250–253.

Nwana, H. S. (1996). Software agents: An overview. *Knowledge Engineering Review 11*(3), 205–244.

Odell, J. J. (2002). Agents and complex systems. *Journal of Object Technology 1*(2), 35–45.

Omicini, A. (2001). Soda: Societies and infrastructures in the analysis and design of agent-based systems. In P. Ciancarini and M. Wooldridge (Eds.), *Agent-Oriented Software Engineering*, Volume 1957 of LNAI, pp. 185–193. Springer-Verlag.

Parunak, H. and S. Brueckner (2004). Engineering swarming systems. In F. Bergenti, M. P. Gleizes, and F. Zambonelli (Eds.), *Methodologies and Software Engineering for Agent Systems*. Kluwer.

Parunak, H., S. Brueckner, and J. Sauter (2002). ERIMs approach to fine-grained agents. In *Proceedings of NASA/JPL Workshop on Radical Agent Concepts (WRAC02)*.

Parunak, H. V. D. and S. Brueckner (2001). Entropy and self-organization in multi-agent systems. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pp. 124130. ACM Press.

Pearl, J. (2000). *Causality: Models, Reasoning and Inference.* New York: Cambridge University Press.

Perla, P. P. and J. J. Loughran (2003). Using gaming and agent technology to explore C2. In *Proceedings of the 8th International Command and Control Research and Technology Symposium*, Washington DC. ThoughtLink, Inc., Vienna, VA.

Perry, N. (2006). Verification and validation of the fractal attrition equation. Technical report DSTO-TR-1822, DSTO, Australia.

Pfleeger, S. L. (1998). *Software Engineering: Theory and Practice.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Phillips, J., M. J. McCloskey, P. L. McDermott, S. L. Wiggins, D. A. Battaglia, M. L. Thordsen, and G. Klein (2001). Decision-centered mout training for small unit leaders. Research Report 1776, U.S. Army Research Institute for the Behavioral and Social Sciences.

Przemieniecki, J. S. (2000). *Mathematical Methods in Defense Analyses* (Third Edition ed.). American Institute of Aeronautics and Astronautics (AIAA).

Ramamohanarao, K., J. Bailey, and P. Busetta (2001). Transaction oriented computational models for multi-agent systems. In *Proceedings of the 13th International Conference on Tools with Artificial Intelligence*, Dallas, Texas, USA, pp. 11–17.

Rao, A. S. (1996). Decision procedures for propositional linear-time belief-desire-intention logics. In M. Wooldridge, J. P. Muller, and M. Tambe (Eds.), *Intelligent Agents II (LNAI 1037)*, pp. 33 – 48. Berlin, Germany: Springer-Verlag.

Rao, A. S. and M. P. Georgeff (1991). Modeling rational agents within a BDI-architecture. In R. Fikes and E. Sandewall (Eds.), *Proceedings of Knowledge*

Representation and Reasoning (KR&R-91), San Mateo, CA, pp. 473 – 484. Morgan Kaufmann Publishers.

Rao, A. S. and M. P. Georgeff (1995). BDI agents: From theory to practice. In *Proceedings of the 1st International Conference on Multi-Agent Systems*, San Francisco, USA, pp. 312–319.

Rex, J. H., J. E. Turnbull, S. J. Allen, K. V. Voorde, and K. Luther (2000). Systematic root cause analysis of adverse drug events in a tertiary referral hospital. *Joint Commission Journal on Quality and Patient Safety 26*(10), 563–575.

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics 21*(4), 25–34. access to http://www.red3d.com/cwr/papers/1987/SIGGRAPH87.pdf on 12 January 2006.

Rooney, J. J. and L. V. Heuvel (2004). Root cause analysis for beginners. *Quality Progress*, 45–53. http://www.asq.org/pub/qualityprogress/past/0704/index.html.

Sandoz, J. F. (2001). Red teaming: A means to military transformation. IDA Paper P-3580, Institute for Defense Analyses.

Sargent, R. G. (1999). Validation and verification of simulation models. In P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans (Eds.), *Proceedings of the 1999 Winter Simulation Conference*, Phoenix, AZ, USA, pp. 39–48.

Savall, M., M. Itmi, and J.-P. Peuchet (2001). YAMAN - a new organization model for multi-agent systems and its platform named phoenix. In *2001 Summer Computer Simulation Conference*.

Sawyers, W. A. (1998). Marine corps analytic modeling and simulation. In D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan (Eds.), *Proceedings of the 1998 Winter Simulation Conference*, Washington, DC, pp. 855–858.

Scherrer, J. H. (2003). Risks and vulnerabilities of network-centric forces: Insights from the science of complexity. Newport r.i., Naval War College.

Schmitt, J. F. (1997). Complexity, global politics, and national security. In D. S. Alberts and T. J. Czerwinski (Eds.), *Command and (Out of) Control: The Military Implications of Complexity Theory*, Chapter 9, pp. 99–111. Washington, DC: National Defense University.

Serenko, A. and B. Detlor (2002). Agent toolkits: A general overview of the market and an assessment of instructor satisfaction with utilizing toolkits in the classroom. Working Paper 455, McMaster University, Hamilton, Ontario, Canada.

Shalizi, C. R. (2004). Methods and techniques of complex systems science: An overview. In T. S. Deisboeck, J. Y. Kresh, and T. B. Kepler (Eds.), *Complex Systems Science in Biomedicine*. New York: Kluwer Academic.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal 27*(3), 379–423.

Sheldon, B. (2002). Comparing the results of a nonlinear agent-based model to lanchester's linear model. In G. Horne and S. Johnson (Eds.), *Maneuver Warfare Science 2002*. Quantico, VA, USA: USMC Project Albert.

Siekkinen, M., G. Urvoy-Keller, E. W. Biersack, and T. En-Najjary (2005). Root cause analysis for long-lived TCP connections. In *CoNEXT '05: Proceedings of the 2005 ACM Conference on Emerging Network Experiment and Technology*, New York, NY, USA, pp. 200–210. ACM Press.

Simlote, A. (2003). Joint warfare system (JWARS). In *the Proceeding of 2003 User Group Conference (DoD UGC'03)*, pp. 325–329. IEEE.

Smith, T., P. Husbands, P. Layzell, and M. O'Shea (2002). Fitness landscapes and evolvability. *Evolutionary Computation 10*(1), 1–34.

Smith, T., P. Husbands, and M. O'Shea (2001). Not measuring evolvability: Initial investigation of an evolutionary robotics search space. In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, COEX, World

Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, pp. 9–16. IEEE Press.

Stadler, P. F. (1995). Towards a theory of landscapes. In R. Lopéz-Peña, R. Capovilla, R. García-Pelayo, H. Waelbroeck, and F. Zertuche (Eds.), *Complex Systems and Binary Networks*, Volume 461, Berlin, New York, pp. 77–163. Springer-Verlag.

Stadler, P. F. (1996). Landscapes and their correlation functions. *Journal of Mathematical Chemistry 20*, 1–45.

Stadler, P. F. (2002). Fitness landscapes. In M. Lässig and A. Valleriani (Eds.), *Biological Evolution and Statistical Physics*, Berlin, pp. 187–207. Springer-Verlag.

Stone, G. F. and G. A. McIntyre (2001). The joint warfare system (JWARS): A modeling and analysis tool for the defense department. In B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer (Eds.), *Proceedings of the 2001 Winter Simulation Conference*, Volume 1, Arlington, VA, USA, pp. 691–696. IEEE.

Strogatz, S. (2003). *Sync: The Emerging Science of Spontaneous Order*. New York: Hyperion Press.

Sycara, K. P. (1998). Multiagent systems. *AI Magazine 19*(2), 79–92.

Tailby, D., D. Grieger, A. Gill, C. Stanford, and R. Cause (2001). Operational synthesis for reconnaissance and surveillance. In *Proceedings of the Land Warfare Conference 2001*, Sydney.

Teo, J. and H. A. Abbass (2003). Search space difficulty of evolutionary neuro-controlled legged robots. *The International Journal of Knowledge based Intelligent Engineering Systems 7*(3), 149–156.

Teo, J. and H. A. Abbass (2004a). Automatic generation of controllers for embodied legged organisms: a pareto evolutionary multi-objective approach. *Evolutionary Computation 12*(3), 355–394.

Teo, J. and H. A. Abbass (2004b). An information-theoretic landscape analysis of neuro-controlled embodied organisms. *Neural Computing and Applications Journal 13*(1), 80–89.

Thornhill, N. F., S. L. Shah, and B. Huang (2001). Detection of distributed oscillations and root-cause diagnosis. In *Proceeding of CHEMFAS 4*, Jejudo(Chejudo) Island, Korea, pp. 167–172.

Tisue, S. and U. Wilensky (2004). Netlogo: A simple environment for modeling complexity. In *Proceedings of the International Conference on Complex Systems*, Boston.

Tobias, R. and C. Hofmann (2004). Evaluation of free Java-libraries for social-scientific agent based simulation. *Journal of Artificial Societies and Social Simulation 7*(1).

Tuli, R. W. and G. E. Apostolakis (1996). Incorporating organizational issues into root-cause analysis. *Process Safety and Environmental Protection 74*(B1), 3–16.

Vassilev, V. K., T. C. Fogarty, and J. F. Miller (2000). Information characteristics and the structure of landscapes. *Evolutionary Computation 8*(1), 31–60.

Vassilev, V. K. and J. F. Miller (2000). The advantages of landscape neutrality in digital circuit evolution. In *Proceedings of the 3rd International Conference on Evolvable Systems: From Biology to Hardware*, Berlin, pp. 252–263. Springer-Verlag.

Vázquez-Salceda, J., V. Dignum, and F. Dignum (2005). Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems 11*(2), 307–360.

Waldmann, M. R. (1996). Knowledge-based causal induction. In D. R. Shanks, K. J. Holyoak, and D. L. Medin (Eds.), *The Psychology of Learning and Motivation*, Volume 34, Chapter Causal learning, pp. 47–88. Diego, CA: Academic Press.

Waldmann, M. R. (2000). Competition among causes but not effects in predictive and diagnostic learning. *Journal of Experimental Psychology: Learning,* *Memory, and Cognition 26*(1), 5376.

Waldmann, M. R. (2001). Predictive versus diagnostic causal learning: Evidence from an overshadowing paradigm. *Psychological Bulletin and Review 8*(3), 600608.

Waldrop, M. (1992). *Complexity: The Emerging Science at the Edge of Order and Chaos*. New York: Simon and Schuster.

Walton, G. H., R. M. Patton, and D. J. Parsons (2001). Usage testing of military simulation systems. In B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer (Eds.), *Proceedings of the 2001 Winter Simulation Conference*, Arlington, VA, USA, pp. 771–779.

Wasserman, S. and K. Faust (1994). *Social Network Analysis: Methods and Applications.* Cambridge, UK: Cambridge University Press.

Weinberger, E. (1990). Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics 63*, 325–336.

Wheeler, S. (2005). A theoretic analysis of combined arms teaming. Technical report DSTO-TR-1748, DSTO, Australia.

White, G. (2004). The mathematical agent - a complex adaptive system representation in bactowars. In *First Workshop on Complex Adaptive Systems for Defence*, Adelaide, Australia.

Wilensky, U. and W. Stroup (1999). Learning through participatory simulations: Network-based design for systems learning in classrooms. In *Proceedings of the Computer Supported Collaborative Learning Conference*, Stanford University, pp. 667–676.

Wilson, C. (2004). Network centric warfare: Background and oversight issues for congress. Report for congress, CRS.

Wittman, R. L. and C. T. Harrison (2001). Onesaf: A product line approach to simulation development. In *Proceedings of the European Simulation Interoperability Workshop*, Harrow, Middlesex, UK.

Wooldridge, M. J. (1999). 1: Intelligent agents. In G. Weiss (Ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pp. 3–44. The MIT Press.

Wooldridge, M. J. and N. R. Jennings (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review 10*(2), 115–152.

Wooldridge, M. J., N. R. Jennings, and D. Kinny (2000). The gaia methodology for agent- oriented analysis and design. *Autonomous Agents and Multi-Agent Systems 3*(3), 285–312.

Wright, S. (1932). The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In D. Jones (Ed.), *Proceedings of the Sixth International Congress on Genetics*, Volume 1, Brooklyn, NY, pp. 356–366.

Xu, J. and H. Chen (2005a). Crimenet explorer: A framework for criminal network knowledge discovery. *ACM Transactions on Information Systems (TOIS) 23*(2), 201–226.

Xu, J. and H. Chen (2005b). Criminal network analysis and visualization. *Communications of the ACM (CACM) 48*(6), 100–107.

Yang, A., H. A. Abbass, and R. Sarker (2005a). A comparison study of WISDOM and MANA in a networked environment. Technical Report TR-ALAR-200512001, University of New South Wales, ADFA, Canberra, Australia.

Yang, A., H. A. Abbass, and R. Sarker (2005b). WISDOM-II: A network centric model for warfare. In *Ninth International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES 2005), LNCS 3683*, Melbourne, Australia.

Zhang, K., A. Yen, X. L. Zhao, D. Massey, S. F. Wu, and L. X. Zhang (2004). On detection of anomalous routing dynamics in BGP. In N. Mitrou, K. Kontovasilis, G. N. Rouskas, I. Iliadis, and L. Merakos (Eds.), *Proceedings of Third International IFIP-TC6 Networking Conference, LNCS3042*, Athens, Greece, pp. 259–270.

Zitzler, E. (1999). *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications.* Ph.d. thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Appendix A

Results of orthogonal simulations

Table A.1: NLER of the non-dispersed force with short range vision and firing (30) in the baseline scenario

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	1.04	2.03	3.56	4.65	5.62	3.38
Gap8	0.99	1.85	2.74	3.17	3.28	2.41
Gap4	0.99	1.32	1.47	1.75	1.90	1.49
Gap3	1.05	1.47	1.78	2.13	2.22	1.73
Gap2	0.99	1.54	2.15	2.68	2.49	1.97
Gap1	0.99	1.71	2.31	2.78	2.81	2.12
Average	1.01	1.65	2.34	2.86	3.05	–

Table A.2: NLER of the dispersed force with short range vision and firing (30) in the baseline scenario

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	1.01	1.46	2.16	3.04	3.59	2.25
Gap8	0.98	1.51	2.05	2.37	2.77	1.94
Gap4	0.97	1.61	2.06	2.53	2.91	2.02
Gap3	1.03	1.42	2.03	2.13	2.76	1.87
Gap2	0.98	1.47	1.84	2.45	2.86	1.92
Gap1	1.01	1.37	2.09	2.47	2.63	1.91
Average	1.00	1.47	2.04	2.50	2.92	–

Table A.3: NLER of the non-dispersed force with long range vision and firing (99) in the baseline scenario

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	1.02	0.95	1.00	0.96	0.92	0.97
Gap8	0.94	1.21	1.42	1.50	1.44	1.30
Gap4	0.95	1.32	1.45	1.79	1.90	1.48
Gap3	0.97	1.48	1.82	2.19	2.35	1.76
Gap2	0.95	1.56	2.14	2.58	2.47	1.94
Gap1	1.04	1.66	2.33	2.74	3.09	2.17
Average	0.98	1.36	1.69	1.96	2.03	–

Table A.4: NLER of the dispersed force with long range vision and firing (99) in the baseline scenario

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	1.02	0.95	1.00	0.96	0.92	0.97
Gap8	1.00	1.30	1.56	1.85	1.93	1.53
Gap4	0.98	1.35	1.70	2.00	2.33	1.67
Gap3	1.03	1.35	1.63	1.89	2.24	1.63
Gap2	0.97	1.24	1.55	1.89	2.32	1.59
Gap1	1.13	1.28	1.50	1.96	2.19	1.61
Average	1.02	1.25	1.49	1.76	1.99	–

Table A.5: NLER of the non-dispersed force with short range vision and firing (30) in the extended scenario A

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.22	0.38	0.58	0.76	1.07	0.60
Gap8	0.68	1.08	1.46	1.85	2.05	1.42
Gap4	0.81	1.16	1.46	1.64	1.67	1.35
Gap3	0.84	1.28	1.51	1.56	1.91	1.42
Gap2	0.73	1.21	1.45	1.63	1.81	1.37
Gap1	0.75	1.21	1.43	2.08	1.89	1.47
Average	0.67	1.05	1.31	1.59	1.73	–

Table A.6: NLER of the dispersed force with short range vision and firing (30) in the extended scenario A

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.95	1.57	2.14	2.85	3.59	2.22
Gap8	0.89	1.45	2.00	2.40	2.67	1.88
Gap4	0.77	1.24	1.75	2.24	2.58	1.72
Gap3	0.77	1.33	1.83	2.31	2.39	1.73
Gap2	0.74	1.33	1.87	2.24	2.70	1.78
Gap1	0.67	1.26	1.77	2.26	2.33	1.66
Average	0.80	1.36	1.89	2.38	2.71	–

Table A.7: NLER of the non-dispersed force with long range vision and firing (99) in the extended scenario A

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	1.06	1.01	0.97	0.95	0.90	0.98
Gap8	0.78	1.05	1.15	1.32	1.33	1.13
Gap4	0.84	1.22	1.40	1.62	1.73	1.36
Gap3	0.81	1.33	1.54	1.74	1.94	1.47
Gap2	0.75	1.12	1.40	1.73	1.64	1.33
Gap1	0.88	1.17	1.56	1.86	1.88	1.47
Average	0.85	1.15	1.34	1.54	1.57	–

Table A.8: NLER of the dispersed force with long range vision and firing (99) in the extended scenario A

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	1.06	1.01	0.97	0.95	0.90	0.98
Gap8	1.01	1.31	1.60	1.78	2.00	1.54
Gap4	0.74	1.24	1.66	1.60	2.04	1.45
Gap3	0.74	1.37	1.61	1.64	2.09	1.49
Gap2	0.69	1.30	1.88	2.12	2.34	1.67
Gap1	0.64	1.13	1.56	1.97	1.88	1.44
Average	0.81	1.23	1.55	1.68	1.87	–

Table A.9: NLER of the non-dispersed force without communication in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	1.02	2.17	3.18	4.22	5.24	3.17
Gap8	0.98	1.75	2.79	3.40	3.27	2.44
Gap4	0.99	1.33	1.52	1.83	1.92	1.52
Gap3	0.98	1.50	1.93	2.16	2.29	1.78
Gap2	1.06	1.57	2.04	2.77	2.60	2.01
Gap1	1.00	1.65	2.41	2.83	2.89	2.15
Average	1.01	1.66	2.31	2.87	3.04	–

Table A.10: NLER of the non-dispersed force with communication range of 1 in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.79	1.22	1.73	2.31	2.52	1.71
Gap8	0.85	1.28	1.79	2.06	2.20	1.64
Gap4	1.02	1.29	1.37	1.64	1.88	1.44
Gap3	1.01	1.44	1.64	2.08	2.50	1.73
Gap2	0.98	1.57	1.86	2.48	2.59	1.90
Gap1	0.98	1.66	2.23	2.97	3.24	2.21
Average	0.94	1.41	1.77	2.26	2.49	–

Table A.11: NLER of the non-dispersed force with communication range of 5 in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.37	0.46	0.48	0.51	0.50	0.46
Gap8	0.60	0.83	1.01	1.18	1.32	0.99
Gap4	1.04	1.26	1.38	1.66	1.81	1.43
Gap3	1.02	1.45	1.70	2.17	2.56	1.78
Gap2	1.05	1.54	1.89	2.53	2.46	1.89
Gap1	0.99	1.69	2.31	2.99	3.18	2.23
Average	0.84	1.21	1.46	1.84	1.97	–

Table A.12: NLER of the non-dispersed force with communication range of 10 in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.19	0.23	0.22	0.21	0.21	0.21
Gap8	0.52	0.71	0.85	0.93	1.01	0.80
Gap4	0.84	1.20	1.56	1.99	2.30	1.58
Gap3	0.76	1.16	1.56	2.11	2.52	1.62
Gap2	0.71	1.22	1.61	2.22	2.59	1.67
Gap1	0.99	1.86	2.51	2.74	3.03	2.23
Average	0.67	1.06	1.38	1.70	1.94	–

Table A.13: NLER of the non-dispersed force with communication range of 15 in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.13	0.15	0.16	0.15	0.15	0.15
Gap8	0.43	0.53	0.65	0.77	0.84	0.64
Gap4	0.80	1.33	1.77	2.11	2.42	1.68
Gap3	0.70	1.27	1.84	2.16	2.41	1.68
Gap2	0.70	1.23	1.77	2.05	1.96	1.54
Gap1	0.98	1.67	2.03	2.39	2.51	1.92
Average	0.62	1.03	1.37	1.61	1.71	–

Table A.14: NLER of the non-dispersed force with communication range of 20 in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.12	0.12	0.12	0.12	0.11	0.12
Gap8	0.38	0.48	0.54	0.64	0.73	0.56
Gap4	0.82	1.35	1.74	2.13	2.25	1.66
Gap3	0.77	1.22	1.85	2.08	2.18	1.62
Gap2	0.73	1.19	1.63	1.92	1.82	1.46
Gap1	0.89	1.61	2.08	2.08	2.35	1.80
Average	0.62	0.99	1.33	1.50	1.57	–

Table A.15: NLER of the dispersed force without communication in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	1.00	1.56	2.12	3.00	3.69	2.28
Gap8	1.05	1.51	1.98	2.36	2.76	1.93
Gap4	0.98	1.58	1.91	2.51	3.19	2.03
Gap3	0.94	1.49	1.90	2.18	2.74	1.85
Gap2	1.01	1.40	1.86	2.30	2.78	1.87
Gap1	1.06	1.52	1.97	2.61	2.76	1.98
Average	1.01	1.51	1.96	2.49	2.99	–

Table A.16: NLER of the dispersed force with communication range of 1 in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.99	1.38	1.92	2.63	3.19	2.02
Gap8	0.98	1.56	1.96	2.30	2.73	1.91
Gap4	1.07	1.72	2.12	2.34	2.84	2.02
Gap3	1.01	1.56	2.09	2.35	2.73	1.95
Gap2	1.03	1.48	2.03	2.45	2.79	1.96
Gap1	0.99	1.30	1.89	2.56	2.65	1.88
Average	1.01	1.50	2.00	2.44	2.82	–

Table A.17: NLER of the dispersed force with communication range of 5 in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.78	0.86	1.06	1.20	1.28	1.03
Gap8	0.91	1.33	1.65	1.84	2.07	1.56
Gap4	0.99	1.66	2.00	2.35	2.59	1.92
Gap3	1.11	1.56	1.94	2.23	2.84	1.94
Gap2	1.05	1.59	2.01	2.74	3.05	2.09
Gap1	1.15	1.53	2.17	2.68	2.99	2.10
Average	1.00	1.42	1.81	2.17	2.47	–

Table A.18: NLER of the dispersed force with communication range of 10 in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.52	0.52	0.51	0.56	0.57	0.54
Gap8	0.76	1.09	1.33	1.46	1.63	1.25
Gap4	1.03	1.71	2.15	2.49	2.59	1.99
Gap3	1.09	1.73	2.27	2.43	2.58	2.02
Gap2	1.32	2.00	2.21	2.61	3.03	2.23
Gap1	1.23	1.87	2.26	2.52	2.76	2.13
Average	0.99	1.49	1.79	2.01	2.19	–

Table A.19: NLER of the dispersed force with communication range of 15 in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.38	0.34	0.35	0.36	0.38	0.36
Gap8	0.69	0.94	1.10	1.29	1.43	1.09
Gap4	0.86	1.28	1.56	1.76	2.03	1.50
Gap3	0.86	1.35	1.64	1.71	1.82	1.48
Gap2	1.00	1.37	1.46	1.62	1.78	1.45
Gap1	1.05	1.43	1.53	1.72	1.84	1.51
Average	0.81	1.12	1.27	1.41	1.55	–

Table A.20: NLER of the dispersed force with communication range of 20 in the extended scenario B

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.27	0.26	0.26	0.27	0.28	0.27
Gap8	0.67	0.88	0.97	1.16	1.20	0.98
Gap4	0.76	1.04	1.43	1.60	1.82	1.33
Gap3	0.84	1.07	1.36	1.46	1.53	1.25
Gap2	0.79	1.14	1.33	1.46	1.43	1.23
Gap1	0.84	1.08	1.23	1.47	1.49	1.22
Average	0.69	0.91	1.10	1.24	1.29	–

Table A.21: NLER of the non-dispersed force without communication in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.25	0.39	0.61	0.72	1.03	0.60
Gap8	0.67	1.07	1.49	1.83	1.91	1.40
Gap4	0.87	1.16	1.36	1.50	1.75	1.33
Gap3	0.83	1.18	1.43	1.72	1.77	1.39
Gap2	0.71	1.19	1.42	1.61	1.73	1.33
Gap1	0.86	1.18	1.51	1.75	1.79	1.42
Average	0.70	1.03	1.30	1.52	1.66	–

Table A.22: NLER of the non-dispersed force with communication range of 1 in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.18	0.27	0.38	0.45	0.55	0.36
Gap8	0.58	0.91	1.20	1.42	1.54	1.13
Gap4	0.83	1.16	1.35	1.66	1.76	1.35
Gap3	0.78	1.18	1.43	1.71	1.90	1.40
Gap2	0.67	1.00	1.43	1.57	1.53	1.24
Gap1	0.85	1.01	1.51	1.90	1.88	1.43
Average	0.65	0.92	1.22	1.45	1.53	–

Table A.23: NLER of the non-dispersed force with communication range of 5 in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.14	0.17	0.17	0.20	0.22	0.18
Gap8	0.49	0.69	0.92	1.05	1.17	0.86
Gap4	0.90	1.19	1.40	1.54	1.81	1.37
Gap3	0.83	1.18	1.45	1.71	1.88	1.41
Gap2	0.66	1.08	1.40	1.49	1.69	1.26
Gap1	0.84	1.19	1.61	1.96	2.20	1.56
Average	0.64	0.92	1.16	1.32	1.50	–

Table A.24: NLER of the non-dispersed force with communication range of 10 in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.12	0.11	0.12	0.13	0.13	0.12
Gap8	0.49	0.65	0.87	0.97	1.07	0.81
Gap4	0.85	1.27	1.43	1.62	1.79	1.39
Gap3	0.80	1.20	1.57	1.91	2.01	1.50
Gap2	0.66	1.13	1.49	1.82	1.91	1.40
Gap1	0.82	1.40	1.81	2.28	2.55	1.77
Average	0.62	0.96	1.21	1.46	1.58	–

Table A.25: NLER of the non-dispersed force with communication range of 15 in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.10	0.10	0.10	0.10	0.10	0.10
Gap8	0.48	0.60	0.77	0.90	1.00	0.75
Gap4	0.85	1.28	1.54	1.76	1.88	1.46
Gap3	0.77	1.18	1.57	1.84	1.93	1.46
Gap2	0.64	1.01	1.25	1.62	1.72	1.25
Gap1	0.79	1.21	1.48	1.85	2.18	1.50
Average	0.60	0.90	1.12	1.35	1.47	–

Table A.26: NLER of the non-dispersed force with communication range of 20 in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.10	0.09	0.09	0.09	0.09	0.09
Gap8	0.47	0.64	0.75	0.86	0.98	0.74
Gap4	0.79	1.30	1.66	1.75	1.92	1.48
Gap3	0.73	1.36	1.61	1.91	2.07	1.54
Gap2	0.64	1.06	1.39	1.81	1.95	1.37
Gap1	0.80	1.35	1.67	2.16	2.41	1.68
Average	0.59	0.97	1.19	1.43	1.57	–

Table A.27: NLER of the dispersed force without communication in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.97	1.51	2.11	2.73	3.47	2.16
Gap8	0.87	1.42	1.94	2.30	2.69	1.84
Gap4	0.83	1.29	1.80	2.23	2.51	1.73
Gap3	0.76	1.21	1.76	2.05	2.55	1.67
Gap2	0.75	1.25	1.87	2.13	2.60	1.72
Gap1	0.73	1.29	1.68	2.24	2.30	1.65
Average	0.82	1.33	1.86	2.28	2.69	–

Table A.28: NLER of the dispersed force with communication range of 1 in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	1.00	1.36	1.83	2.58	3.15	1.99
Gap8	0.94	1.47	1.98	2.24	2.63	1.85
Gap4	0.84	1.31	1.79	2.24	2.65	1.77
Gap3	0.71	1.31	1.72	2.13	2.57	1.69
Gap2	0.72	1.22	1.71	2.31	2.69	1.73
Gap1	0.70	1.19	1.72	2.18	2.44	1.64
Average	0.82	1.31	1.79	2.28	2.69	–

Table A.29: NLER of the dispersed force with communication range of 5 in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.68	0.84	1.04	1.18	1.31	1.01
Gap8	0.81	1.18	1.61	1.79	2.07	1.49
Gap4	0.79	1.33	1.69	2.19	2.53	1.70
Gap3	0.75	1.29	1.77	2.19	2.69	1.74
Gap2	0.79	1.58	1.82	2.45	2.76	1.88
Gap1	0.80	1.56	2.04	2.56	2.81	1.96
Average	0.77	1.30	1.66	2.06	2.36	–

Table A.30: NLER of the dispersed force with communication range of 10 in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.47	0.50	0.52	0.54	0.57	0.52
Gap8	0.75	0.98	1.23	1.43	1.65	1.21
Gap4	0.75	1.34	1.79	2.13	2.46	1.69
Gap3	0.74	1.37	1.78	2.15	2.32	1.67
Gap2	0.84	1.39	1.81	2.14	2.44	1.72
Gap1	0.89	1.26	1.79	2.18	2.37	1.70
Average	0.74	1.14	1.49	1.76	1.97	–

Table A.31: NLER of the dispersed force with communication range of 15 in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.37	0.35	0.34	0.35	0.38	0.36
Gap8	0.64	0.92	1.10	1.27	1.39	1.07
Gap4	0.77	1.21	1.59	1.84	2.03	1.49
Gap3	0.70	1.27	1.57	1.76	1.80	1.42
Gap2	0.66	1.17	1.33	1.54	1.76	1.29
Gap1	0.74	1.07	1.43	1.58	1.72	1.31
Average	0.65	1.00	1.23	1.39	1.52	–

Table A.32: NLER of the dispersed force with communication range of 20 in the extended scenario C

	50 vs 50	100 vs 50	150 vs 50	200 vs 50	250 vs 50	Average
Open	0.26	0.26	0.26	0.27	0.27	0.26
Gap8	0.61	0.87	0.97	1.09	1.20	0.95
Gap4	0.77	1.15	1.38	1.58	1.71	1.32
Gap3	0.76	1.05	1.37	1.41	1.52	1.22
Gap2	0.71	0.86	1.27	1.39	1.57	1.16
Gap1	0.77	1.13	1.29	1.40	1.52	1.22
Average	0.65	0.89	1.09	1.19	1.30	–

Appendix B

Reasoning Log

- 1: **The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.**
- 2: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 3: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 4: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 5: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 6: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 7: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 8: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 9: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 10: **An order has been sent to the leader of group 1 in the blue team to move toward (30, 70).**
- 10: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.

- 11: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 12: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 13: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 14: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 15: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 16: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 17: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 18: The group 2 in the red team is advancing to the flag.
- 19: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 20: An order has been sent to the leader of group 1 in the blue team to move toward (30, 50).
- 20: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 21: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 22: The group 1 in the blue team is advancing to the flag.
- 23: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 24: The group 1 in the blue team is advancing to the flag.
- 25: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 26: The group 1 in the blue team is advancing to the flag.
- 27: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.

- 28: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 29: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 30: An order has been sent to the leader of group 1 in the blue team to move toward (50, 70).
- 30: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 31: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 32: The group 1 in the blue team is advancing to the flag.
- 33: The group 2 in the red team is advancing to the flag. The group 1 in the blue team is advancing to the flag.
- 34: The group 1 in the blue team is advancing to the flag.
- 35: The group 1 in the blue team is advancing to the flag.
- 36: The group 2 in the red team is advancing to the flag.
- 37: The group 2 in the red team is advancing to the flag.
- 38: **The engagement starts.**
- 38: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 38: **The agents in the blue team are spreading their fire to achieve maximum damage in the red team.**
- 38: **Without any damage, the blue team achieves damage of 10 to the red force.**
- 39: The situation awareness of blue team is gained mainly through its vision component. The situation awareness of red team is gained mainly through its vision component.
- 40: An order has been sent to the leader of group 1 in the blue team to move toward (50, 70).
- 40: The situation awareness of blue team is gained mainly through its vision component. The situation awareness of red team is gained mainly through its vision component.

- 40: **The agents in the blue team are coordinating their firing to achieve maximum damage in the red team.**
- 40: Without any damage, the blue team causes damage of 10 to the red force.
- 41: The situation awareness of blue team is gained mainly through its vision component. The situation awareness of red team is gained mainly through its vision component.
- 42: The situation awareness of blue team is gained mainly through its vision component. The situation awareness of red team is gained mainly through its vision component.
- 42: **The agents in the red team are maximizing their fire range to achieve maximum damage in the blue team. The agents in the blue team are spreading their fire to achieve maximum damage in the red team.**
- 43: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 44: **The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.**
- 44: The agents in the red team are maximizing their fire range to achieve maximum damage in the blue team.
- 44: Without any damage, the red team causes damage of 10 to the blue force.
- 44: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 44: An average damage of 4.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team.
- 45: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 45: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 45: An average damage of 2.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team.

- 46: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 46: The agents in the red team are spreading their fire to achieve maximum damage in the blue team.
- 46: Without any damage, the red team achieves damage of 10 to the blue force.
- 46: An average damage of 6.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 46: An average damage of 2.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team.
- 47: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 47: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 48: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 48: The agents in the red team are maximizing their fire range to achieve maximum damage in the blue team.The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 48: **An average damage of 6.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.**
- 48: **An average damage of 2.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of friend on agent level, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.**
- 49: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.

- 49: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 49: An average damage of 2.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of friend on agent level, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 50: An order has been sent to the leader of group 1 in the blue team to move toward (50, 50).
- 50: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 50: The agents in the red team are spreading their fire to achieve maximum damage in the blue team.The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 50: An average damage of 6.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 50: An average damage of 4.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 51: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 51: The agents in the red team are spreading their fire to achieve maximum damage in the blue team.
- 51: Without any damage, the red team causes damage of 10 to the blue force.
- 51: An average damage of 6.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 51: An average damage of 4.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 52: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.

- 52: The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 52: Without any damage, the blue team causes damage of 10 to the red force.
- 52: An average damage of 6.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level, the red force's situation awareness of friend on force level.
- 52: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of friend on force level.
- 53: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 53: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 53: Without any damage, the blue team causes damage of 10 to the red force.
- 53: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of friend on force level.
- 53: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 54: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 54: The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 54: Without any damage, the blue team causes damage of 10 to the red force.
- 54: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of friend on force level.
- 55: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.

- 55: **The agents in the red team are spreading their fire to achieve maximum damage in the blue team.The agents in the blue team are coordinating their firing to achieve maximum damage in the red team.**
- 55: **The blue team causes more damage to the red team. The damage ratio is 2.00.**
- 55: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of friend on force level.
- 55: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team.
- 56: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 56: An average damage of 2.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of friend on force level.
- 56: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of enemy on agent level.
- 57: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 57: An average damage of 2.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of friend on force level.
- 57: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.
- 58: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 58: The agents in the red team are maximizing their fire range to achieve maximum damage in the blue team.The agents in the blue team are coordinating their firing to achieve maximum damage in the red team.

- 58: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level, the red force's situation awareness of friend on force level.
- 58: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.
- 59: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 59: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 59: Without any damage, the blue team causes damage of 10 to the red force.
- 59: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level, the red force's situation awareness of friend on force level.
- 59: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of enemy on agent level.
- 60: An order has been sent to the leader of group 1 in the blue team to move toward (50, 70).
- 60: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 60: An average damage of 2.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 60: An average damage of 4.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.
- 61: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 61: The agents in the red team are coordinating their firing to achieve maximum damage in the blue team.The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.

- 61: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 61: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 62: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 62: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 62: Without any damage, the blue team causes damage of 10 to the red force.
- 62: An average damage of 4.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 63: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 63: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 63: Without any damage, the blue team causes damage of 20 to the red force.
- 63: An average damage of 2.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 63: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.
- 64: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 64: An average damage of 2.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 64: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team.

- 65: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 65: The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 65: Without any damage, the blue team achieves damage of 20 to the red force.
- 65: An average damage of 12.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team.
- 66: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 66: The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 66: Without any damage, the blue team achieves damage of 10 to the red force.
- 66: An average damage of 12.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of enemy on agent level.
- 67: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 67: The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 67: Without any damage, the blue team causes damage of 10 to the red force.
- 67: An average damage of 12.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of enemy on agent level.
- 68: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 68: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.

- 69: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 69: The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 69: Without any damage, the blue team causes damage of 10 to the red force.
- 69: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 70: An order has been sent to the leader of group 1 in the blue team to move toward (30, 50).
- 70: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 70: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.
- 71: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 71: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 71: Without any damage, the blue team causes damage of 10 to the red force.
- 71: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 72: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 72: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 72: Without any damage, the blue team achieves damage of 30 to the red force.

- 72: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of friend on force level.
- 73: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 74: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 74: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 74: Without any damage, the blue team achieves damage of 10 to the red force.
- 75: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 75: The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 75: Without any damage, the blue team causes damage of 10 to the red force.
- 75: An average damage of 12.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of enemy on agent level.
- 76: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 76: The agents in the blue team are coordinating their firing to achieve maximum damage in the red team.
- 76: Without any damage, the blue team causes damage of 10 to the red force.
- 76: An average damage of 12.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of enemy on agent level.
- 77: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 77: The agents in the blue team are coordinating their firing to achieve maximum damage in the red team.

- 77: Without any damage, the blue team achieves damage of 10 to the red force.
- 77: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 78: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 78: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team.
- 79: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 79: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.
- 80: An order has been sent to the leader of group 1 in the blue team to move toward (50, 50).
- 80: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 80: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 80: Without any damage, the blue team causes damage of 10 to the red force.
- 81: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 81: An average damage of 4.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of friend on force level.
- 82: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 82: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.

- 82: Without any damage, the blue team causes damage of 10 to the red force.
- 82: An average damage of 4.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 83: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 83: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 83: Without any damage, the blue team causes damage of 10 to the red force.
- 83: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 84: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 84: The agents in the blue team are coordinating their firing to achieve maximum damage in the red team.
- 84: Without any damage, the blue team causes damage of 10 to the red force.
- 84: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 85: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 85: The agents in the red team are spreading their fire to achieve maximum damage in the blue team.The agents in the blue team are coordinating their firing to achieve maximum damage in the red team.
- 85: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.

- 86: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 86: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 86: Without any damage, the blue team achieves damage of 10 to the red force.
- 87: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 87: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 87: Without any damage, the blue team achieves damage of 20 to the red force.
- 87: An average damage of 12.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.
- 88: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 88: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.
- 89: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 89: The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 89: Without any damage, the blue team causes damage of 20 to the red force.
- 89: An average damage of 12.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 90: An order has been sent to the leader of group 1 in the blue team to move toward (30, 50).

- 90: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 90: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of friend on force level.
- 91: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 91: The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 91: Without any damage, the blue team causes damage of 10 to the red force.
- 91: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of friend on force level.
- 92: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 92: The agents in the blue team are coordinating their firing to achieve maximum damage in the red team.
- 92: Without any damage, the blue team achieves damage of 20 to the red force.
- 92: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.
- 93: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 93: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level.
- 94: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 94: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team.

- 95: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 95: The agents in the blue team are spreading their fire to achieve maximum damage in the red team.
- 95: Without any damage, the blue team achieves damage of 20 to the red force.
- 95: An average damage of 10.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of friend on force level.
- 96: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 96: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 97: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 97: The agents in the red team are coordinating their firing to achieve maximum damage in the blue team.
- 97: Without any damage, the red team achieves damage of 10 to the blue force.
- 97: An average damage of 2.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level.
- 98: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 98: Without any damage, the blue team achieves damage of 20 to the red force.
- 98: An average damage of 2.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level, the red force's situation awareness of enemy on force level.
- 98: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of friend on force level.

- 99: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 99: An average damage of 2.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level, the red force's situation awareness of enemy on force level.
- 99: An average damage of 8.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of friend on force level.
- 100: An order has been sent to the leader of group 1 in the blue team to move toward (50, 70).
- 100: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 100: An average damage of 2.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on agent level, the red force's situation awareness of enemy on force level.
- 100: An average damage of 4.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of friend on force level.
- 101: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 101: The agents in the blue team are maximizing their fire range to achieve maximum damage in the red team.
- 101: Without any damage, the blue team causes damage of 10 to the red force.
- 101: An average damage of 2.00 occurred in the blue team over the last 5 timesteps is probably caused by the red force's situation awareness of enemy on force level.
- 101: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of friend on force level.
- 102: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.

- 102: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of friend on force level.
- 103: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 103: Without any damage, the blue team causes damage of 20 to the red force.
- 103: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the activities in the communication network of the blue team, the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.
- 104: The situation awareness of blue team is gained mainly through its vision component.The situation awareness of red team is gained mainly through its vision component.
- 104: An average damage of 6.00 occurred in the red team over the last 5 timesteps is probably caused by the blue force's situation awareness of enemy on agent level, the blue force's situation awareness of friend on force level.