2012

Coevolutionary algorithms for the optimization of strategies for red teaming applications

Tirtha Ranjeet
Edith Cowan University

Edith Cowan University

Copyright Warning

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

Coevolutionary Algorithms for the Optimization of Strategies for Red Teaming Applications

by

Tirtha Ranjeet

Master of Science (Information Technology)

Submitted in fulfilment of the requirements for the Degree of

Doctor of Information Technology

Edith Cowan University

School of Computer and Security Science

(December, 2012)

Declaration

I certify that this thesis does not, to the best of my knowledge and belief:

i. incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education;
ii. contain any material previously published or written by another person except where due reference is made in the text of this thesis; or
iii. contain any defamatory material.

Dedication

I would like to dedicate this thesis to my respected mother Maiya, my wife Monica and our newly born son Safal. I could never have completed this thesis without their enormous support. Their motivation has always proven to be a boon for me in times of difficulty.

Acknowledgements

I would like to express my gratitude to my supervisors Professor Chiou-Peng Lam, Professor Philip Hingston and Dr Martin Masek for their enormous encouragement and motivation throughout the course. I could not imagine this thesis coming into reality without their generous assistance. I will never forget how they helped me to explore the opportunities to extend my professional network and career. Also, I am grateful to my family for supporting me in various ways and for their consideration in allowing me to skip the social and ritual responsibilities.

It is a pleasure to thank Dr Judy Clayden who always stood beside me whenever I needed; she has made available her support in a number of ways. I always appreciated her excellent English writing skills. I would like to thank Dr Greg Maguire and Dr Jo McFarlane for organising excellent writing workshops and their suggestions, from which I benefited greatly. For the statistical analysis, Dr Tapan Rai gave useful guidance in determining the statistical tools for the data analysis. Also, I would like to thank Paul Patak for providing me with the necessary equipment and software for my experiment.

I am also very thankful for the lecturing and tutoring experience I gained by working with Barbara Combes and Dr Mark Brogan. I would also like to thank the graduate research school staff members for providing me an opportunity to work as a SOAR Ambassador, which involves providing peer-to-peer support. Also, I am thankful to Dr Leisa Armstrong for organizing various social gatherings and her kind support throughout this journey. I am indebted to all my colleagues and fellow students for the productive interactions which always encouraged me to improve my work.

List of Publications

Ranjeet, T. R., Hingston, P., Lam, C.-P., & Masek, M. (2011). *Analysis of Key Installation Protection using Computerized Red Teaming*. Paper presented at the 34th Australian Computer Science Conference (ACSC), Perth, Australia. (Includes work described in Chapter 3)

Ranjeet, T. R., Lam, C.-P., Masek, M., & Hingston, P. (2011). *The Effects of Diversity Maintenance on Coevolution for an Intransitive Numbers Problem.* Paper presented at the 24th Australasian Joint conference on Artificial Intelligence, Perth, Australia. (Includes work described in Chapter 5)

Ranjeet, T. R., Hingston, P., Lam, C.-P., & Masek, M. (2012). *Evaluating Coevolution on a Multimodal Problem*. Poster session presented at 21st International Conference on Genetic and Evolutionary Computation Conference (GECCO), Philadelphia, USA. (Includes work described in Chapter 6)

Hingston, P., Ranjeet, T. R., Lam, C.-P., & Masek, M. (2012). *A multimodal problem for competitive coevolution.* Paper presented at the 25th Australasian Joint conference on Artificial Intelligence, Sydney, Australia. (Includes work described in Chapter 6)

Abstract

Red teaming (RT) is a process that assists an organization in finding vulnerabilities in a system whereby the organization itself takes on the role of an "attacker" to test the system. It is used in various domains including military operations. Traditionally, it is a manual process with some obvious weaknesses: it is expensive, time-consuming, and limited from the perspective of humans "*thinking inside the box*". Automated RT is an approach that has the potential to overcome these weaknesses. In this approach both the red team (enemy forces) and blue team (friendly forces) are modelled as intelligent agents in a multi-agent system and the idea is to run many computer simulations, pitting the plan of the red team against the plan of blue team.

This research project investigated techniques that can support automated red teaming by conducting a systematic study involving a genetic algorithm (GA), a basic coevolutionary algorithm and three variants of the coevolutionary algorithm. An initial pilot study involving the GA showed some limitations, as GAs only support the optimization of a single population at a time against a fixed strategy. However, in red teaming it is not sufficient to consider just one, or even a few, opponent‟s strategies as, in reality, each team needs to adjust their strategy to account for different strategies that competing teams may utilize at different points. Coevolutionary algorithms (CEAs) were identified as suitable algorithms which were capable of optimizing two teams simultaneously for red teaming. The subsequent investigation of CEAs examined their performance in addressing the characteristics of red teaming problems, such as intransitivity relationships and multimodality, before employing them to optimize two red teaming scenarios. A number of measures were used to evaluate the performance of CEAs and in terms of multimodality, this study introduced a novel *n-peak problem* and a new performance measure based on the Circular Earth Movers‟ Distance.

Results from the investigations involving an intransitive number problem, multimodal problem and two red teaming scenarios showed that in terms of the performance measures used, there is not a single algorithm that consistently outperforms the others across the four test problems. Applications of CEAs on the red teaming scenarios showed that all four variants produced interesting evolved strategies at the end of the

optimization process, as well as providing evidence of the potential of CEAs in their future application in red teaming.

The developed techniques can potentially be used for red teaming in military operations or analysis for protection of critical infrastructure. The benefits include the modelling of more realistic interactions between the teams, the ability to anticipate and to counteract potentially new types of attacks as well as providing a cost effective solution.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ABD:	Agent-Based Distillation
ABS:	Agent-Based System
ANOVA:	Analysis of Variance
AO:	Area of Operation
CCEA:	Competitive Coevolutionary Algorithm
CEA	Coevolutionary Algorithm
CEACFH:	Coevolutionary Algorithm with Combination of FS and HOF
CEAFS:	Coevolutionary Algorithm with Fitness Sharing
CEAHOF:	Coevolutionary Algorithm with Hall of Fame
CEAN:	Coevolutionary Algorithm (Naïve)
CEMD:	Circular Earth Movers‟ Distance
COCEA:	Cooperative Coevolutionary Algorithm
CROCADILE:	Conceptual Research Oriented Combat Agent Distillation
DOD:	Department of Defence
EA:	Evolutionary Algorithm
EINSTein:	Enhanced ISAAC Neural Simulation Toolkit
EMD	Earth Mover‟s Distance
ES:	Evolution Strategy
FS:	Fitness Sharing
GA:	Genetic Algorithm
GP:	Generalisation Performance
HOF:	Hall of Fame
IDM:	Implicit Diversity Maintenance
ISAAC:	Irreducible Semi Autonomous Adaptive Combat
KIN:	Key Installation
MANA:	Map Aware Non-uniform Automata
MOEA:	Multi-Objective Evolutionary Algorithm
NM:	Nautical Mile
OT:	Optimization Technique
PR:	Peak Ratio
PV:	Patrolling Vessel

RT:	Red Teaming
SME:	Subject Matter Expert
SPSS:	Statistical Package for the Social Sciences
SR:	Success Ratio
WBS:	Work Break-down Structure
WISDOM:	Warfare Intelligent System for Dynamic Optimization of Missions

1 Introduction

1.1 Overview

Red teaming (RT) is a risk assessment activity associated with the evaluation and analysis of plans and strategies of an organization. The main purpose of using RT is to reduce risk and create opportunities (Andrews, 2005; Department of Defence, 2003; Meehan, 2007). There are two categories of RT, namely human-based and computer-based RT. The RT process involves a number of different teams, with colours being used to indicate the team‟s role. Typically, blue, red and green are used to represent a group of defenders, adversaries and neutral agents respectively (Meehan, 2007; Yang, Abbass, & Sarker, 2006). As RT is utilised to detect vulnerabilities of a system and may be used to identify different approaches to eliminate them, it has been successfully used in many organizations, including the military, police and manufacturing companies, for evaluating the performance of strategies within their enterprises (Andrews, 2005; Meehan, 2007; Yang, et al., 2006).

According to Fontenot (2005), during the 19th century, the German military introduced the RT approach, where it was used in the planning of combat scenarios. In 1897, the US army introduced a systematic approach and rules for RT and coined the term “red team” for the first time. Subsequently many countries, including France and the United Kingdom, also used human-based RT to practice their war games. Human-based RT involves two teams; the red team simulating the enemy and the blue team simulating defenders. This can be an extremely expensive exercise, as it involves a large amount of manpower. It is also limited in terms of difficulty in exploring all aspects of a specific scenario. This traditional approach has less manpower intensive alternatives, with different approaches such as sand tables, game theory and mathematical models such as Lanchester equations (Sidran, 2004). These approaches model combat scenarios, such as planning attacks and defences and estimating the probabilities of winning or losing, without deploying real troops. However, they have difficulty addressing some existing problems associated with RT exercises, specifically in addressing the non-linear characteristic associated with combat scenarios (Yang, 2006). Attempts to address this have been made since 1994 with Ceranowiez (1994) first introducing the simulation software, modular semi-automated forces (ModSAF) for practicing war games. With

the increasing popularity of ModSAF, many other combat simulation software such as the extended Lanchester model (ELAN) and the combined arms and support task force evaluation model (CASTFOREM) were developed (Ilachinski, 2000; Yang, 2006). These attempts, however, still do not make the RT process any easier since they were based on top-down approaches (Ilachinski, 2003). In addition, the absence of graphic user interfaces (GUIs) was a barrier to be a user-friendly simulator (Ilachinski, 2003). Some limitations of conventional RT software were addressed by introducing agent-based systems (ABS) and multi-agent systems (MAS). In ABS, the interactions between the autonomous agents themselves and with their environment produce emergent behaviours. The introduction of ABS applications also reduced the user‟s involvement in modelling combat simulation as these autonomous agents produce a global behaviour that is dependent upon the interactions of the local entities (autonomous agents), which is a bottom-up approach.

Agent-based distillation (ABD) is a type of ABS where the level of detail associated with the modelling of individual agents is very low. The ABD concept was first implemented in a simulation application called irreducible semi-autonomous adaptive combat (ISAAC) (Ilachinski, 2000). Since then, a number of ABDs have been introduced, including the enhanced ISAAC neural simulation toolkit (EINSTein) (Ilachinski, 2003), map aware non-uniform automata (MANA) (Lauren, 1999), conceptual research oriented combat agent distillation (CROCADILE) (Barlow & Easton, 2002) and warfare intelligent system for dynamic optimization of missions (WISDOM) (Yang, 2006).

1.2 Statement of the Problem

“If you know your enemies and know yourself, you will not be imperilled in a hundred battles.”

The above quote was made by Sun Tzu (Giles, 2005) and highlights the importance of RT in military organizations. The effectiveness of RT exercises in modern warfare has been proven by the introduction of various war games including „kriegspiel“ and „the American kriegspiel“ (Fontenot, 2005). However, RT is not an easy process. Manual RT involves massive human participation which makes it expensive and time-

consuming. The entire process relies upon the specific individual as a planner directing every step in executing the combat scenario. The effectiveness of such practice games is limited from the perspective of humans "*thinking inside the box*". Thus, there is a high possibility of a whole range of differences as to what may occur in the predicted versus the actual sequence of events.

The introduction of computerized simulators such as EINSTein and MANA enhances the RT process (Ilanchiski, 2003). However, it is very hard to explore all possible outcomes of highly non-linear military operations using simulators. Therefore, researchers including Upton and McDonald (2003) and Choo, Chua and Tay (2007) introduced search optimization methods that automate the vulnerability discovery process using a combination of search algorithms and agent-based simulations. The purpose of these optimization methods was to identify vulnerabilities and also to detect optimal strategies that best addressed specific scenarios. However, most of these existing optimization models use a variety of artificial intelligence techniques such as single objective genetic algorithms (GAs). Using these optimization models, the analyst has to run the simulation many times to find vulnerabilities in their planning. In addition, most of them find an optimal solution in response to a fixed opponent‟s strategy, whereas opponents may use more than one attack or defence strategy in reality. This limitation may be a barrier to existing optimization methods in identifying realistic strategies. In a realistic scenario, both sides can change their strategies in response to the opponent‟s strategy.

In order to address the limitations of the existing optimization models, this thesis aims to investigate suitable search methods that can be effectively used in RT and other similar applications. The purpose of this study is described in the following section.

1.3 Purpose of the Study

The aim of this research is to carry out a systematic study incorporating evolutionary algorithms (EAs) for finding good solution sets for RT scenarios and other similar applications. For this, a commonly used technique as suggested by Upton and McDonald (2003), Hingston (2011) and Choo et al. (2007) is used, in which simulations

of scenarios are combined with search algorithms for finding good solutions. In this thesis, MANA is used as a military RT simulator and Genetic Algorithms (GAs) and Coevolutionary Algorithms (CEAs), are the search algorithms. CEAs have many advantages over the GAs. In particular, they can be used even when there is no objective measure to determine the individual‟s fitness. In addition, CEAs can also evolve multiple populations simultaneously. These two properties of CEAs are particularly relevant in RT applications where two teams are evolved concurrently and objective fitness functions are not easily defined. Despite the many advantages associated with CEAs, there are some limitations that compromise the effectiveness of their performance and these are commonly known as the CEA pathologies (Ficici, 2004; Wiegand, 2003). These pathologies can be problematic, as RT applications possess characteristics that are associated with the manifestation of certain CEA pathologies and thus pose a challenge in terms of the ability of CEAs in finding good solutions.

Therefore, the applications of CEAs for optimization of RT scenarios is investigated through a series of test problems designed to isolate various properties of RT scenarios, such as intransitivity and multimodality. A problem domain is intransitive when a simple ranking of solution strength cannot be performed. For example, intransitivity occurs if a solution *A* is better than solution *B*, and *B* is better than solution *C*, yet *C* is better than *A*, as in the example of the rock-paper-scissors game. With regard to RT, a strategy that is considered ineffective in one scenario may turn out to be a winning strategy in another (Sidran, 2004). This indicates that even in RT, a simple ranking of good strategies is not possible, which may demonstrate some elements of intransitivity. Another characteristic is multimodality, where a number of different „good“ solutions exist. In RT applications, there may be more than one effective strategy to defeat an opponent‟s plan. Thus, the assumption is that RT applications demonstrate the properties of multimodality. Eiben and Smith (2003) also suggested that multimodality may occur in most domains. When this issue was investigated in the RT domain as part of a pilot study in this thesis, the study also demonstrated some evidence of multimodality (see Chapter 3). This study investigates suitable techniques that can be incorporated into a basic CEA to address these CEA pathologies in RT and other similar domains.

In summary, the purpose of the study can be listed as below:

- Investigate approaches incorporating EAs, specifically GAs and CEAs, for finding good solution sets for RT scenarios and other similar applications.
- Identify suitable techniques that enhance CEAs. Incorporate the identified variants in CEAs for investigating the issues of intransitivity and multimodality in RT scenarios and other similar domains.
- Investigate suitable measures to evaluate CEAs‟ performance in various problems, including RT.

With the above mentioned purpose, this study makes a number of contributions in relation to the use of CEAs for the optimization of RT and similar domains. The detailed contributions are outlined in the following section.

1.4 Contributions of this study

This thesis makes contributions in the area of optimization for RT and other associated applications using CEAs. In the process of examining this subject, the work also makes contributions of more general applicability to competitive CEAs. The detailed contributions of this research are listed in the following points.

- **Extending knowledge of the factors affecting the performance of coevolutionary algorithms (CEAs) by conducting a systematic study of CEAs with and without common enhancements, in the context of RT and other similar applications.**

  Many researchers including Lauren & Stephen (2002), Upton and McDonald (2003) and Choo, Chua and Tay (2007) have investigated EAs to optimize RT scenarios. One of the limitations of EAs when employed in RT applications is that the population evolves in response to just one or a few opponent strategies. In reality, opponents may have a very large number of unpredictable strategy options. In order to address this issue, researchers such as Choo, Chua, Low and Ong (2009) and Hingston and Preuss (2011) introduced CEAs in developing optimization techniques for RT. The incorporation of CEAs for RT is still in its

infancy and existing studies that used only CEAs which do not address (1) the pathologies associated with CEAs and (2) RT characteristics such as intransitivity and multimodality. This thesis presents a more complete and systematic study of these issues in the following way.

Firstly, this thesis carried out a systematic study of variants of CEAs on a number of artificial test problems with different characteristics: intransitivity and multimodality. Subsequently, two RT scenarios with different objectives were analysed. Secondly, different performance measures were used to evaluate the performance of the variants of CEAs when they are applied to each of a set of four domains (an intransitive number, multimodal and two RT scenarios). For this, the study explored CEAs with or without archives and diversity maintenance techniques to enhance the algorithms‟ performance in finding a good solution set for RT and other similar applications. As an archive, a memory mechanism called Hall of Fame (HOF) (Rosin & Belew, 1997) was used. For the purpose of maintaining diversity, implicit and explicit techniques suggested by Chong, Tino and Yao (2008) were used. For explicit diversity maintenance, mutation rate is varied; whereas a technique called fitness sharing (FS) is used for implicit diversity maintenance. In addition, the combination of diversity maintenance techniques and archives is also evaluated.

This study also presents a multimodal test-problem and various performance measures for CEAs. Generalisation Performance, which was introduced by Chong et al. (2008) for single population CEAs, are implemented in this study for multi-population CEAs. In addition, Circular Earth Movers‟ Distance, Peak Ratio and Success Ratio are incorporated to measure the algorithms‟ performance in dealing with a specific domain characteristic, multimodality.

- **A novel scalable multimodal problem is introduced to test the ability of CEAs to identify multiple global optima.**

  Researchers including Deb and Goldberg (1989), Zitzler, Deb and Thiele (2000) Hansen and Kern (2004) and Singh and Deb (2006) have introduced various multimodal problems. Multimodality refers to the existence of more than one

good solution in the search space. However, the previously introduced test problems are suitable only to evaluate the performances of GAs and multi-objective evolutionary algorithms (MOEAs). Since CEAs are different from other EAs, the existing test problems are not suitable for evaluating CEAs. One main difference relates to the fitness evaluation: GAs and MOEAs are dependent upon objective fitness function(s) whereas in CEAs fitness is calculated on the basis of the interaction between individuals in CEAs. To the best knowledge of this author, multimodal problems that are suitable for evaluating competitive coevolution are not found in the existing literature.

The interest in evaluating the performance of competitive CEAs in addressing multimodality is due to the common belief that multimodality is a characteristic associated with RT applications. As one of the aims of this study involved employing competitive CEAs in RT applications, this research introduces a test multimodal *„n-peak"* problem (Ranjeet, Hingston, Lam, & Masek, 2012) to facilitate a systematic study involving multi-population CEAs. The test problem is designed to be scalable, supporting the capability to be multi-dimensional and can be composed of a variable number of peaks which can be defined by the user. This study employed a one dimensional version of this test problem and the performance of CEAs is judged according to how effectively they identify the peaks.

- **Introduction of the Circular Earth Movers' Distance as a novel metric for measuring the ability of EAs in finding multiple global optima.**
  A technique called Circular Earth Mover"s Distance (CEMD), for comparing two histograms, has been widely used in image processing related research including image retrieval (Rubner, Tomasi, & Guibas, 2000), measurement of texture and colour similarities (Levina & Bickel, 2001), image matching (Ling & Okada, 2006) and image comparison (Rabin, Delon, & Gousseau, 2008). However, no existing work has been found where the CEMD technique has been utilized as a performance measure for any kind of search algorithm.

In order to test the ability of CEAs in detecting multiple peaks, this study employed the *„n-peak"* test problem. For the purpose of measuring the performance of CEAs in detecting multiple optima in the test problem, this thesis introduced the use of CEMD as a performance measure of CEAs. In the *„n-peak"* test problem, the exact locations of the peaks are known. On the basis of these known peaks, an ideal histogram can be created in which all buckets with a peak contain equal number of solutions and the remainder of the buckets are empty. Likewise, an actual histogram can be created for each evolving population which can then be compared with the ideal histogram. A smaller variation between these two histograms implies better performance of the algorithms in finding the known peaks.

- **Introduction of Peak and Success Ratio as novel metrics for measuring the ability of CEAs in detecting multiple peaks.**
  Researchers including Beasley, Bull and Martin (1993) and Thomsen (2004) have measured GAs and differential evolution (DE) algorithms respectively using Peak Ratio and Success Ratio to identify the capabilities of the algorithms in detecting multiple optima. Subsequently, Epitropakis, Plagianakos, Vrahatis (2011) and Otani, Suzuki and Arita (2011) also used the same techniques to measure the capabilities of DEs for multimodal optimization.

  This study introduces the use of the Peak and Success Ratio as performance measures of CEAs for multimodal optimization. The Peak Ratio is the number of peaks identified out of the specified number of defined peaks in the multimodal problem. The Success Ratio is a percentage of how many times all peaks have been successfully identified.

- **Exploring and evaluating the combination of diversity maintenance and archives to enhance the performance of CEAs.**
  Researchers have introduced various techniques to enhance CEAs‟ capabilities to address associated pathologies. Axelrod (1987), Hillis (1990) and Rosin and Belew (1997) have used diversity maintenance techniques to address CEAs‟ cycling pathology. Likewise, Rosin and Belew (1997), Ficici and Pollack (2003)

and Avery, Michalewicz and Schmidt (2008) used an archive as a memory mechanism to address CEAs‟ forgetting pathology.

Diversity maintenance techniques such as fitness sharing disperse the population by encouraging solutions different to existing ones, which helps to locate the best solution in the search space. The memory mechanism approach stores a number of the highest ranked solutions from each generation, which helps to remember the good strategies that occurred in past generations. Researchers discovered that both these techniques are useful on an individual basis in enhancing the performance of CEAs. However, there is limited existing work that integrates these two approaches, and as far as to the knowledge of the author, none that specifically involve red teaming. This thesis investigates and provides insights in terms of the application of the combination of these two techniques with CEAs for the test problems and RT scenarios.

- **Adaptation of Chong et al.'s Generalisation Performance measures to multi-population CEAs.**

  In EAs, the performance of the algorithms is often measured based on fitness values, because individuals are evaluated according to the objective fitness function used. Specifically in RT applications, researchers including Chua et al. (2008) and Xu, Low and Choo (2009) examine only objective fitness as a performance measure of EAs. Two recent studies have involved the use of CEAs in RT applications. The first study was presented by Choo, Chua, Low and Ong (2009) and the second by Hingston and Preuss (2011). In the first study, the performance measure of the CEAs was not disclosed, whereas in the second study, the winning ratio of the team is considered as a performance measure of the algorithm.

  However, objectively evaluating the performance of an individual solution in a CEA is problematic, due to the lack of an objective fitness function especially when the population it competes against, also evolves. Thus, Chong, Tino and Yao (2008) introduced Generalisation Performance (GP), a performance measure for CEAs. The authors have used GP to measure the capability of CEAs in solving

a problem called Iterated Prisoner‟s dilemma (IPD). The IPD is a single population game, in which players receive “reward” or “punishment” on the basis of the cooperation and defection they make with the other players.

Unlike Chong et al. (2008; Chong, Tino, & Yao, 2009)‟s work, this study extends the definition of GP (details are given in section 3.3.1), making it suitable to use as a performance measure for multi-population CEAs. This performance measure is used as a generic tool for measuring performance of CEAs (with or without variants) in finding good solution sets in the various test problems incorporated in this thesis.

- **Examination of the relationship between genotypic and phenotypic diversity, and the relationship between diversity and quality in the context of RT.**
  With respect to the use of CEAs in RT applications, researchers including Choo, Chua, Low and Ong (2009) and Hingston and Preuss (2011) have presented analyses of the fitness value and measure of effectiveness. No literature has been found that investigates the relationship between diversity and quality of solutions for algorithms in RT applications as the application of CEAs in this area is still in its infancy.

  In this study, the basic CEA is used with or without variants to enhance the capacity of CEAs in finding a good solution set. In terms of CEA variants, implicit and explicit diversity maintenance and a HOF archive and their combinations are used. Diversity of the population is then measured on the basis of genotypic and phenotypic variation. This study investigates the relationship between genotypic diversity and phenotypic diversity. Additionally, the relationship between the diversity and quality of the evolved population is also investigated in order to analyse whether a diverse population enhances the CEAs‟ performance (in terms of generalisation performance and evolving towards a theoretical optimum if one is known).

- **Examination of multimodality characteristics in RT scenarios.**
  Although Eiben and Smith (2003) stated that most domains demonstrate multiple good solutions, no literature has been found that studied multimodality for RT scenarios. In this thesis, a systematic study was conducted to demonstrate that multiple good strategies can be used in RT scenarios.

In summary, this thesis contributes towards a better understanding of factors affecting the performance of CEAs in RT domains, as well as making useful contributions concerning CEAs in general, especially with regard to performance of CEAs in the presence of intransitivity and multimodality.

1.5 Organization of the Thesis

This chapter described the background information, purpose and the contributions of the study. The following chapters are arranged as follows:

Chapter 2 is a literature review. This chapter is divided into two major sections: the applications of RT and algorithms that can be used to develop solutions for RT. In the applications section, descriptions of RT, overviews of agent-based systems, existing RT optimization methods and their limitations are presented. In the second section; EAs, specifically GAs and CEAs are discussed.

Chapter 3 describes a pilot study in which a RT scenario was optimized using GAs. The chapter demonstrates the performance and limitations of the algorithm in optimizing RT applications when a single population is optimized at a time. This chapter serves as motivation for exploring coevolution as the empirical study revealed that the GAs is unsuitable to deal with realistic RT problems.

Chapter 4 presents problem domains, algorithms and performance measures which are employed in this thesis. The first section presents descriptions of a variety of problem domains that are used to test the algorithms. The second section details the algorithms used in this thesis - GAs and CEAs along with the techniques of diversity maintenance

and archive that are integrated into the CEA. The third section describes various performance measures used to evaluate the algorithms.

Chapter 5 describes an empirical study to test the effectiveness of a CEA on an intransitive number problem. The CEA was used with and without variants to evaluate the effect of archives and diversity maintenance in addressing the problem of intransitivity. A series of experiments were conducted by varying parameters that affect the search process and the relationship between the diversity and quality of the evolved population are also discussed in this chapter.

Chapter 6 involves an empirical study that was used to identify the multiple optima in a multimodal domain. As in chapter 5, a CEA was used with and without variants to evaluate whether the archive only, diversity maintenance only or the combination of both these two techniques can better address the multimodal problem. A new multimodal problem, known as the *n-peaks* problem is introduced in this chapter.

Chapter 7 details the investigation for finding optimal strategies for the red and blue teams in RT using CEA techniques that have been evaluated in chapter 5 and 6. The relationship between diversity and quality in various algorithms tested are also shown in this chapter. In addition, the emerged tactics in both scenarios are demonstrated and analysed.

Chapter 8 summarizes the main findings from this thesis. This chapter discusses how the CEA with commonly used variants behaves differently in different domains studied in this thesis. It also exposes how diversities of a population influences quality in the various domains studied. This chapter also includes a discussion of possible future research directions.

1.6 Conclusion

This chapter provided an overview of RT, purpose of the study, and contribution of the study as well as detailing the structure of this thesis. In the next chapter, the literature review, the studies of the existing techniques, algorithms and applications that are associated with this thesis are detailed.

2 Literature Review

Chapter 1 described the aims and contributions of the study. This chapter reviews literature relevant to this thesis. There are three major sections reviewed in this chapter. They are: RT applications, EAs as search algorithms, and optimization techniques. The first section details a review of manual and automated RT approaches with ABS characteristics. The second section presents a review of EA techniques used in this thesis, specifically focusing on GAs and CEAs. The last section presents a review of existing RT optimization techniques.

2.1 Red Teaming

RT is a process in which a system‟s plans and strategies are analysed to detect vulnerabilities, challenge assumptions and propose a number of alternative visions (Andrews, 2005; Meehan, 2007; Yang, et al., 2006). Fontenot (2005) defined manual RT as a structured and iterative process, practised by experts with a capacity of assessing systems. RT approaches involve understanding the culture, technology, needs, laws, market research, risk factors, available resources and ideological frameworks associated with a system. Furthermore, it focuses on how the enemy thinks (Andrews, 2005; Fontenot, 2005). Likewise the Homeland Security Exercise and Evaluation Program (HSEEP) (2007) defined RT as:

> … a group of subject matter experts (SMEs) of various appropriate disciplinary backgrounds who provide an independent peer review of plans and processes; act as the adversary‟s advocate; and knowledgeably role-play the adversary, using a controlled, realistic, interactive process during operations planning, training, and exercising. (HSEEP, 2007)

Consequently, manual RT is a process that involves a group of experts who play the roles of outsiders or adversaries. The team members work on behalf of enterprises to find weaknesses in systems and ultimately demonstrate a better plan of operations (Meehan, 2007; Yang, et al., 2006). According to Abbas, Bender, Gaidow and Whitbread (2011), RT was introduced to improve the performance of enterprises‟ plans, progress, assumptions and strategies. The purpose behind its use is to reduce risk and

create opportunities (Andrews, 2005; Department of Defence, 2003). Both government and commercial enterprises, such as military, police, and manufacturing companies have successfully used it as a vital tool for successful systems management (Department of Defence, 2003; Meehan, 2007). Within an enterprise, it can be implemented at different levels such as strategic, operational and tactical. Each of these level addresses different aspects such as challenging estimations and visions; testing plans and training for development (Department of Defence, 2003).

According to the Department of Defence (2003), the success of the red team depends on several factors such as management support, its objectives, available information and the relationship with the blue team. The strengths of RT recognized by the Department of Defence (DOD) are listed below:

- Since it expands the problem‟s definitions, spotting vulnerabilities is easier.
- It can view the problem as an adversary or competitor, thus playing a vital role in enhancing decision making by specifying the adversary‟s preferences and strategies.
- As it deals with risk assessment, there is strong potential for creating new opportunities for development.
- RT is a goal-oriented process. Thus, there is always a level of independence and accountability.
- The red team members have access to overall system information which may be utilized to direct a variety of alternative approaches to solving existing system problems.
- Besides eliminating vulnerability, it may propose a new way to deal with various factors such as motivation, connections and risks (Department of Defence, 2003).

Undoubtedly, there are advantages in using RT to find vulnerabilities in enterprises and provide better security. However, it is not completely free from problems. The DOD (2003) discovered some challenges of RT:

- RT is labour intensive; it is heavily dependent on the expertise of the team members and their responsibilities. However, the process may fail if they are not motivated or lack sufficient skills, training or resources.

- The success of RT usually relies on co-operation with management and the blue team but if members do not limit their inter-team interaction, information leakage may negatively affect the entire process.
- If the red team is given limited or incorrect information, it may not be able to achieve its objectives.
- Usually, team members focus within the system, whereas experience has shown that external factors such as the cultural, religious and geographical issues may also influence the success of the process.
- Despite its numerous advantages, RT is still a costly process and may need to be considered on a cost-benefit basis.

According to Fontenot (2005), the history of RT in combat began during the 19th century when the German military developed the war game, „kriegspiel", as a training tool to make team members confident about the training they had received. The war game evaluates leadership, concepts and plans. In 1897, U.S. army captain, W.R. Livermore introduced systematic approaches and rules for a war game called „The American kriegspiel" that included a team specifically denoted as the Red Team. Subsequently, many countries used different RT approaches to understand the opponents" strategies. After World War I, the Germans and the British used RT to review the conduct of the war. The United States (US) also successfully implemented RT during World War II against the Germans. The major reasons behind the U.S. success were the effective analysis of German intentions and the ability to break through the German deception plans (Fontenot, 2005). Traditionally, RT uses a massive number of people to conduct the war game, via human-based or manual RT in which a force is divided into enemy and friend groups, known as the red and blue teams respectively (Sidran, 2004). Manual RT does not explore all aspects of problems and is difficult to implement in term of cost. Thus, manual RT was progressively replaced by various models, including software-based RT which uses computer simulations of multi-agent systems to detect the vulnerabilities in a plan or operation. Computer-based RT also involves two major groups including the red and blue teams. Each of these is a representation of a set of adversarial behaviours and defenders respectively (Yang, et al., 2006).

As shown in Figure 2.1, RT approaches can be categorized into two types, conventional and software models. The conventional model contains manual approaches of conducting RT such as the Lanchester Equations and sand tables. The software model consists of two categories, namely the conventional simulation model and ABS.

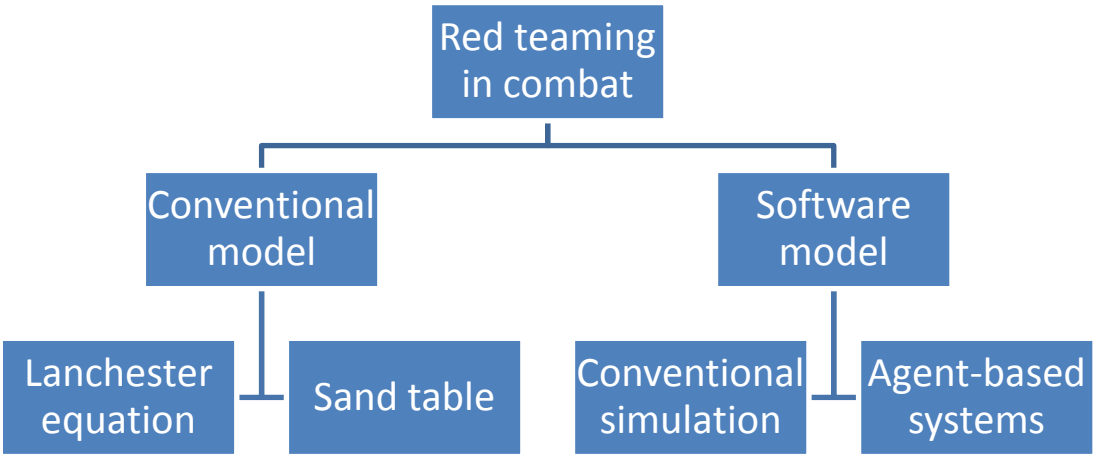

Figure 2.1: Approaches of the RT in combat (Source: Yang, 2006; Parunak, 2007)

2.1.1 Conventional Combat Models

Combat simulations allow commanders to model a variety of scenarios in which they can exercise to find alternative strategies based on specific scenarios. In this section, some of the conventional combat models such as Lanchester Equation and game theory are discussed.

2.1.1.1 Lanchester Equations

Lanchester Equations (LEs) (Lanchester, 1916) are a set of mathematical formulae that calculate the relative strengths of two opposing teams in a battle. LEs are differential equations that consider combatants‟ casualties as a continuous function over time (Yang, 2006). The equations demonstrate how increasing the number of combatant will reduce the total number of casualties (Ilachinski, 2000, 2003).

In the equation, there are representations for two teams, the blue and red. The numbers of soldiers in the blue and red teams are represented by symbols *B* and *R* respectively. *B* (0) and *R* (0) are initial strengths of *B* and *R* team. *R* (t) and *B* (t) are numerical strengths

at a time *t*. Each team has offensive firepower, which corresponds to the number of adversaries the team can defeat per unit time. The constant effective firing rates at one unit of strength on each side are represented by α_b and α_r for the *B* and *R* respectively. In the following equation, the symbol *dR/dt* is the rate of change of the number of red combatants over time *t.* The loss of soldiers is indicated by a negative value. Likewise, *dB/dt* represents the rate of change of the number of blue combatants.

$$\frac{dR}{dt} = -\alpha_b B(t), \quad R(0) = R$$

$$\frac{dB}{dt} = -\alpha_r R(t), \quad B(0) = B$$

As described by Sidran (2004), the Lanchester‟s Equation states that mathematically, the fighting strength of an army is proportional to both the efficiency of its weapons and the number of troops; also, the rate at which one force loses its fighters is proportional to the number of opponents.

However, according to Ilachinski (2000), LEs have the following limitations:

- They ignore the spatial variation of forces such as relation between movement and attrition.
- They do not have event driven behaviours as they neglect the capability of human factors in decision-making and their impact in combat.
- They follow a top-down or reductionist approach to simulate combat.
- They are incapable of addressing the changes in combat scenarios that show different outcomes in combat due to sudden decisions made by local commanders.

2.1.1.2 Sand Table

A well known conventional combat approach, in addition to the use of mathematical equations, is the sand table approach. The sand table can be conducted on a table or the ground in which each soldier on each opposing side is represented by a specific entity. The simulation designer simulates a part of the real world using the environment set up on the ground. The designer physically moves entities to show the movement and interaction amongst the two opposing sides. The movement of entities on the sand table

assists designers to understand the combat scenario which mitigates the vulnerabilities and aids in preparing an effective strategy to respond to a situation. In this type of simulation, once the vulnerabilities in the plan and strategy are identified, real troops have to be engaged to conduct the war game (Parunak, 2007).

2.1.2 Software Simulation Models

Conventional models such as LEs and sand tables provide an idealised model of military operation but have trouble addressing non-linear behaviours associated with combat. Owing to the complexities of the combat behaviours, these conventional approaches may produce unrealistic results. In order to address the limitations of conventional approaches, software simulation was introduced. The basic idea of a software combat model was to represent each entity as a software agent. The approach is inexpensive and faster in comparison to the manual approaches. Initially, software models were introduced as computer games that required user control. These are being superseded by multi-agent systems. Thus, this section is discussed in two parts: conventional software simulations and agent-based systems.

2.1.2.1 Conventional Software Simulation

Conventional software simulation involves a computer program that simulates a combat zone with controllable warriors, semi-automated forces, and an environment, which is directed by one or more players. Some examples of conventional combat simulations are Modular Semi-Automated Forces (ModSAF), Extended LANchester model (ELAN) and combined arms and support task force evaluation model (CASTFOREM) (Ilachinski, 2003; Yang, 2006). Software simulation brought changes in combat simulation since the involvement of a massive numbers of people in the RT process is reduced to just a program and the hardware system. However, Yang (2006) has outlined some limitations in this approach including limited data collection and analysis facilities; focus on high level instead of low level competencies; difficult to use due to unavailability of user friendly interface and high dependence on users.

2.1.2.2 Agent-Based Systems

In a computer simulation that simulates a part of the real world, an agent is a software or hardware component with the capacity to represent an entity of the real world (Buckle et al., 2002; Bui & Barlow, 2003; Cleary, Ball, Madahar, & Thorne, 2008). An ABS possesses characteristics similar to many complex adaptive systems (CASs), including that of emerging global behaviour through interactions of low-level individual components or agents (Buckle, et al., 2002). The following are some characteristics of an ABS (Flores-Mendez, 1999; Jennings, Sycara, & Wooldridge, 1998; Liu & Zhang, 2008; Schumacher, 2001).

- Every entity has its specific goal whilst lacking awareness of the overall objective of the system. The overall goal of the system is a black box to the individual agents and vice versa.
- It is built on bottom-up approaches since individual entities have limited viewpoints and there is no central controller in this system.
- It is a fault tolerant system as any agent can substitute for the failure of another agent.
- Since it is a robust system, an ABS may be reconstructed by removing or adding a number of agents.
- An ABS system is reusable since the environment created for one application may be implemented in another application with necessary modifications.
- The data is decentralized and distributed to the agents and the environment.

In software combat simulation, the limitation of conventional models can be addressed by introducing agent-based systems (Ilachinski, 2003), which are based on the principle that complex behaviours emerge due to the autonomous agents‟ interactions. The autonomous agents could be homogeneous or heterogeneous. The effect of using an ABS is that the behaviours and interactions of individual entities are modelled rather than a coarse grained model of a group of entities as a whole (such as with the Lanchester Equation) (Lauren, 1999; Levin, 2002).

Many researchers have successfully implemented ABS in a number of application domains such as a computer-based tutoring system (Shakshuki & Kajonpotisuwan, 2002), mobile robots (Innocenti, Lopez, & Salvi, 2003), an error-detecting system for

spreadsheet applications (Clearly et al., 2008), system for controlling intelligent buildings (Davidsson & Boman, 2000) and in production and distribution systems (Davidsson & Wernstedt, 2002). A multi-agent system (MAS) is a type of ABS that includes many such agents with their specific environments and processes of achieving goals. When a group of autonomous agents with local behaviours interact among themselves and with their environment, they can produce an emergent global behaviour (Cleary, et al., 2008; Poslad, Buckle, & Hadingham, 2000).

Consequently, ABSs have proven to be versatile and useful, able to be applied in a number of domains. One domain is that of combat simulation for RT, in which agents play the role of defenders and attackers.

Agent-based distillation (ABD) is a term associated with ABS in which the level of detail associated with the modelling of individual agents is very low (Bui & Barlow, 2003). Due to their simplicity, ABDs are suitable in applications where multiple agents need to be simulated, and where the simulation needs to be repeated many times. One example is in evolving strategies for combat scenarios, as addressed in this thesis. Several simulators already exist in this domain. Some of these include:

- Irreducible Semi Autonomous Adaptive Combat (ISAAC)
- Enhanced ISAAC Neural Simulation Toolkit (EINSTein)
- Conceptual Research Oriented Combat Agent Distillation (CROCADILE)
- Warfare Intelligent System for Dynamic Optimization of Missions (WISDOM)
- Map Aware Non-Uniform Automata (MANA)

2.1.2.2.1 ISAAC

Irreducible Semi Autonomous Adaptive Combat (ISAAC) is a land combat simulation, developed by Ilachinski (1997) for the US Marine Corporation. It is built in ANSI C and executes in DOS in command mode. This software is considered as the first agent based simulation of small unit combat used by the military operations research community. The basic foundation of ISAAC is modelled according to mobile cellular automata (CA) rules. The basic element of ISAAC is an ISAAC agent (ISAACA). Agents are basic elements in an ISAAC model that represent all components used in a combat system, including infantryman, tank, transport and other vehicles. Every agent

possesses four characteristics, namely doctrine, mission, situational awareness and adaptability. These characteristics represent: a default strategy to be taken in a generic environment; a main goal or objective that directs behaviour; a sensor-generated internal map of an environment; and a technique for changing behaviour of agents to the changing environment respectively (Ilachinski, 1997).

Each agent considers six factors; four of them represent an agent‟s personality such as how the agent behave with alive friend, alive enemy, injured friend and injured enemy troops within the agent sensor range. The other two factors relate to the goal of capturing the blue or the red flags by opponent teams. Each agent has seven capability attributes including probability of hit, maximum targets, sensor, fire, movement, communication and threshold ranges. The „probability of hit‟ indicates the possible number of shots that an agent can endure before getting killed within the firing range. Maximum target is a maximum number of targets that an agent can engage with, at any single time step. Sensor range performs the job of recognizing other agents in its surroundings. Fire range represents the suitable distance to fire upon an opponent. Movement range is the number of grid squares an agent can move in a single step. Communication range is the distance within which an agent can interact with other friendly agents. Threshold range is the area within which the number of other agents is taken into account when deciding on the next move (Ilachinski, 1997).

2.1.2.2.2 EINSTein

Enhanced ISAAC Neural Simulation Toolkit (EINSTein) (Ilachinski, 2003) is a successor to the text-based combat simulator, ISAAC. EINSTein includes a graphical user interface (GUI) that helps in visualizing agents‟ activities. Most of the characteristics of ISAAC are incorporated EINSTein. Additionally, some characteristics from ISAAC are improved and integrated into EINSTein, described below:

- There are five run modes including interactive, playback, multiple time series, two-parameter fitness and the use of optimization technique using a GA. The GA optimizes only one team at a time, against a fixed opponent‟s strategy.
- Three graphical views are available namely main battlefield, race and combat view. Each of them represents terrain elements, territorial occupancy and combat intensity respectively.

- On-line data collection and visualization of recorded combat is available.
- Additional meta-personality behaviour added into EINSTein include hold, pursuit-I, pursuit-II, Retreat, Support-I and Support-II characteristics.
- Different levels of agents are defined such as: combat agent, local command, global command and supreme command level.
- Multiple terrain types, which can be used as per the requirement of users, are available within the system. Users can choose any terrain environment such as roads, woods and rivers. Besides the default terrain, user may use imported or manually designed terrain as well.
- Independent multi-squads are capable of interacting and making decisions as per the situation.

2.1.2.2.3 CROCADILE

Conceptual Research Oriented Combat Agent Distillation (CROCADILE), developed by M. Barlow and A. Easton in 2003, is an object oriented ABD built in the Java programming language. It consists of features including 2D and 3D battlefield environments, user extensible agent behaviours, backup of combat objects, multi-team structure, probabilistic or projectile-physics combat resolution and improved communication structure. Each agent can be modelled with firepower, mobility, sensing, communication and command. The six personality features of the agent including the conditions of dead, health, force-ration, mission, timing and command triggers are defined in this ABD. User defined terrain and integrated digital terrain maps can be used in the system (Bui & Barlow, 2003).

2.1.2.2.4 WISDOM

Warfare Intelligent System for Dynamic Optimization of Missions (WISDOM) was initially developed by Yang, Abbas and Sarker (2003). It was subsequently extended to WISDOM-II by using the concept of network centric multi-agent architecture (NCMAA) in 2004. There are five components in WISDOM-II namely Command, control and communication (C3), sensor, engagement, visualization and reasoning. The C3 component contains command, control and communication functionality. The sensor component receives information from the environment. The engagement component

contains firing and movement activities. The visualization component demonstrates visual information such as graphs. The reasoning component interprets the results in natural language during the simulation process (Yang, 2006).

WISDOM-II has four defined types of agents: combatant, group leader, team leader and general commander in which the team leader and general commander are the virtual agents. Each agent has different characteristics such as health, skill, obedience, visibility, vision, communication, movement and engagement. Various weapons, including explosives, direct and indirect weapons, have individual characteristics such as fire power, range and damage radius (Yang, 2006).

2.1.2.2.5 MANA

The ABD, Map Aware Non-uniform Automata (MANA), is used as a simulator to run military scenarios for this study. MANA is a cellular automaton combat simulation model, designed by M. K. Lauren in 2001 at the New Zealand Defence technology agency (DTA). MANA was inspired by ISAAC and EINSTein, hence, most of the characteristics defined by Ilachinski, including strategy of agents‟ movement based on its surrounding cells, are also implemented in MANA (Yang, 2006). MANA was introduced as a scenario exploration model to address some of the limitations of conventional combat simulation models. The limitations of user dependence in conventional models are addressed by providing an environment in which interactions of autonomous local entities produce an emergent global behaviour. The autonomous behaviours, which reduce the user involvement on a planning process, produce a realistic outcome. There are many versions of the MANA simulator, MANA version IV introduced in 2004 was utilized in this study. The simulator contains some extra features which were not addressed in early ABDs including:

- Situational awareness (SA): Each squad uses a common platform for interchanging information among squad members, called a SA map. The concept of a SA map for squads is implemented in two ways, the squad SA map, which allows sharing information amongst squad members, and the inorganic SA map, which works with the information obtained from sensors or other squads.

- A communication model: This is one of the unique characteristics included in MANA IV. Agents in MANA communicate in various ways, individual, inter-squad and intra-squad.
- Terrain map: It is a map that demonstrates the type of environment in which agents are situated. The environment can be created using various terrain types that are available, such as billiard table, easy going, wall and hilltop which represents plain terrain, smooth road, blockade and bushlands. Besides the available combat scenarios, users may import their own terrain.
- Way-points: These are used to guide the agents to a particular destination. Users can pre-define or change the way-points during a simulation.
- Event-driven personality changes: Certain events such as being shot at, taking a shot, reaching a waypoint and enemy contact, trigger different personality in agents; this change lasting a certain period of time. The effect of personality changes can be triggered for an agent or for a whole squad.
- Data farming: MANA is built in data farming capability facilitate fully automated mapping of a scenario‟s parameter space.
- Furthermore, built in data farming capability facilitate fully automated mapping of a scenario‟s parameter space.
- Movement constraints: The constraints associated with movement can be used to prevent agents moving towards a goal without a minimum number of friendly agents in their squad. The combat type determines the required number of minimum agents before approaching the enemy.
- Capabilities of agents: An agent‟s performance or capability is fully dependent upon its weapons, sensors, movement, speeds and interactions among agents and environments.

Table 2.1: Evaluation of existing ABDs

Features	ISAAC	EINSTein	MANA	CROCADILE	WISDOM
GUI interface	No	Yes	Yes	Yes	Yes
Developed in	C++	C++	Delphi	Java	*
Cellular automata	Yes	Yes	Yes	Yes	Yes
SA map	No	No	Yes	Yes	Yes
Saving scenario	No	Yes	Yes	Yes	Yes
Terrain selection	No	Yes	Yes	Yes	Yes
Genetic Algorithm	No	Yes	Yes	Yes	Yes
Multi-run facility	No	Yes	Yes	Yes	Yes
Simulation type	Land	Land/air/water			
Agent architecture	Reactive				NCMAA

Source: Ilachinski, 2004; Lauren, 2005; Lui & Barlow, 2003; Yang, 2006

* Not found in the literature

2.1.2.2.6 Analysis of ABD Simulators

An analysis of existing ABDs was conducted, a summary of features is provided in Table 2.1. The analysis indicated that all simulators have their pros and cons. Some of the points that were discovered during the investigation are:

- WISDOM-II supports a hierarchy of combatants such as, team leader and general commander agent; however, their strengths and effects are still unclear in the process.
- WISDOM-II uses a network centric multi-agent architecture (NCMAA) modelling that defines the relationships between the agents as a network; however, gaps exist for agent communication within and between squads. For example, there are no provisions for reinforcement of troops in emergency situations.
- Despite introducing 2D and 3D terrain features in CROCADILE, the large influence on movement of valleys, rivers, hills and mountains were not satisfactorily addressed.
- EINSTein, MANA, CROCADILE and WISDOM all feature a variety of battlefields including land, air and water. However, their utilizations are limited to an individual type of battlefield. Exchanges between air combatants and land and water combatants and vice versa are yet to be developed in these ABDs.

- Department of Defence (DOD) has recognized that social, cultural and religious factors can influence combat outcomes (Department of Defence, 2003); however, those factors are not included in existing ABDs.
- MANA has defined the number of hits required to kill an enemy combatant; however, it does not allow for varying degrees of injured combatant such as some whom are able to continue to battle and some who cannot.
- Most of the ABDs have defined squads of warriors who maintain their distance from each other but yet stay together all of the time. However, existing ABDs have not addressed the possibilities of splitting groups according to changing scenarios, for example, to surround another team.
- During combat, combatants might seize high-tech weapons from their adversaries, which then could be used against an enemy squad. These possibilities are not addressed by the current ABDs.

The analysis above highlights some of the limitations related to the functionality of existing ABDs in which a combat scenario is simulated to identify vulnerabilities in a plan. In general for all ABDs, the individual agents interact together and with their environments. From these interactions there emerge global behaviours. Most simulators available share the limitation that their scenarios are based on characteristics of entities which are user-defined. Simulators on their own do not recommend optimal values of characteristics for entities so that they can respond to various situations. Finding vulnerabilities in a team‟s plans is very complex as there are numerous characteristics to consider. Therefore, researchers including Upton and McDonald (2003), Hingston (2011) and Choo et al. (2007), used techniques in which simulated scenarios were combined with search algorithms to identify appropriate entities‟ characteristics to best suite the scenario. The most commonly used search algorithms for this task are EAs. The following section presents a review of the use of EAs.

2.1.3 Real Time Strategy Games

Simulation and computer-based games have become a crucial part of modern military training and entertainment industry respectively (Buro, 2003). These two fields have commonality in terms of using a real-time AI technique in developing applications

(Herz & Macedonia, 2002). Real-time games are computer war games which are based on simulations in which each character performs their role simultaneously in real time (Schadd, Bakkes, & Spronck, 2007).

There are some difference between turn-based games and real time games. In a turn-based game, such as a Chess game, a player and the opposition take action turn-wise. Users can view all the actions of the opposing players as the entire playground is displayed on the screen. However, in real time games each entity operates in real time. Since the game is based on a scenario, users can display a global view of the scenario to observe the overall activity of entire players or can concentrate on only a specific part of the scenario. Maps are an essential part of real time games without which users cannot effectively play the game (Johnson, Yannakakis, & Togelius, 2010).

Real time games are divided into two categories, first person shooter (FPS) and real time strategy (RTS) games (Claypool, 2005). In FPS games, a part of the scenario is displayed on the screen as seen by the eyes of character. In the entire scenario, a battle occurs only in a particular part of the scenario in which the main player exists. Entities do not perform an active role in other parts of the scenario in which a main player does not exist. *Maze War, Quake* and *Doom* are some examples of FPS games in which the players move towards a destination by defeating deadly monsters and collecting armours and weapons. Unlike in FPS games, entire entities perform their own duty within a scenario in RTS games regardless of which portion of the scenario is displayed on the users screen.

RTS games use a centralised server in client-server architecture with multiple users over the network or also can be played against AI opponents within a single computer system. Games are played in a provided scenario or users can built their own scenario. A scenario consists of an environment, workers, army and invaders. *Command and Conquer, Starcraft, Warcraft, Age of Empires* and *Age of Mythology* are some examples of RTS games. For example, according to (Claypool, 2005), in *Warcraft III*, there are three major units: building, exploration and combat. The building unit involves construction of infrastructure and upgrading buildings during a war. These tasks are mostly performed by the worker entities who are also involved in gathering resources

such as money and materials. Geographical information is gathered by entities in exploration units which helps to locate enemy towns or units. Combat units consist of attackers and defenders whose purpose is to defeat and destroy invaders.

With increasing popularity of RTS games, researchers have used various algorithms to find suitable strategies and tactics that defeat enemy agents in a scenario. Ponsen (2004) used a GA to optimize strategies of a *Wargus* RTS game scenario in which each chromosome consists of 4 types of genes: build, research, economy and combat. Jang, Yoon and Cho (2009) optimized scenarios of *Command and Conquer* using EAs. Churchill and Buro (Churchill & Buro, 2012) used heuristic search-based AI system, depth first branch and bound algorithms (Churchill & Buro, 2011), to optimize strategies in *StarCraft*. Likewise, Togelius et al. (2010) used MOEA and Mora et al. (2012) and Othman et al. (2012) used EAs to optimize strategies in *StarCraft* scenarios. Miles (2007) and Avery and Louis (2010) used a CEA to optimize strategies in *LagoonCraft* and *Capture-the-flag* RTS games respectively. The authors considered building, finance and combat as evolving factors. Munir, Kitchin and Crampton (2010) used a Monte-Carlo planning approach, Upper Confidence Tree and Rapidly Exploring Random Tree (Chung, Buro, & Shaeffer, 2005) in solving the path finding problem in an RTS game called *RC-RTS*.

There are some significant differences between RTS games and MAS combat RT. In RTS games, an entire island or town is simulated and each entity in the simulated area performs simultaneously. A player or multiple players control entities‟ activities in the scenario especially to set the preference of entities tasks such as building infrastructure and attacking enemies. In terms of optimization, factors such as building infrastructures, gathering resources and finding optimal strategies to win over enemies using available resources are mostly used in RTS games. However, in combat RT, such as in MANA, a small part of the real world is simulated as a scenario which enables to clearly view each entity‟s interaction that allows an analyst to determine the strategy to be incorporated. Unlike in RTS games, users are not involved in combat scenarios but autonomous agents interact with each other and also with their environment to achieve their own goal. In optimization of a combat scenario, the environment remains

unchanged but strategies evolve on the basis of agents‟ decision making (for example, personality and waypoints).

2.2 Evolutionary Algorithms

Evolutionary algorithms (EAs) were first introduced in the late 1950s as utilities for finding optimal solutions to problems (Coello, Lamont, & Veldhuizen, 2007). In mathematical terminology, “optimization” is defined as the process of minimizing and maximizing given specific criteria (Coello, 1999a; Veldhuizen, 1999). EAs apply biological principles of natural evolution to artificial systems. The foundation of EA is Darwin‟s evolution theory; the EA suggests that problems may be solved by an evolutionary process that selects the best solution from a population (Hussein A Abbass, Sarker, & Newton, 2001; Abraham, Jain, & Goldberg, 2005; Alcala, Alcala-Fdez, Gacto, & Herrera, 2007; DeJong, 1975; Eiben & Smith, 2007; Veldhuizen, 1999; Zitzler, 1999). According to Coello et al. (2007), EAs are adaptive heuristic search algorithms that derive the set of best solutions by using natural selection: each solution gets a chance to reproduce a certain number of times depending on its performance. The specific steps of the search process for selecting the set of best solutions iterates until the desired results are attained. Thus, quality results are achieved by selecting among the set of best solutions. (Hussein A Abbass, et al., 2001; Abraham, et al., 2005).

EAs have been successfully implemented in various sectors, including ecology, machine learning, automatic programming, population genetics, economics, optimization, social systems and operations research (Abraham, et al., 2005; Coello & Pulido, 2001; Eiben & Smith, 2007; Zitzler, 1999). Several evolutionary search methodologies have been proposed, depicted in Figure 2.2 in the realm of computational intelligence, namely GA, evolution strategies (ES), multi-objective evolutionary algorithm (MOEA) and CEA.

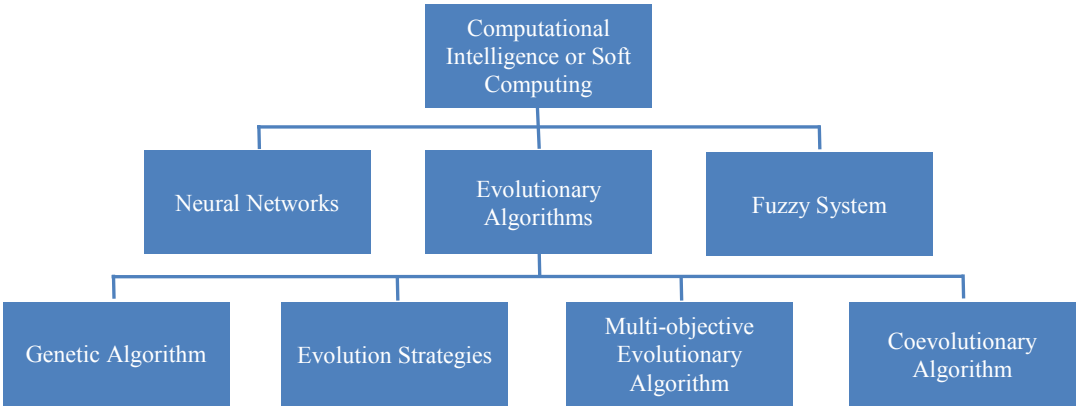

Figure 2.2: Taxonomy of evolutionary algorithms in the computational intelligence realm.

All EAs, including GA, ES, MOEA and CEA, are population-based search algorithms. The search begins with a population of the potential solutions (Cliff & Miller, 1995; Ficici, 2004; Ficici & Pollack, 2001; Guo, Cao, & Yin, 2007; Seredynski, Zomaya, & Bouvry, 2003). All EAs have a common strategy of selecting the best solutions. Figure 2.3 depicts the general steps involved in generic EAs, including CEA. The process begins by generating random individuals for a population. The individuals are possible solutions. In the second step each individual is evaluated and ranked according to some objective function, called fitness value. Thirdly, selection methods consider individuals for crossover or mutation; individuals with a higher fitness value have higher chances to participate in breeding. Crossover is a sexual process in which two individuals participate, whereas mutation is an asexual process which involves only an individual for the breeding process. Fourthly, an offspring is generated for the next generation. The entire process of creating generations is repeated until a set of satisfactory outcomes is obtained.

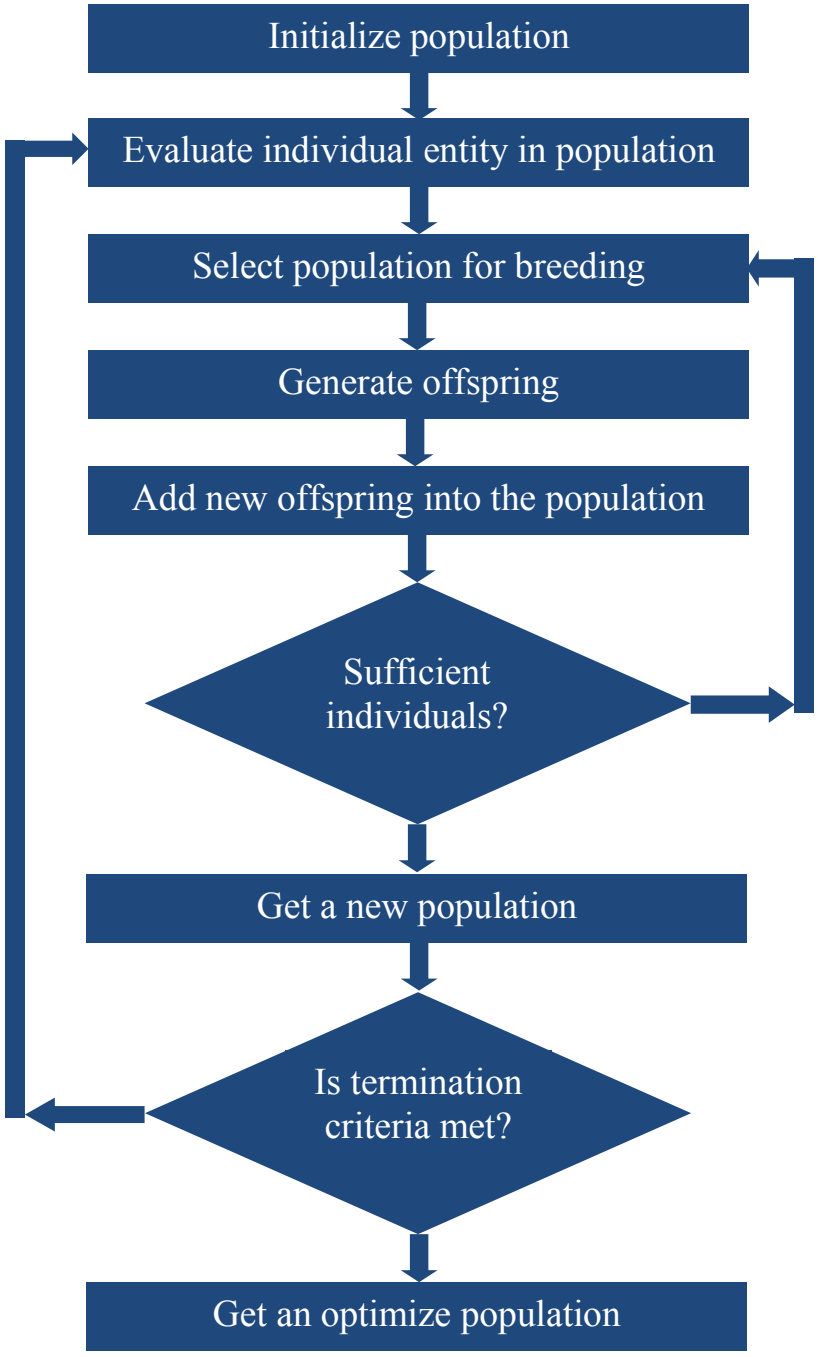

Figure 2.3: Flowchart indicating EA processes (Source: Ficici, 2004)

Although all EAs have some common principles, there are also some variations which are briefly described in the following sections. Since this research utilized a GA and CEA, the focus will be given to these two algorithms.

2.2.1 Genetic Algorithms

There are four important components in a GA: an individual that represents a candidate solution; a fitness function that evaluates an individual; selection of the fittest parents; and reproduction to generate new individuals using crossover and/or mutation (Goldberg, 1989). Chromosomes are the representation of strategies which also play a role in generating new genetic material for the subsequent generation by transforming the „genes" (Goldberg, 1989).

The general steps that occur in GAs are depicted in the flowchart in Figure 2.4. Generally, individuals are created by randomly generated numbers or strings that represent a solution. There is an objective function to calculate the fitness of individuals in GAs which varies according to the domain. If elitism is implemented, a solution with the highest fitness value, an elite solution, is copied to the subsequent generation without any modification. Other „good" solutions are then selected, based on the fitness value using a selection algorithm. According to (Pohlheim, 2006), some common selection algorithms are roulette wheel, stochastic uniform and tournament selection. Using a selection algorithm, two parents are selected. In order to generate offspring, the selected parents undergo crossover and mutation and produce new offspring which are then added to the new population. There are various types of crossover and mutation available such as single-point and two-point crossover and bit flips, Gaussian and polynomial mutation (Deb & Goyal, 1996). As shown in Figure 2.4, the process of evaluation, selection and producing new offspring continues until the termination criterion is satisfied. In the figure, the termination criterion shown was a specified number of generations; however, it could also be another factor such as amount of time, minimum fitness threshold satisfied or whether fitness has reached a plateau.

Although GAs have been used as effective search optimization techniques for many application domains, they also have limitations. Issues associated with GAs include their inability to be applied in the absence of objective measures and in large search spaces when there are Cartesian products involving two or more interacting subspaces. The Cartesian product in EAs arises when each individual from a population interacts with every individual in one or more populations. It has been found that CEAs are capable of overcoming these limitations.(Wiegand, 2003).

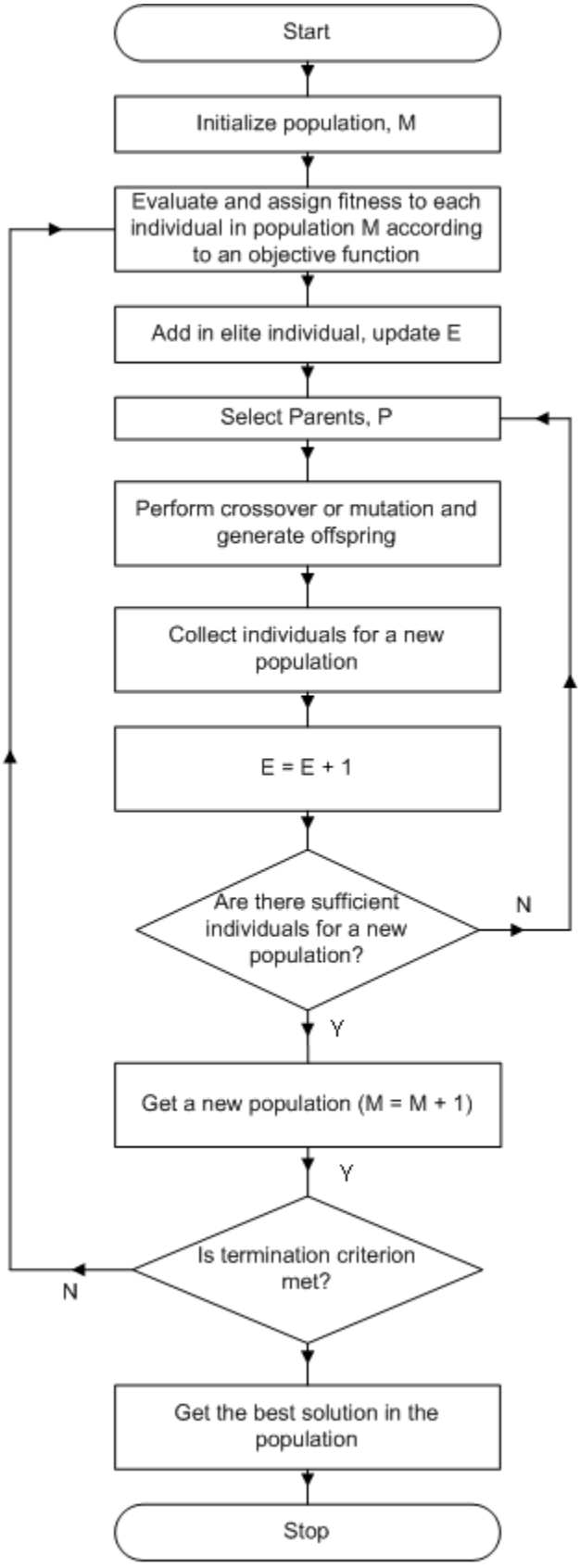

Figure 2.4: Flowchart showing various steps of GAs

2.2.2 Evolution Strategies

Evolution strategy (ES) is a branch of EAs. The purpose of an ES is to optimize an objective or quality function. A set of decision variables or control parameters are tested in a given objective function. The major steps involved in this approach are: evaluation, selection and reproduction. The evaluation process begins by providing fitness values to each individual. For the selection process, parents are selected based on fitness. Subsequently, the selected solutions go through the recombination and mutation processes to generate a child solution. The newly created child solution again competes against the current solution and the winning solution is added to the population for the new generation (Eiben & Smith, 2003).

There are some differences between ES and GAs in terms of evolving a new population. Traditional GAs used binary representation and ES used real numbers; however, a real coded GA also uses real numbers representation. A major difference between these two algorithms is the selection of individuals for a new population. In GAs, two individuals are selected on the basis of their fitness value and offspring are produced by using a crossover and mutation process. The new offspring become a part of a new population. However, in ES, typically, an individual is selected on the basis of the fitness value and a new offspring is produced by mutating the individual. Subsequently, a newly created offspring and the parent individuals are compared and the best fit individual joins the new population.

The ES involves the following steps (Jacob, 2001):

1. Initialize the population P
2. Evaluate each individual in population P using a specific objective measure (fitness function)
3. Select two parents i_{p1} and i_{p2} from P
4. Recombine and/or mutate the selected individuals i_p to generate i_c
5. Generate a new population by using any one of the following methods:
    a. Select the best individuals from a set of parents, i_p, and a new offspring, i_c. In this process i_c and i_p compete and the best one (fittest individual) either a child or parent stays in subsequent generations

b. Select the best individuals from a set of parents, i_p, and a new offspring i_c. In this process a best parent and a best offspring move to subsequent generations

6. Repeat from step 2 until a termination criterion is satisfied.

2.2.3 Multi-Objective Evolutionary Algorithm

One of the emerging research areas in which EAs have been successfully employed is that of multi-objective optimization problems (MOPs) (Coello, 1999a, 1999b; Coello, et al., 2007; Coello & Pulido, 2001). There are many problems in the real world that involve many factors which need to be considered in order to identify a competent solution. (Hiroyasu, Miki, Iwahashi, Okamoto, & Dongarra, 2003). For example, in order to design a new model of an existing car, many factors need to be considered including wheels, body, bumper and engine. All factors cannot be considered as a single objective problem; in order to optimize a car design it is not sufficient to provide only a better engine. Some factors in MOPs complicate the process of finding the best solution. Optimising each factor on its own often does not lead to an optimal solution, rather there are several combinations of factors that produce individual optimal solutions along a Pareto front. To deal with MOPs, a class of EAs, known as multi-objective evolutionary algorithms (MOEAs) (Abraham, et al., 2005) can be utilised.

The first real implementation of an MOEA was in 1980 (Zitzler, 1999). Since then, a considerable amount of research has been done in this area. According to Zitzler (1999) and Coello (1999a), many MOEA approaches have been introduced, some of these are listed in Table 2.2. These algorithms are capable of concurrently searching for solutions in a single simulation run.

Table 2.2: Various approaches of MOEA

Approaches	Inventors	Date
Vector Evaluated Genetic Algorithm (VEGA)	David Schaffer	1985
Weighting-based Genetic Algorithm (WGA)	Hajela and Lin	1992
Multi objective GA (MOGA)	Fonseca and Fleming	1993
Niched Pareto GA (NPGA)	Horn and Nafpliotis	1993
Non-dominated Sorting GA(NSGA)	Srinivas and Deb	1994
Strength Pareto EA (SPEA)	Zitzler and Thiele	1998
Pareto Achieved Evolution Strategy (PAES)	Knowles and Corne	2000
Pareto Envelop-based Selection Algorithm (PESA& PESA-II)	Corne et al.	2000
Strength Pareto EA (SPEA2)	Zitzler et al.	2001
Micro-GA	Coello	2001
Niched Pareto GA (NPGA-II)	Erikson, Mayor and Horn	2001
Non-dominated Sorting GA (NSGA-II)	Deb et al.	2002
Indicator-based Evolutionary Algorithm (IBEA)	Zitzler and Kuzli	2004
Multiple Criteria Decision Making (MCDM)	Thiele, Miettinen, Korhonen and Molina	2009
Reference Point Evolutionary Algorithm (RPEA)	Figueiraa, Liefooghe, Tabli and Wierzbickie	2011

Sources: Zitzler (1999), Coello (1999a), Alcala et al. (2007), Coello et al. (2007), Zitzler & Kuzli (2004), Thiele, Miettinen, Korhonen, & Molina (2009), Figueiraa, Liefooghe, Talbi, & Wierzbickie (2010)

2.2.4 Coevolutionary Algorithms

The CEA involves a process of mutual adaptation which emerges when agents in some domains interact for a purpose. The process involves multiple entities that attempt to find the optimal solution for a problem by interacting with each other. The fitness of each entity depends on its relationships between other entities and their interactions. In CEA, a set of individuals is categorized into different sub-populations which coevolve, whilst influencing each other (Ficici, 2004). Wiegand (2003) describes a CEA as a collection of EAs in which fitness is determined by the interaction of the individuals involved. The fitness measure, on the basis of individuals‟ interaction, is called a subjective measure. Wiegand (2003) argued that some limitations of GAs, such as their inability to be applied in the absence of objective measures and their problems when applied in complex problem domains, can be addressed by introducing CEAs since they do not rely on objective fitness measures.

The differences between generic EAs and CEAs are summarised in Table 2.3. Like other EAs, CEA is also a population-based search algorithm which consists of a population of potential solutions. It may contain one or more initial populations in which the fitness of individuals determines the possibilities of selection for reproduction (Casillas, Cordon, Herrera, & Merelo, 2002; Guo, et al., 2007). In contrast to traditional EAs, CEAs are believed to be effective in searching for solutions to problems with complex structures which do not have intrinsic objective measures and may have infinite search spaces (Sameer et al., 2012). The evaluation process is another dissimilarity, which requires interaction between multiple individuals. Depending on the search problems, the interacting individuals may be members of the same or different populations (Ficici, 2004; Guo, et al., 2007; Rosin, 1997). When applied specifically to a military RT process, a traditional EA can propose the best solution in response to an opposition‟s single tactic. When an opposition‟s ability to respond and change tactics is recognized, traditional EA cannot be effectively utilized to address the situation. Instead, a CEA has the potential to deal with such situations by modelling the opponents‟ tactics as another, also evolving population.

Table 2.3: Difference between EAs and CEAs

Generic EAs	**CEAs**
Suitable to apply for those problems in which intrinsic objective measures are known. Fitness is absolute or objective.	Suitable to apply for problems in absence of intrinsic objective measures to evaluate fitness of individuals. Fitness is relative or subjective.
The selection process examines each individual and rates it according to some objective functions to produce fitness values. Higher fitness values indicate higher possibilities of becoming parents.	The fitness values, which determine the possibilities of becoming parents, are calculated according to an individual‟s interaction with other internal or external entities rather than a known objective.
Since a population competes against an objective entity, the fitness of evolving population usually gets better in each generation until it converges as measured against that fixed objective entity.	Since evolving individuals are evaluated against other individuals that are also evolving, the problem of “relativism” occurs, where the fitness values may not converge.

Axelrod (1987) and Hillis (1990) successfully adapted the concept of coevolution as a general problem solving technique. Axelrod (1997) investigated a game called iterated prisoner‟s dilemma (IPD) in which two players play prisoners‟ dilemma repeatedly. In

this game, when two players meet again, they remember their opponent‟s action in their previous meeting and change their strategy accordingly. A GA was utilized to study the prisoner‟s dilemma problem (Axelrod, 1987) in which a player‟s choice is either „cooperate‟ or „defect‟. The choice of defection leads a player to a higher payoff than cooperation; however, the payoff is lower when both parties choose to defect rather than to cooperate. Axelrod (1987) found that when a player repeatedly meets multiple players, each possessing a unique strategy, cooperation strategy appeared to be more effective than the defection strategy. However, the best strategy found was called tit-for-tat. In this strategy, a player cooperates on the first move of the game. Subsequently, the player responds according to how he/she was treated on the previous meeting.

One characteristic of coevolving competing populations is the occurrence of an “arms race” phenomenon. In an arms race a change in one population forces others to improve the quality of certain behaviours. The population may not appear to be better than the opposing population as both evolve simultaneously. However, better performance can be seen when the evolved populations are measured against a fixed external test. Hillis (1990) demonstrated the use of CEAs in complex problem domains that demonstrate an „arms race‟. In this domain, one population represents a sorting network and another population represents data sets. The first population coevolves by seeking better devices to handle the data sets, whereas the other population coevolves by making itself more complex for the sorting network. The approach of evolving the network and data sets simultaneously prevents the problem of getting trapped in local optima because the opposing individuals encourage generation of new data sets and networks in every generation. This approach also increases the efficiency of testing networks and data sets by removing tests that are ineffective, which speeds up the optimization process in subsequent generations. With this method Hillis (1990) was able to generate a sorting network that was more effective than any found previously.

Likewise, there have been many other applications developed utilizing CEAs. Miller and Todd (1993) applied CEA to investigate the role of sexual selection in evolution. One population represents male characteristics and another represents female attractions towards the male depending on their characteristics. Potter and DeJong (1994) also used CEAs in static function optimization and neural network learning.

There are two types of CEAs: competitive and cooperative (Ficici, 2004; Wiegand, 2003). In competitive CEAs (CCEAs), the evolving population tries to defeat all individuals from other populations. The repetitive evolution process tends to become an „arms race" in which populations eventually improve their behaviours, such as in the predator-prey model (Wiegand, 2003). Ficici (2004) defined CCEA as a zero-sum game in which advancement of one population also forced other populations to advance. As both populations improve their performances, the overall outcomes remain unchanged. Unlike CCEA, in cooperative CEAs (COCEA), the populations coevolve by supporting each other and this helps them to advance simultaneously. During the coevolution process, the populations support each other to improve their quality so that they can both benefit. The bee/flower model, in which the bee benefits from collecting the food and the flower benefits from pollination, provides an example. Ficici (2004) highlighted the need for a work break-down structure (WBS), which he considered as a major characteristic of COCEA, to deal with a large complex problem. A complex problem decomposes into a set of small problems. Coevolving those small sets collectively presents the solution for the complex problem.

Since this study uses a CCEA, all analyses of CEAs in the remainder of this thesis represent CCEAs. Likewise, a CEA may also be used with a single population; however, this thesis concentrates only on multiple populations (Ficici, 2004). In a CEA (as in other GAs), optimization begins with generating a population then various operators are applied for evaluation, selection of the parents, elitism, crossover and mutation. The descriptions of these components will now be detailed:

2.2.4.1 CEA Components

The algorithms require various operators in optimizing any domains. There are various operators available for conducting each stage of the optimization. The stages are evaluation, selection, elitism, crossover and mutation. The following sub-sections discuss the specific operators that were chosen in this thesis.

- **Evaluation of Individuals**

DeJong (2004) explained a number of ways to calculate an individual‟s fitness value relative to another population of individuals (subjective fitness). Figure 2.5 shows three interaction methods: all versus best, all versus some and all versus all. In the first method, each individual in one population receives a fitness value by only competing against the best individual in the opposing population, whereas in the second method, each individual in one population receives a fitness value by competing against a subset of randomly selected individuals in the opposing population. In the third method, every individual competes against all members from the opposing population in order to get its fitness value. All three methods have their specific pros and cons. The first two methods may be computationally faster but may not be very effective as individuals are evaluated against one or a subset of the potential strategies in the opposing population. The third method evaluates an individual more rigorously than the first two methods; however, it is computationally more intensive. In order to ensure more rigorous evaluation of the individuals the third method was chosen for this thesis.

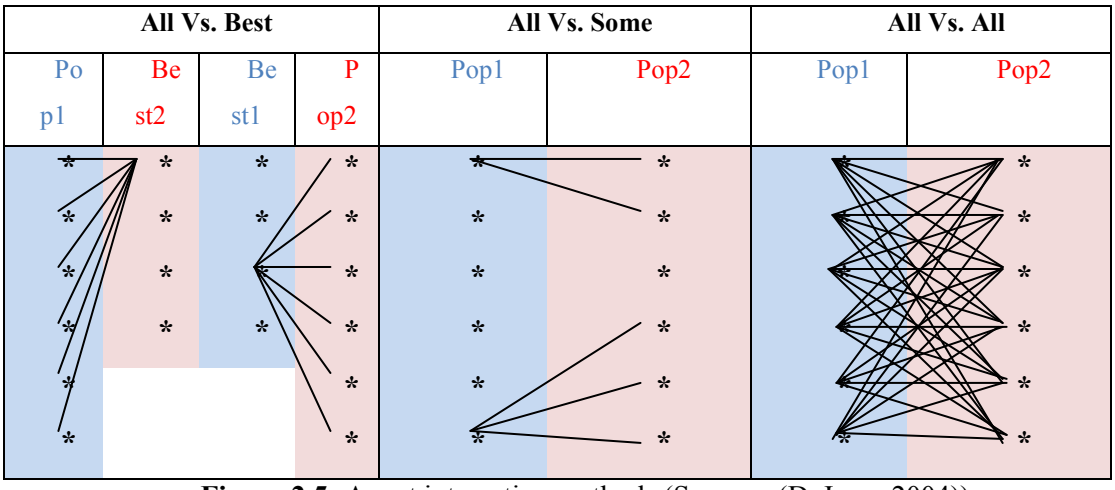

Figure 2.5: Agent interaction methods (Source: (DeJong, 2004))

The fitness depends on the evaluation of an individual with its competing population, which makes the fitness dynamic rather than static (DeJong, Stanley, & Wiegand, 2007; Ficici, 2004). Equation (2.1) shows how subjective fitness f_i of individual i is calculated in a naïve CEA.

$$f_i = \frac{1}{\mathrm{n}} \sum_{j=1}^{n} score_{i,j} \quad \textbf{(2.1)}$$

Each individual *i* from one population competes against each of its opponent *j* from the other population and gets a score $score_{i,j}$, where *n* is the number of opponents. The average of the score values, f_i, is a fitness value for the individual *i*. Fitness values are calculated this way for every individual. These fitness values are then used to drive a simulated evolutionary process as described in the pseudo code of Figure 2.6.

1. Randomly initialize Population1 and Population2
2. Evaluate each individual of Population1 with Population2 to determine the values $score_{i,j}$ as in Equation (2.1)
3. Store each evaluated score in a matrix format as shown in Table 2.4
4. Calculate fitness values of Population1‟s individuals by averaging the column values. Likewise average row values from the matrix to calculate fitness values of individuals from Population2
5. Copy the fittest individual in each population (the *elite* individual) into the next generation of each population
6. Do until a new generation of Population1 is completed:
   (a) Select two parents from the old Population1 according to a selection function based on fitness
   (b) Perform crossover according to the crossover operator and apply mutation to obtain two new offspring
   (c) Add the new offspring to the new generation
7. Repeat step 6 for Population2
8. Repeat steps 2-7 for the required number of generations. The solution is the final two populations.

Figure 2.6: Pseudo code of a CEA

When evaluating individuals, each individual needs to be evaluated against individuals in the competing population. In each evaluation, the individual receives a score which is added into a table as shown in Table 2.4. The purpose of creating a table is to avoid the re-evaluation of the individuals while calculating the opponent‟s fitness value. Note that this is only the case where the fitness calculation is symmetric, such that $score_{i,j} = score_{j,i}$, otherwise a separate table is needed for evaluating individuals of population2 against population1. Table 2.4 illustrates how each pair of individual interaction scores is listed and produces a fitness value by averaging those scores horizontally or

vertically. The averages of column or row values represent the fitness of the members of a population and its opponents respectively.

Table 2.4: Matrix format showing a fitness calculation method

10	23	72	66	76	49.4	Row Average (Fitness of Populatin2)
55	31	30	46	32	38.8	
67	63	95	84	34	68.6	
32	56	73	34	55	50.0	
84	62	21	67	84	63.6	
49.6	47.0	58.2	59.4	56.2		
Column Average (Fitness of Population1)						

(Source: DeJong (2004))

After calculating individuals‟ fitness values the selection process detailed in the following section is then employed.

- **Selection Procedure**

In CEAs as in the general EAs, the individuals are first sorted according to their fitness value f_i. Elite individuals directly copied to the new generation without further processing, as shown in step 4 of Figure 2.6. Subsequently, the selection procedure is applied to select a pair of parents for generating offspring. A selection procedure is one of the important processes which determine the selection of individuals that will be parents to produce offspring for the next generation. Parents are selected according to their fitness values: the fitter individuals have more chances to be selected. Examples of existing selection procedures include Tournament Selection and Roulette Wheel Selection (Singh & Deb, 2006). For this thesis, stochastic universal sampling (Baker, 1987) is chosen as a selection operator and is depicted in Figure 2.7.

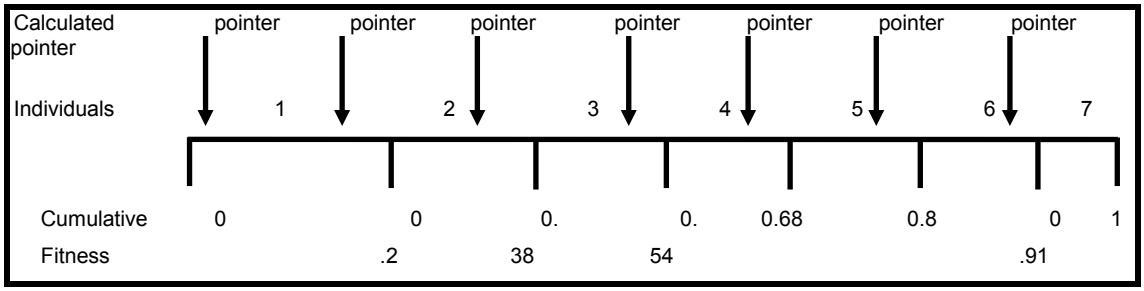

Figure 2.7: Stochastic universal selection with the range of bins (Source: Pohlheim, 2006)

Each individual is assigned a bin, with the width of the bin corresponding to the individual‟s fitness value. A cumulative fitness is calculated across the bins and is used as a selection index into the set of bins. The steps of stochastic universal sampling are listed below:

1. A common difference *d* is calculated by dividing the value for the total cumulative sum (in Figure 2.7) of fitness by population size. In the figure if the population size is 7 the value of *d* will be $d = \frac{1}{7} = 0.14$
2. A random number *u* between 0 and the last value of the cumulative sum is generated. In the above figure a random number in the range of 0 to 1 will be generated. For this example assume the randomly generated value is 0.75.
3. This generated number, 0.75 (generated in step 2), is used as a pointer to choose the individual whose bin this pointer falls. In this case 0.75 lies in the bin of individual 5; thus, the fifth individual is selected.
4. The common difference, d, is added to the pointer index to choose the next individual. The second selection will be in the position of (0.75+0.14=0.89) which is in the sixth bin. Likewise, the third selection will be (0.89+0.14=1.03) which is beyond the range of 0 to 1. Therefore, the maximum value will be subtracted from the sum value such as (1.03 - 1 = 0.03), 0.03 falls in between 0 and 0.2 which is on the first bin. Again the common ratio will be added (0.03+0.14=0.17) which is still in the first bin. Likewise the process continues until the stipulated number of individuals for the next generation has been selected.

Since individuals with higher fitness have wider bins, they are more likely to be selected. In the above example, the individual in the first bin is selected twice whereas

the fourth and seventh individuals are never selected. Once the parents are selected, they go through the reproduction process using the crossover and mutation operators. These are described next.

- **Crossover Operator**

Crossover is a process in which two or more parents exchange their genes in order to introduce diversity into their offspring. Like selection procedures, there are a number of existing crossover operators. In this thesis, a single point crossover operator (Poli & Langdon, 1997) is chosen.

A single point crossover executes according to the predefined crossover probability. The system generates a random number between 0 and 1. Crossover occurs only if the generated number is within the provided crossover probability (for example, if 80% was a pre-defined crossover probability, the generated value was 0 to 0.8 but not greater than 0.8). When the individuals are chosen for the crossover, a second process begins with another random number generated to determine for the crossover point. The crossover point determines the position within the parents‟ chromosome at which to cut and swap the genes to create a pair of new offspring.

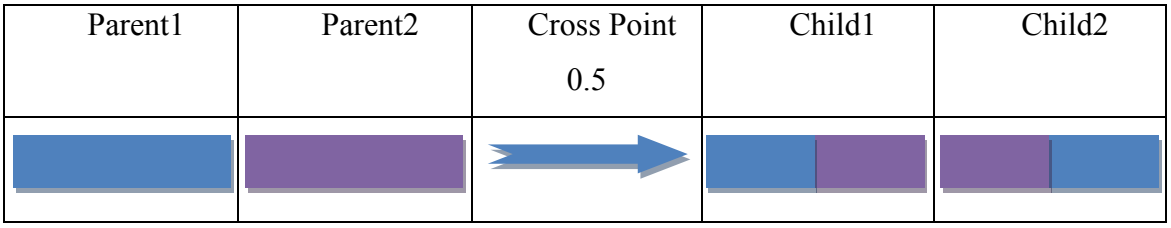

Figure 2.8: Chromosomes before and after the single point crossover

For this example, assume that the randomly generated crossover point is 0.5, which indicates that two parents will equally share their chromosomes to generate new children. Figure 2.8 depicts the two parents at the left side, with distinct chromosome structures (blue and purple). When the crossover is applied at the 0.5 point the new children, given at the right side of the figure, contain the first half of their chromosome from parent and the second half from the other.

- **Mutation Operator**

Like selection and crossover, mutation is also a basic process of the CEA which is applied to new individuals immediately after the crossover. Like selection and crossover, there are many existing mutation operators. For this thesis, two mutation operators have been selected for different problems. In both of these mutation operators, the process begins with the generation of a random number. A particular gene in a chromosome changes when the generated number is less than the pre-defined *mutation rate*, otherwise the specific gene remains unchanged. The process of generating random numbers and changing the genes continues until every gene is considered in the chromosome. The amount of change that occurs in the gene value is determined by the type of mutation operator selected.

There are various mutation algorithms developed including Polynomial and Gaussian. These two different types of mutation algorithms were used in this thesis. These two mutation algorithms will now be discussed.

- **Polynomial Mutation**

Polynomial mutation is a technique introduced by Deb and Goyal (1996) whereby a gene is mutated by adding the product of a noise value, the maximum value of change permissible, as depicted in Equations (2.2) and (2.3). The maximum value permissible to change is calculated by subtracting the maximum (τ^{max})and minimum (τ^{min}) gene value. For example if the gene values range from 10 to 100, the maximum change that can be applied is 90 (100-10). Equation (2.2) shows the calculation of mutated value γ^i by adding the product of the noise value λ and the maximum changeable value (τ^{max}-τ^{min}) to the un-mutated value ρ^i. Equation (2.3) depicts the calculation of the noise value λ which begins with generating a random value r_i between 0 and 1. The equation uses a constant, η_m, which represents a distribution index. Rubner et al. (2000) have suggested that 10 is a suitable value to be used for η_m.

$$\gamma^i = \rho^i + \lambda(\tau^{max} - \tau^{min}) \quad \textbf{(2.2)}$$

$$\lambda = \begin{cases} (2r_i)^{\frac{1}{(\eta_m+1)}} - 1, & If\ r_i < 0.5 \\ 1 - [2(1-r_i)]^{\frac{1}{(\eta_m+1)}}, & If r_i \geq 0.5 \end{cases} \quad \textbf{(2.3)}$$

- **Gaussian Mutation**

This mutation operator adds a Gaussian distributed random value to a chosen gene value as depicted in Equation (2.4) (Higashi & Iba, 2003; Hinterding, Michalewicz, & Peachey, 1996; Ursem, 2002). The user can specify the upper and lower bounds of the gene values. If a new gene value falls outside of the user specified lower or upper bounds for the gene, the new gene value is clipped to fit within the range of the lower and upper bound.

$$\gamma^i = \rho^i + \sigma \times G \quad \textbf{(2.4)}$$

The mutated value γ^i is calculated by summing the non-mutated gene value ρ^i with the product of standard deviation for the Gaussian noise (σ) and Gaussian noise (G). The standard deviation σ is set to 0.1 as per Hinterding, Michalewicz and Peachey (1996).

After applying the mutation operator to the individuals, a new population is created. The process of evaluation, selection, crossover and mutation repeats until the required number of generations. These are the components of the naïve CEA, which has long been used in optimization tasks. Despite the successful utilization of the coevolution concept in evolutionary computing (EC) by many authors including, for example Axelrod (1987) and Hillis (1990), Garcia-Pedrajas, Castillo and Ortiz-Boyer (2009), Kouchmeshky, Quino, Bongard and Lipson (2007) and Choo, Chua, Low and Ong (2009), as a general EC problem-solving technique, there are certain situations in which naïve CEAs are less effective. These are discussed next.

2.2.4.2 Pathologies of CEAs

Naïve CEAs has been known to demonstrate various pathologies, such as *disengagement*, *cyclic behaviours*, and *forgetting* when employed in different problem domains (Wiegand, 2003).

- **Disengagement**

Disengagement or *loss of gradient* occurs when one population no longer seems to impacts on the other, indicating that there is no relative fitness assessment between the individuals of the populations (this situation is known as evolutionary drift). This typically occurs when it is much easier for one population to achieve a superior optimization than another competing population. *Disengagement* can be exemplified by an untrained naïve soccer team playing a match with a professional well known team. No matter how much effort the naïve soccer team players put in, it can never win over the expert team. Even though same individuals in the naïve team will be better than others, subjective fitness against the professional team will rank them all equally badly, impacting the naïve team‟s evolution as poor individual as likely to be picked for the next generation as stronger individuals. The strategies that the expert team is equipped with are almost beyond the reach for the naïve team to achieve.

- **Cyclic Behaviours**

According to DeJong et al. (2007) and Wiegand (2003), cyclic behaviours are major problems that occurs in CEA applications. There are a number of causes for coevolutionary cycling. Often, it can be a direct consequence of relative fitness since individuals in the population(s) are rewarded on the basis of out-performing their opponents. This characteristic may result in earlier evolved “fitter” solutions being lost after many subsequent generations and leads to cycling owing to a loss of evolutionary momentum.

An intransitive superiority relationship is another cause that can also trap a system in a repetitive cycle. The intransitivity problem often occurs in CEA domains (Ficici, 2004) including in the RT domain. Intransitivity can be exemplified by an example: If a *solution A* is better than *solution B*, and *B* is better than *solution C* and yet *C* is better than *A* as in the example of the rock-paper-scissors game. In this game, there is no solution that is superior: paper beats rock and scissors beats paper but scissors does not beat rock, instead the rock beats scissors.

- **Forgetting**

Another major CEA pathology explained by Wiegand (2003), Ficici (2004) and DeJong et al. (2007) is the *forgetting* problem which occurs when the population concentrates only on the existing strategies from the other population and overlooks the ancestor populations‟ solutions. The fitness measure of a CEA is subjective depending on the individuals‟ interaction with another population. Individuals are evaluated against a set of opposing individuals from the same generation. On the basis of their performance new populations are evolved for the subsequent generation. These populations again are evaluated against the opposing population from the same generation to produce new populations for the subsequent generation. The selection procedure determines which individuals persist in the subsequent generations, on the basis of their fitness. This leads to certain individuals which were discovered during the coevolutionary search, later disappearing from the population.

2.2.4.3 Remedies of CEA Pathologies

In order to address cycling and disengagement pathologies some researchers, including Rosin and Belew (1997), DeJong et al. (2007) and Chong, Tino and Yao (2009), stressed increasing the diversity of the population. In addition, the use of an archive that remembers old solutions, when added to the basic algorithm, is believed to be effective in addressing the *forgetting* problem. Details of diversity maintenance techniques and archives are detailed in this section.

- **Diversity Maintenance Techniques**

CEA pathologies, namely *disengagement* and *cycling* may lead the search process to a premature convergence or no convergence at all (DeJong et al., 2007; Rosin & Belew, 1997). Since the fittest individuals are repeatedly selected, all individuals in subsequent generations may be very similar. Therefore, the objective of finding the optimal solution may suffer from premature convergence, where the population is driven to a single local optimum (or cycles between local optima) before the search space has been adequately explored. Additionally, disengagement can occur as there are not many diverse solutions available that can perform better than the existing solution. In order to address such premature convergence or lack of convergence owing to *disengagement* or *cycling*, various diversity maintenance techniques can be introduced to the naïve CEA.

Researchers, such as Eshelman and Schaffer (1993), Ryan (1994) and Rosca (1995) discovered that maintaining diversity in the population can address premature convergence as well as help in finding global optima, thus addressing the problem of disengagement and cycling. A diversity maintenance technique can preserve unique solutions and downgrades similar solutions. When there are a variety of solutions available in the search space, one specific solution would not obstruct the improvement of the process of finding the optimal solution. Chong et al. (2009) argued that diversity in a population can be maintained in two ways: explicit and implicit.

- **Explicit Diversity Maintenance: Varying Mutation Rates**

Maintaining diversity of the population may be carried out using an explicit method. Explicit diversity maintenance methods achieve diversity through variation. A simple method is to increase the *mutation rate*. The mutation operator changes the genome according to the *mutation rate* set. When a high *mutation rate* is applied it increases the chances of altering the individuals‟ genome (Ursem, 2002). A higher *mutation rate* forces higher diversity; however, a higher diversity caused by extreme mutation and hybridization (crossover) may not give a favourable outcome in terms of finding the optimal solution. These reproductive methods create new solutions by changing individuals‟ genomes which produce a genetically different individual i.e. increase diversity in the population. However, their performance may not be significantly different from their parents as they share common genes. Therefore, Wright (1986) stresses the balance between genetic homogeneity and heterogeneity, which also supports the argument that extreme diversity may not be always good.

- **Implicit Diversity Maintenance**

Whilst *mutation rate* explicitly affects the diversity, a population‟s diversity may also be maintained implicitly. Implicit diversity maintenance (IDM) methods achieve diversity by favouring more diverse solutions in the selection process. IDM can be categorised into three categories, namely the *kernel*, *nearest-neighbour* and *histogram* approaches (Talbi, 2009). The *kernel* method considers the distance between the involved solutions, which is a concept of *fitness sharing*. The *nearest-neighbour* maintains diversity on the basis of the distance of the individual from other individuals

within a specific distance. In the *histogram* approach, a number of partitions are created on the basis of individuals‟ gene values. Each individual is then evaluated and put into a specific partition. Thus, the number in each partition is counted and reward is provided to the individuals in the partition in which there are less number of individuals.

A typical method of maintaining diversity implicitly is by using *fitness sharing* (FS) (Goldberg & Richardson, 1987), a method belonging to the category of *kernel* approaches (Talbi, 2009). The FS approach was introduced by Holland (1975) and later extended by Goldberg and Richardson (1987). According to Goldberg and Richardson (1987), in FS, diversity is maintained in the population by discouraging individuals with similar characteristics. In this process, the fitness values are reduced for those individuals within the population who have similar gene structures. Subsequently, individuals with unique gene structures get higher fitness values, which then diversify the population.

Goldberg and Richardson (1987)‟s FS is implemented by many authors in various domains (Beasley, et al., 1993; Cioppa, Stefano, & Marcelli, 2004; Mckay, 2000; Rosin & Belew, 1997; Sareni & Krahenbuhl, 1998). According to Goldberg and Richardson (1987) as depicted in Equation (2.5), the shared fitness $f_i^{''}$ (fitness value after the implementation of the FS) of an individual i is calculated by dividing the raw fitness f_i of an individual i (calculated in Equation (2.5)) by the *niche count* c_i.

$$f_i^{'} = \frac{f_i}{c_i} \tag{2.5}$$

Niche count is calculated on the basis of the individual‟s gene structure variation $d_{i,j}$ in the population. The $d_{i,j}$ is the Euclidean distance between the chromosomes of two individuals, i and j, in the population. Equations (2.6) and (2.7) are used to calculate niche count c_i and gene variations $d_{i,j}$ respectively.

$$c_i = \sum_{j=1}^{n} \begin{cases} 1 - \left(\frac{d_{i,j}}{n_r}\right)^{\tau}, & if\ d_{i,j} \le n_r \\ 0, & Otherwise \end{cases} \tag{2.6}$$

$$d_{i,j} = \sqrt{\sum_{m=1}^{u} (x_{i,m} - y_{j,m})^2} \quad \textbf{(2.7)}$$

In Equation (2.6) τ is a constant parameter that determines the shape of the *sharing function*, n_r is a constant niche radius and n is the population size. In Equation (2.7), u represents the length of genome, $x_{i,m}$, represents the m^{th} gene value of the i^{th} individual, $y_{j,m}$ is the m^{th} gene value of the other individual j.

Researchers employed FS in various ways. Smith, Forrest and Perelson (1993) introduced some variations on the use of FS, known as emergent FS where FS was used in a single population EA. In this FS, the fitness of an individual is calculated in three steps.

Step 1: An individual is randomly chosen from a population.

Step 2: Secondly, a set of individuals (N number) from the same population is chosen randomly.

Step 3: The distance is measured between the individual selected in step 1 and each of the N individuals selected in step 2. Each of the N individuals receives a certain score according to their dissimilarity with the individual from step 1.

The above mentioned three steps iterate up to the required times, mostly the repetition is equal to the number of population size. Finally, individuals with high scores get fitness preferences.

Another FS method introduced by Rosin and Belew (1997) is known as competitive FS. This approach was developed for "two population competitive CEAs" and tested on a 3D tic-tac-toe game. Normally, FS rewards genetically unique solutions; however, competitive FS emphasizes those individuals which may not be genetically unique but can defeat a strong opponent (a strong opponent is one which can defeat many individuals). Therefore, for any individual, it is not how many opposing individuals it defeats, but what type of opposing individuals it defeats that is important to achieve higher preferences in competitive FS. In this FS method every individual has its own

function which counts how many opponents it defeats. The record of its strength (number of individuals it defeats) will then be used when fitness is calculated.

- **Archives**

Another way of addressing CEAs‟ pathologies is the use of archives in the form of a memory mechanism. In CEAs a fit individual (on the basis of the fitness value) has a higher probability of being selected for the next generation. If any individual receives higher fitness than the fittest individual from the previous generation, then the previously fittest individual may never be selected for subsequent generations. Thus, in order to be selected in every generation, the individual has to achieve a high fitness value all the time. If the fit individual, which remained good for many generations, is not selected just once, the same individual (which could be a prominent solution) will be lost. Therefore, in order to preserve such optimal individuals, a memory mechanism is used. The concept can be defined as an extension of elitism in which the best individuals are copied to the memory in every generation. The stored optimal individuals are then used in fitness evaluation in later generations.

In memory mechanism, a number of the fittest individuals are accumulated in memory over many generations. While collecting the individuals, the same solution may be reselected for the memory in different generations. Subsequently, when individuals from one population are evaluated with another population, they are also evaluated with individuals from the memory (individuals in the memory are a collection of the best performers on the basis of the fitness). This extra evaluation of the individuals with memory individuals helps prevent the CEA‟s *forgetting* pathology as the evaluation with memory individuals reminds the evaluating individual about the old counterpart. The evaluation of individuals with the individuals from memory, promotes only those individuals that can perform well over the memory individuals. Researchers introduced various memory mechanisms such as the *Hall of Fame* (Rosin & Belew, 1997), *Shared Memory* (Puppala, Sen, & Gordin, 1998), *Nash Memory* (Ficici & Pollack, 2003) and *History Bank* (Avery, et al., 2008).

A memory mechanism called the Hall of Fame (HOF) which was introduced by Rosin and Belew (1997), is used as an archive in this study, due to its wide use. The HOF has been proven to be a technique that addresses CEAs‟ *forgetting* pathology (DeJong, et

al., 2007; Ficici, 2004; Nolfi & Floreano, 1998; Watson & Pollack, 2001; Wiegand, 2003). The HOF is a technique that allows the population to interact with a set of the best individuals from the previous generations of the opponent population. The best individuals are the ones that receive higher fitness values. Generally, the best individuals from both the populations in every generation are accumulated in an archive, which interacts with the populations during the fitness evaluation. The purpose of the HOF is to preserve the best individuals from different generations so that in every generation when populations are evaluated they not only evaluated against the other population but also evaluated against the individuals in the HOF memory. This additional evaluation reminds the new populations of their previous problems and challenges which help to avoid the CEA pathologies. When the HOF is used, fitness f_i of individual, *i,* is calculated by using the Equation (2.8).

$$f_i = \frac{1}{n+m} \sum_{j=0}^{n+m} score_{i,j} \tag{2.8}$$

The score function is the same as was given in Equation (2.1). The symbol *n* is the population size of the other population; *m* is the size of the HOF archive.

Despite the usefulness of the HOF, some researchers including Nolfi and Floreano (1998) drew attention to the fact that the size of HOF may negate the usefulness of this approach. As a fittest individual is added into the HOF in every generation, there will be many individuals in the HOF as the number of generations increases. As the size of HOF increases, when individuals are evaluated, they are evaluated less with the evolving but more with the non-evolving counterparts from the previous generations. The process of evaluating individuals, in which individuals are evaluated against the competing population and also against the individuals from previous generation of opposing population, was found to be better than the HOF for their predator and prey robots.

In the *Nash memory* approach (Ficici & Pollack, 2003), the best strategies are collected from each population. This memory mechanism consists of two memory spaces. The capacity of the first memory bank is relatively larger in which the number of stored

strategies increases monotonically as the search progresses. The second memory bank; however, has a limited capacity and stores only selective past strategies. The selected best strategy competes against individuals in the first memory and decides whether the second memory bank requires updating. This approach has been tested using the intransitive number problem which was introduced by Watson and Pollack (2001). The authors have shown that this technique was effective in enhancing the algorithms‟ performance in solving the intransitive number problem.

The *history bank* (Avery, et al., 2008) was utilized in the Tempo military planning game. In this technique, the best strategies from both the populations are stored in the same memory space rather than creating separate memory space for each population. Thus, the size of the *history bank* grows continuously as two individuals are added in each generation. In every generation, each individual is evaluated against the other population and also with randomly chosen individuals from the *history bank*. The authors have identified that this approach was beneficial in enhancing the algorithms‟ performance in solving a problem called the Tempo problem.

The above discussion presented the various CEA components. With regard to RT, many researchers including Upton and McDonald (2003), Hingston (2011) and Choo et al. (2007), used techniques in which simulated scenarios are combined with EAs to identify appropriate strategies that best suit the scenario. The following section reviews the existing optimization methods that used various EAs to optimize simulated scenarios.

2.3 RT Optimization

In order to determine optimal combat strategies, researchers have used and developed optimization tools based on various search techniques using the various ABDs presented in section 2.1.2.2 as simulation environments. In computer-based RT, a scenario can be created which can be run many times to gather data. Subsequently, search-based optimizations and machine learning methods can be applied on those gathered data to identify the best solution sets. Military RT applications are optimized by optimizing the combatants‟ characteristics, helping to detect vulnerabilities.

Except for ISAAC, all simulators discussed in section 2.1.2.2 have in-built optimization techniques in which they utilized a GA to optimize the agents‟ characteristics. In addition to these optimization techniques, Upton and McDonald (2003), Choo, Chua and Tay (2007), have also utilized GAs in their RT optimization techniques. Some researchers including Choo, Chua, Low and Ong (2009), Hingston, Preuss and Spierling (2010); and Hingston and Preuss (2011) have introduced RT optimization techniques based on CEAs. Details of some optimization techniques follow.

2.3.1 Optimization Tools Integrated in Simulators

Ilanchiski (2003) introduced an optimization technique in computerized RT via EINSTein. It uses a GA to optimize the agent‟s characteristics in a specific scenario generated in EINSTein. Likewise, MANA also incorporated a GA, which acts to select an agent‟s personality, including the trigger state and behaviour of the agent in special conditions. Additionally, MANA also supports the optimization of both red and blue teams within a single run. However, each team evolves against one fixed opponent strategy; thus, despite both teams evolving simultaneously, it is not coevolution. The evolution of one team does not adapt to the changes brought about by the evolution of the other teams. The MANA GA tool includes the facilities of optimizing multi-squads simultaneously with the facilities of selecting different measures of effectiveness (MOE) and agent‟s personality.

2.3.2 Automated RT

Choo, Chua and Tay (2007) introduced a RT optimization tool called automated red teaming (ART). The ART system consists of some major components; parameter setting, ART Controller, Simulation model, EA models and Condor cluster and controller with ART output module. The functionality begins with the selection of the GA parameters such as crossover rate, *mutation rate* and population size. The entered parameters are received by the controller along with a scenario and sent to the GA to optimize.

ART has been used in optimizing some scenarios including the urban scenario (McIntosh, Galligan, Anderson, & Lauren, 2007), the coastline scenario (Chua, et al., 2008) and anchorage protection scenario (Han et al., 2007). The authors claimed that just by modifying the behavioural parameters, the red force survivability can be improved by 27% in the urban scenario. This information provides an opportunity for blue force to review their plans and strategy in order to reduce casualties and eventually to win the battle.

The authors also claimed that RT was compatible with any algorithms such as GA, MOEA and SPEA. However, in this study only GA has been used as an optimization algorithm. Since GA cannot optimize both teams in RT simultaneously, the optimization results are limited to respond to a single strategy. However, in reality there may be many strategies that an adversary team may incorporate. Additionally, the authors have not disclosed the situation when the opposing team is optimized against an already optimized team. The performance of the algorithm was evaluated by monitoring the fitness value. However, the commonly used technique, diversity maintenance, was not emphasized in any of these studies. Although the fitness is calculated by considering a number of measure of effectiveness (MOE), such as the probability of reaching the goal and the combatants‟ casualties, their individual improvements from the initial generation to the final generation was not discussed.

Decraene, Chandramohan, Malcolm and Choo (2010) argued that two factors, computing budget and constraint handling, should be considered while optimizing military-based RT scenarios. However, the authors only presented an example of ART to describe the computing budget; it is an issue for almost all RT optimization. The fact is that RT optimizations require a high performance computing system to meet the time constraints and experimental requirement. The constraint handling involved focussing on the trade-off in financial or practical cost. The authors stressed that in the optimization of scenarios involving only parameter values, it may not be possible to implement the emerged strategies in the “real-world” owing to their unrealistic nature or their actual implementation cost. When optimizing the red team; mostly, the movement speed value is recommend to be the maximum value in the given range. Red agents with

high speed increase the chances of breaking the blue security. Such features may be expensive to implement in reality.

Decraene et al. (2010) proposed cloud computing to address the issue of computing budget. Cloud computing is a technique to access software service via the internet which avoids the need for one"s own hardware. The authors found that this technique not only reduced the cost but also it solved the issue of scalability as many computer systems can be deployed for conducting a very large scale experiment. To address the issue of constraint handling, the authors proposed to modify preferences. While optimizing the red team, rather than focussing only on maximizing blue casualties and minimizing red casualties, the authors also considered a cost related factor. When the speed of movement increases the cost related factor also increase which reduces the fitness of an individual.

2.3.3 Automated Coevolution

Choo, Chua, Low and Ong (2009) introduced automated coevolution (ACE) that utilized a CEA for military operational analysis. The authors have depicted a comparative analysis of various algorithms including ant colony optimization (ACO), particle swarm optimization (PSO), strength Pareto evolutionary algorithm version 2 (SPEA 2) and elite Pareto genetic algorithm (EPGA). The authors have mentioned that the coevolution runs were executed with EPGA and PSO. The study discovered that the performance of EPGA was better than the PSO in terms of quality of solution. Furthermore, the authors illustrated the comparative study of various interaction process namely „all vs. best", „all vs. top5" and „all vs. all". The authors found that „all vs. all" would evolve more robust and less cyclic solutions; however, it is computationally more intensive and more robust solutions are not always the optimal solutions. Thus, the authors found that „all vs. best" could be the appropriate method to use for their specific scenario.

ACE was capable of evolving two populations simultaneously. They have evaluated ACE using an anchorage protection scenario, in which various evolved strategies for the blue and red team were depicted. However, evolving two teams simultaneously is still

not viable in their study as one team remains constant while the other evolves. In addition, the study analyses the algorithms‟ performance on the basis of the simulation run (attrition rate) - there was no other methods described that measure the performance of the algorithms in optimizing the scenario such as the improvement of fitness or generalisation performance. Based on the pseudo code presented by the authors, the study uses a test population against which the evolving population competes; however, how the test populations were created was also not clearly explained.

2.3.4 RedTNet

With respect to RT applications, a combination of simulators and search-based algorithms such as EAs have been extended by Hingston and Preuss (2011) who presented an optimization tool based on CEAs. The authors stressed the necessity of CEAs in optimizing RT applications due to the dynamic characteristics of the problem. They coevolve two involved teams simultaneously and also tested the developed tool on an intransitive number problem that was introduced by Watson and Pollack (2001) to demonstrate that it could address the problem of intransitivity. The tool was later applied in a RedTNet scenario that represents a critical infrastructure problem.

In most of the previously mentioned RT optimizing tools, GAs are used as a search optimization algorithm, in which the population evolves against a specific strategy from the competing team. However, the ACE and RedTNet tools are based on CEAs. ACE introduced the concept of coevolution in RT but the study contained some limitations. The study did not appear to address problems associated with CEA pathologies. Also, the study discussed only the evolved strategies but did not disclose how the performance of the algorithms was evaluated. In RedTNet, the authors coevolve strategies of both teams simultaneously; however, again, no evidence of integrating any techniques to address CEA pathologies was provided. In addition, performance measures of the algorithms were not included in the study.

In every optimization technique mentioned above, a specific scenario is required. This scenario is run many times and is evaluated with the help of a suitable algorithm. A

comparison of the characteristics of various optimization techniques is provided in Table 2.5.

Table 2.5: Comparative study of the optimization tools

Optimization tool	EINSTein	MANA	WISDOM	ART	ACE	RedTNet
Algorithm used	GA	GA	GA	GA	CEA	CEA
GUI	Yes	Yes	Yes	Yes	Yes	No
ABDs	EINSTein	MANA	WISDOM	MANA	MANA	RedTNet
Cluster facility	No	No	No	Condor	Condor	No

2.3.5 Limitations of Existing RT Optimization Methods

Some of the limitations highlighted below remain in existing RT optimization methods.

- The existing applications use simplistic AI techniques such as single objective GAs which do not address all the expectations of highly non-linear military operations. Although some optimization tools use CEA, more investigation is required into enhancing the algorithms‟ performances in finding good solution sets.
- There is a lack of suitable performance measures to evaluate the algorithms‟ performance.
- Existing studies are based only on objective measures. A systematic study of factors affecting the algorithms has not yet been conducted.
- When optimizing RT scenarios using CEAs, the pathologies associated with the algorithms are not sufficiently addressed.
- RT domain may have intransitivity, which is associated with CEAs‟ cycling pathology. There was no literature that investigates this issue.
- Additionally, the expectation was that there may be more than one good solution to address an opponent‟s strategy in RT scenarios. This introduces a concept of multimodality. No literature was found that investigates multimodality in RT applications.

2.4 Conclusion

This chapter described three major areas associated with this study, a review of RT applications, optimization algorithms (EAs and CEAs) and optimization techniques applied to the RT domain. The review highlighted that despite availability of some optimization techniques that help to detect vulnerabilities in combat plans, limitations remain in every technique. No literature has been found that mention RT issues such as intransitivity and multimodality. In order to investigate the limitations and issues associated with RT applications, a pilot study was conducted and is presented in the following chapter

3 A Pilot Study: Red Teaming Optimization using GAs

Chapter 2 reviewed the literature that is associated with this thesis. This chapter reports on a pilot study in which a GA was used to optimise one team‟s strategy in a RT scenario. The coastline protection scenario used in this study was developed in the Defence Science Organisation (DSO) in Singapore (Chua, et al., 2008). The scenario was built in MANA and has been employed by other researchers including Chua et al. (2008) in investigations using various techniques including data farming and EAs. Scenarios created in MANA have many characteristics including the agent‟s personality, situational awareness and trigger states (see section 2.1.2.2). These characteristics can be used as optimization parameters for realistic outcomes. Additionally, these scenarios can be run multiple times by being integrated with the optimization tools. For the pilot study, GA was chosen as an optimization algorithm. The reason for choosing this algorithm was that it has been commonly used in existing optimization tools, including MANA and ART. GA is also suitable when objective measures can be defined for calculating the fitness of individuals.

The first section of this chapter is a description of the optimization tool (OT) which was developed to optimize RT scenarios and other applications in this study. The second section presents a description of the incorporated DSO scenario. The third section describes the chosen parameters for optimizing the scenario. The fourth section shows the results of optimizing the teams involved in the DSO scenario.

3.1 Description of the Optimization Tool

The OT is an automated tool that was developed to assist with RT and other similar applications in finding optimal strategies. In the case of RT, the OT assists to find vulnerabilities in a security plan. The architecture of the tool is depicted in Figure 3.1 and its various components are described in the following sub-sections.

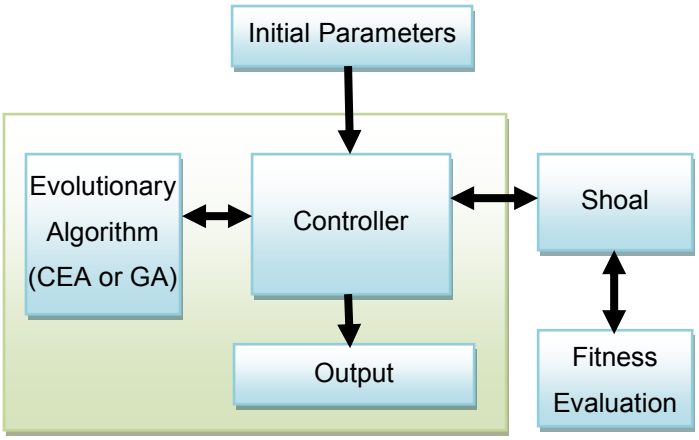

Figure 3.1: Framework showing various sections of the OT tool

3.1.1 Initial Parameters

The OT requires some essential parameters to begin the optimization of a scenario. The parameters needed to set up the optimization task are supplied via an input file. Table 3.1 depicts the common parameters used for the empirical tests in this study.

Table 3.1: Parameters Considered for Experimentation

Parameters	Meaning
Population size	The size of the population
Simulation run	Determines how many times the simulation runs for evaluating each individual
Number of generations	Determines how many generations the populations evolve.
Mutation rate	Probability of executing the mutation operator
Crossover probability	Probability of executing the crossover operator
Type of algorithm	Whether it is GA or CEA
Path for a test file	For example, "c:\File\test.csv"
Path for a scenario file	For example, "c:\Scenario\coastline.xml"
Number of runs	Determines the number of times the program iterates
The following additional components should be defined when using CEAs	
Variants	Variant options: None, FS, HOF or both
Niche radius for FS	Value that determines the extent of diversity
HOF size	Determines how big the HOF space would be
Shoal	Requires a Boolean expression to indicate whether Shoal should be in use

3.1.2 Controller

This component manages the overall optimization process as it deals with receiving the inputs and producing the output. According to the given parameters, an initial population for each team is created. The controller also prepares for the use of either GA or CEA and prepares Shoal, a distributed computing library, if the user specified its use for calculating the fitness of the individuals. This component creates a HOF if users have stipulated this option via the input file. The controller ultimately calculates the performance of the evolving populations in each generation.

3.1.3 Genetic or Coevolutionary Algorithm?

Depending on the user's selections, the OT utilizes GAs or CEAs to optimize various strategies for problem domains. GAs optimize a single population at a time to find a high quality solution to a problem, whereas CEAs are capable of optimizing multiple populations simultaneously. Details of these algorithms are described in section 2.2.4. These algorithms require parameters such as the number of generations, crossover and mutation rates, which they get from the controller. After a specified number of generations, they produce optimized population(s), depending on the algorithm provided as input to the controller.

3.1.4 Output

The output of the OT is an optimized population(s) which contains a list of potential solutions. Since GA optimizes a single population, the output also contains a single population. However, CEA produces two optimized populations. The final populations produced from CEA contain optimal solutions from the optimization process for counteracting the competing population. In the case of RT, the output is a list of agents‟ characteristics, including the ways in which they move and the targets they select. The generated parameters can be supplied to the scenario file, which can be used in the simulation (MANA) to verify the validity of the evolved strategy.

3.1.5 Shoal

Shoal is an open source Java library that can be integrated in distributed applications for load balancing and/or fault tolerance (Danculovic, 2007). Shoal was used to run one or multiple instances of the simulator, MANA. In the network, Shoal performs various tasks including data sharing, communication and notification of relevant messages such as the connection, disconnection, and failure of a node(s). Each node has its own unique identity called a *Global Unique ID* which prevents the conflict of the nodes. A single network can host many nodes; however, each node consumes additional bandwidth and processing resources. Group management service (GMS) is another important concept in Shoal. The GMS manages the nodes, mediates and facilitates nodes communication (Danculovic, 2007).

Shoal does not impose any structure on the cluster; however, according to developers‟ design, Shoal may have a master component or a slave component. Then, the application programming interface (API) sends signals to specific nodes to perform tasks allocated to the components. Figure 3.2 shows three machines interacting in Shoal. Each application instance loaded one Shoal‟s GMS and all are joined in a group to communicate. Machines 1 and 2 are each running a Java virtual machine (JVM) and executing a single instance of the application whereas Machine3 is running two JVMs running two different instances.

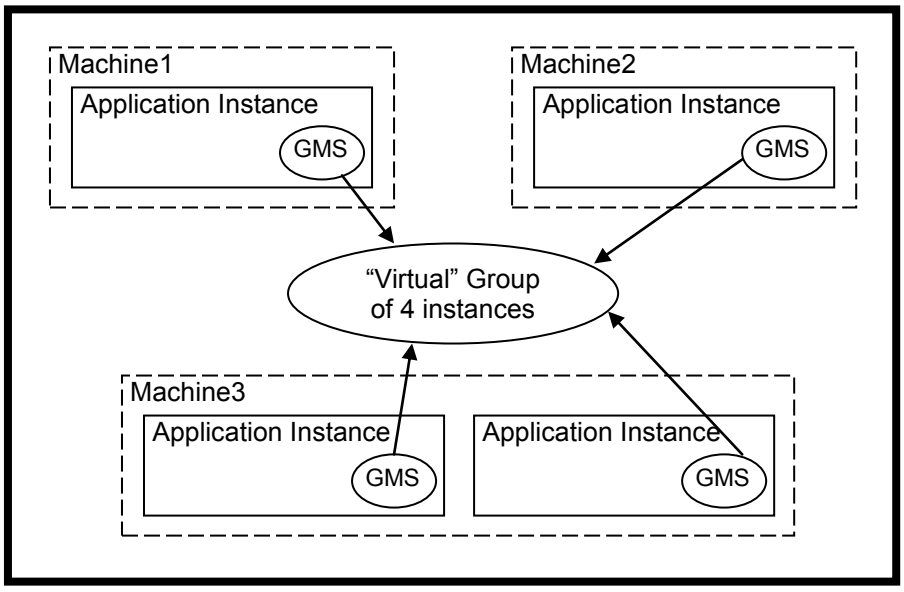

Figure 3.2: A Shoal system‟s architecture (Source: Danculovic, 2007)

Note that Shoal is not essential for problem domains like the intransitive number and the multimodal problem, since computational times for their fitness evaluations are minimal. Even in the case of the RT application, it does not take long to run a single simulation. When the experiments involved applications of GA, it may not be necessary to use Shoal.

When the experiments involved CEA being employed on a RT scenario, the scenario needs to be executed many times via MANA, as individuals from one population must be evaluated against all individuals from the other population. This number of simulations increases the computational time. In such a situation, Shoal becomes necessary to reduce the processing time. When the population‟s fitness is evaluated, a master machine distributes the simulations to other workstations to reduce its workload. The master machine is responsible for the conduct of the entire process from creating genomes, individuals and populations to finally displaying the output. Other workstations are slaves that perform only the tasks allocated by the master machine for the purpose of evaluating individuals according to the fitness functions provided. The slaves return the results of each simulation to the master once they have been executed.

3.1.6 Fitness Evaluation

Fitness evaluation is an important step in the evolutionary process as it determines the fitness of individuals on the basis of defined fitness functions. A fitness function may be a simple equation, e.g. a square of a value, a complex equation such as those described for an intransitive number or multimodal problem (as in chapters 5 and 6 respectively) or it may be calculated from the measures of effectiveness (MOE) received by running a scenario in a simulator in the RT.

In the RT application, MANA is utilized as a simulator that runs scenarios in order to evaluate strategies. In MANA, a scenario file describes the particular environment, which is written in extensible mark-up language (XML). The scenario file also describes at least two military squads with different intentions and targets. When the scenario runs in MANA, it uses the strategies incorporated by both the teams in that particular environment.

3.2 Scenario Description

As a preliminary study the OT tool was used to optimize a RT scenario using the GA. For this study, the coastline Key Installation (KIN) protection scenario which was developed using MANA at the Defence Science Organisation (DSO) National Laboratories was used. The aim of using this scenario was to replicate Chua et al. (2008)‟s investigation and to provide a baseline for comparison with subsequent experiments involving coevolution.

The scenario contains red boats, blue boats and key installations. The key installations are immobile infrastructures placed at the coastline and the red boats are penetrators that try to breach the blue boats‟ surveillance. The blue boats maintain surveillance at a slow speed with lethal weapons, whereas the red boats are unarmed penetrators. Their aim is to breach the blue surveillance and reach the coastline (Chua, et al., 2008).

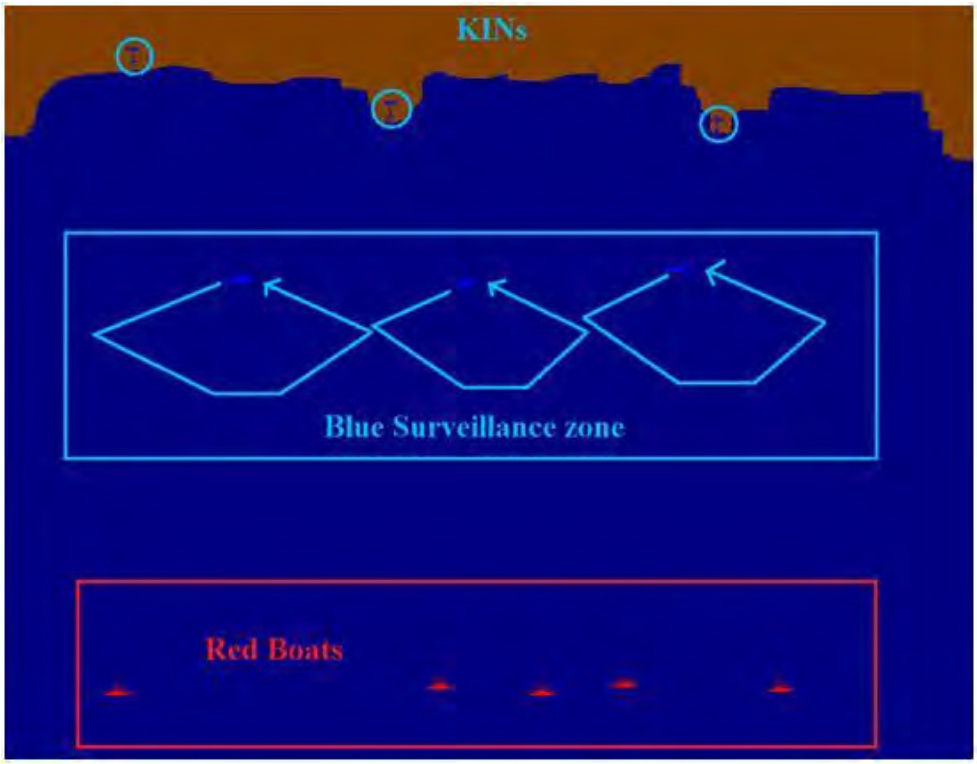

Figure 3.3: Scenario for Key Installation protection

The original scenario consists of three KINs, three blue patrolling boats and five red boats. Each blue boat constantly moves in a specific route to resist any penetrator. Figure 3.3 depicts one possible set of blue surveillance routes and KINs along with the initial positions of the red boats. Each blue boat follows its specified route unless one of the red boats comes into its contact area. When any blue boat finds a red boat within its sensor range, it moves towards the red boat and attacks it. The parameters considered for this experiment are discussed in the following section.

3.3 Experimental Setup

This section presents the parameters used in the pilot study. Combatants‟ personalities listed in Table 3.2 are used as genomes for the optimization process. The table also depicts the chosen personalities and their upper and lower bound values. The GA parameter values are listed in Table 3.3.

In order to explore the strategies available to each side in this scenario, the number of attacking boats (i.e. red boats) is chosen to be a number in the range between two to five boats. Every scenario has the same number of blue agents, with identical patrolling strategy and mission. In the first variation of the scenario, two red boats try to penetrate against the three blue patrolling boats. Subsequently, the second, third and fourth scenarios have three, four and five red boats respectively.

The two factors considered as the red boats‟ measures of effectiveness (MOEs) are: (1) maximizing the goal achievement (that is, breaking the blue boat patrolling tactics by getting at least one boat to the coastline) and (2) minimizing the red casualties. These two factors are combined to define the fitness function to guide the selection in GA. The fitness is calculated for each individual set of personality parameters from 20 simulations using Equation (3.1). In every fitness analysis, MANA simulations are run for a specified number of times and a mean value for four factors: blue team casualty, red team casualty, blue team achieved the goal and red team achieved the goal is obtained. A value for casualty depends on the number of boats involved in the scenario whereas a value for achieving the goal by boats for each team is set to 1 for success and 0 for failure to achieve the goal.

Table 3.2: Selected agent personality parameters

Characteristics Considered	Description	Values in Range
Movement Speed	The value determines the number of cells agents move in a given time step. Its range is 0 to 1000; however it is normalized to 100 so that an agent can move one cell per time step.	0 to 100
Agent Situational Awareness (SA) indicates that agents take actions on the basis of the information available from its own sensors. Negative and positive value indicates repulsion and attraction respectively.		
Enemy	Attraction or repulsion with the agent with enemy allegiance	-100 to 100
Enemy Threat 3	Attraction or repulsion with the agent with enemy allegiance Threat Level 3	-100 to 100
Uninjured Friends	Attraction or repulsion with the agent with same allegiance	-100 to 100
Cover	Determine the distance of shooting by direct fire weapons in the terrain.	-100 to 100
Concealment	Determine the visibility of agents in the terrain.	-100 to 100
Squad SA indicates that agents take actions on the basis of the information available on the squad's SA map. Negative and positive value indicates repulsion and attraction respectively.		
Enemy Threat 3	Attraction or repulsion with the agent with enemy allegiance Threat Level 3	-100 to 100
Friends	Attraction or repulsion with agents of the same squad	-100 to 100

Table 3.3: GA parameter values

Properties	Values
Agent-based simulation	MANA
EA variant	GA
Simulations per individual	20
Population size	20
Generations	50
Crossover rate	60%
Mutation rate	1/gene length
Number of experiments	20

$$Fitness = Red\ Goal\ Success\ Proportion \times (Number\ of\ red\ agents)^2 - Mean\ red\ casualties + Number\ of\ red\ agents \quad \textbf{(3.1)}$$

In MANA, the simulation termination condition was set to 1000 simulation steps, or all red agents destroyed, or any red agent achieving the goal of reaching the coastline. With the above mentioned agents‟ personalities and GA parameters, the OT was used to conduct the experiment. The results derived are discussed in the following section.

3.4 Results and Analysis

This section presents the outcome of optimizing the coastline scenario in two sub-sections. The first sub-section is a discussion of the experiment when different numbers of the red boats are evolved against the fixed blue boats. The second sub-section is a discussion of the empirical study when the blue boats are evolved against the already optimized red boats.

3.4.1 Evolving Red Boats

The results presented in this section are for the red boats‟ optimization against a fixed blue surveillance strategy. Figure 3.4, Figure 3.5, Figure 3.6 and Figure 3.7 depict convergence plots showing the fitness values versus generations in scenarios with the red agents ranging from two to five respectively. Each graph shows median values of best fitness values for each generation over 20 repeats of the GA along with the range of fitness values found in the population. In this particular experiment, it was noticed that the median is more robust than the mean. The theoretical maximum fitness (with all 20 runs resulting in success and no red casualties) is also shown on the graph. The results demonstrate that there is a direct relationship between the number of penetrators involved in the battle and the likelihood of them achieving their goal.

In all convergence plots the search converges quickly, in less than 20 generations. When five attackers were used in a simulation, the GA reliably converges to a solution with a fitness value close to 30 (the theoretical maximum). However, when only two

attackers were used in a simulation, convergence is much less reliable, with a range of final fitness values between 5 and 6 much lower than the theoretical maximum of 10. This indicates that, as expected, with fewer attackers it is more difficult to find good solutions for the red team.

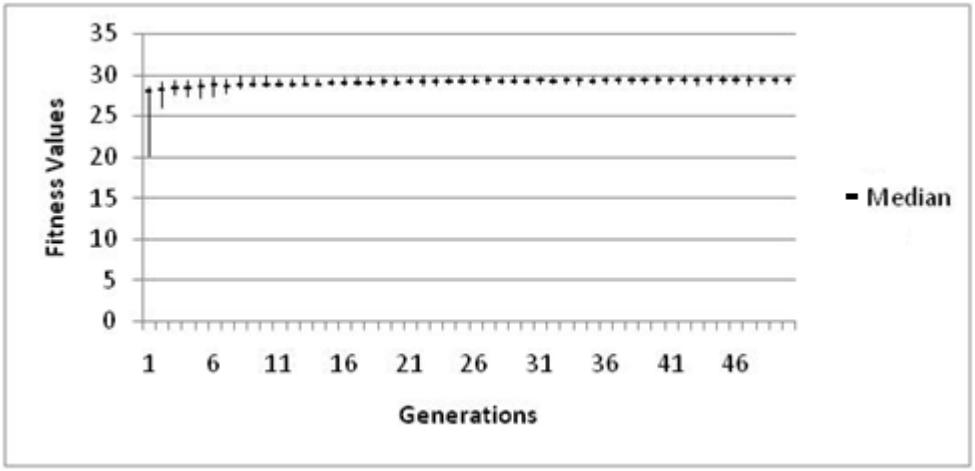

Figure 3.4: Median fitness values of the red team with five red boats.

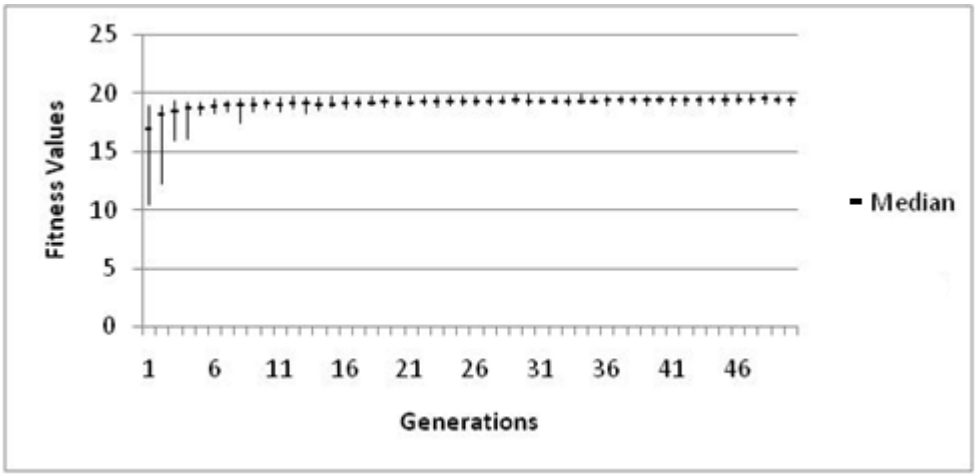

Figure 3.5: Median fitness values, along with an indication of the range of fitness values, of the red team with four red boats.

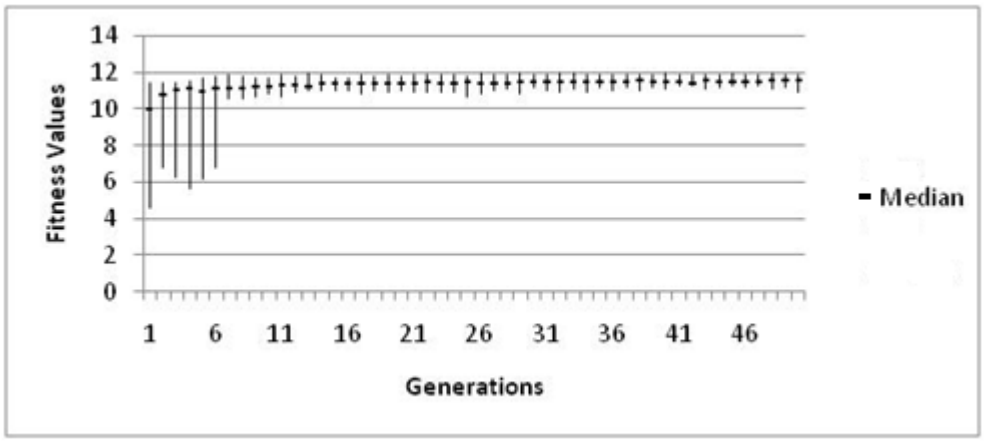

Figure 3.6. Median fitness values, along with an indication of the range of fitness values, of the red team with three red boats

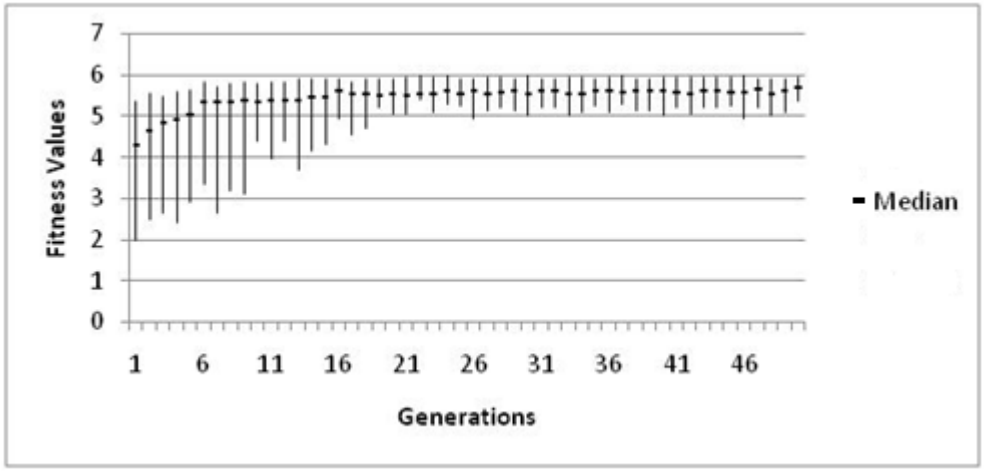

Figure 3.7. Median fitness values, along with an indication of the range of fitness values, of the red team with two red boats.

The series of experiments with different numbers of red boats show that the tactics used by the red boats vary according to the number of red boats involved. Figure 3.8, Figure 3.9, Figure 3.10 and Figure 3.11 depict the tactics used by the red team when different numbers of red boats are involved. Figure 3.8 shows that when there are only two red boats, they tend to avoid confrontation with the blue boats and use a flanking strategy to reach at the coastline. When there are three red boats, as in Figure 3.9, the tactic employed is to use one boat as a distraction so that other two can easily pass through the blue patrol formation.

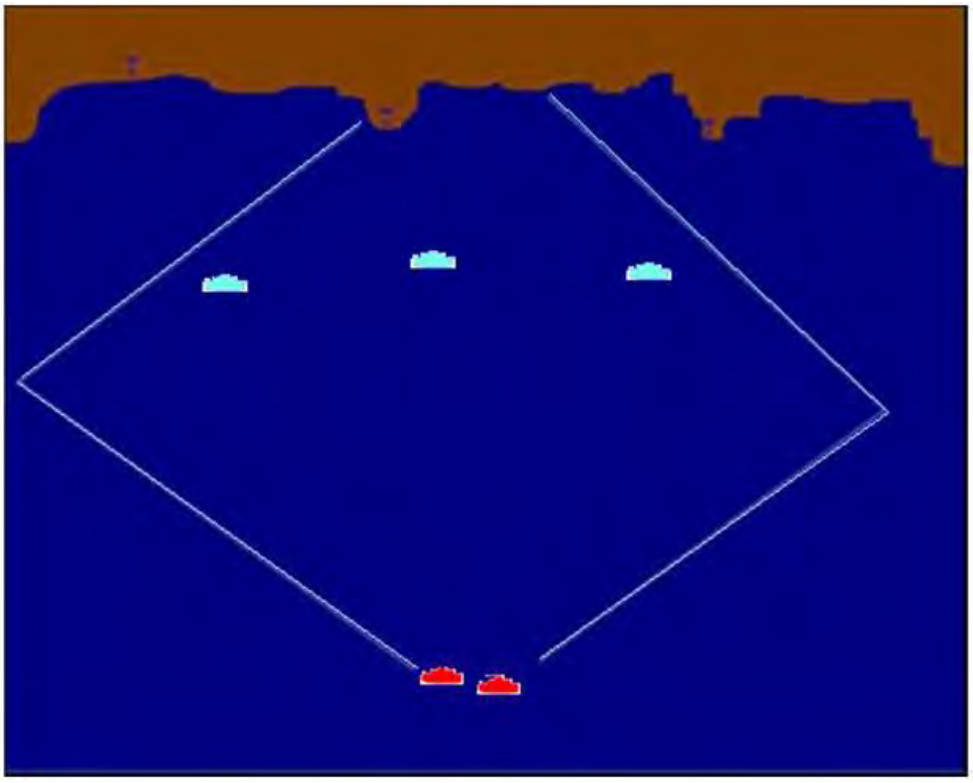

Figure 3.8: An evolved strategy for two red boats to penetrate the blue patrolling boats

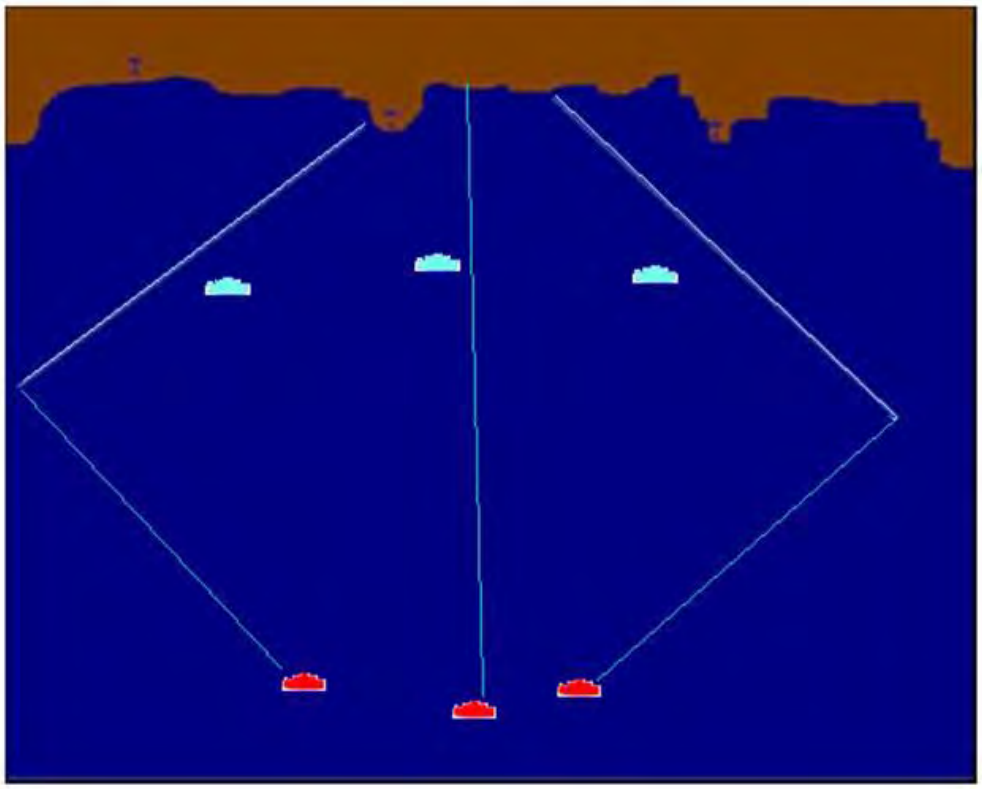

Figure 3.9: An evolved strategy for three red boats to penetrate the blue patrolling boats

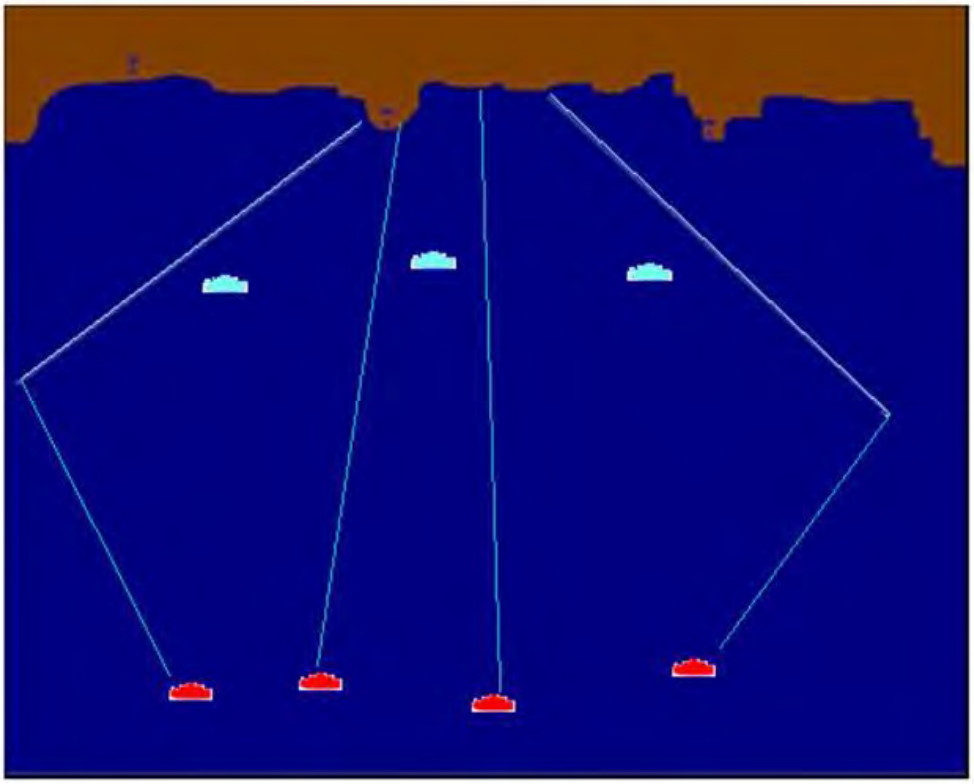

Figure 3.10: An evolved strategy for four red boats to penetrate the blue patrolling boats

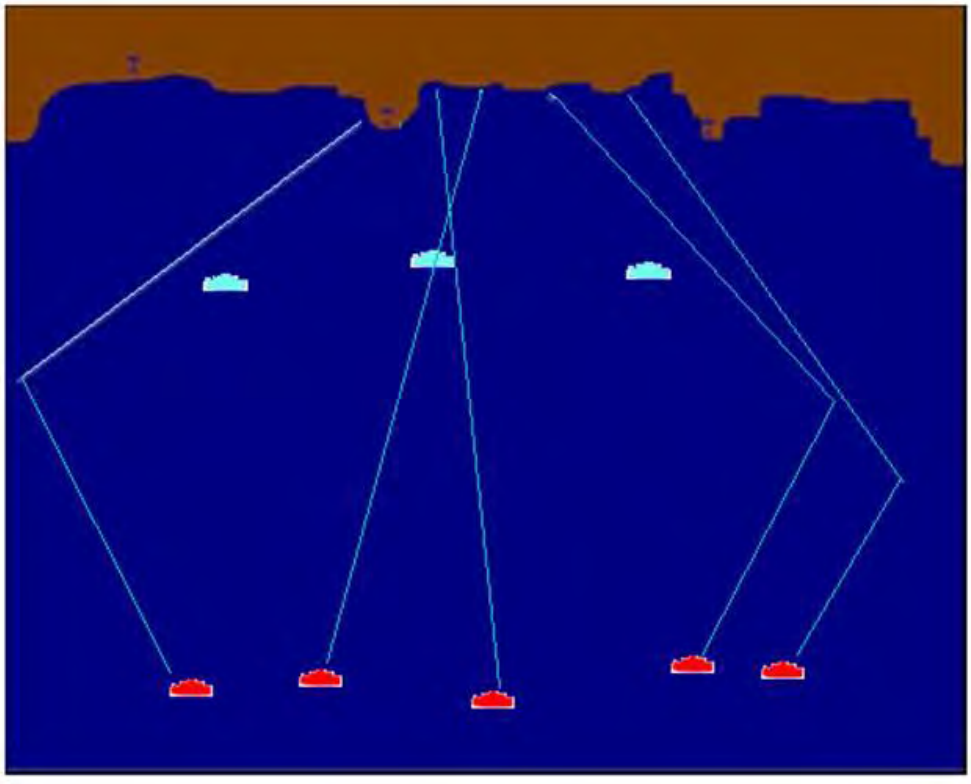

Figure 3.11: An evolved strategy for five red boats to penetrate the blue patrolling boats

When there are more than three attackers, they use saturation strategies which are depicted in Figure 3.10, with 4 red boats, and Figure 3.11 with 5 red boats. Some boats are used as a distraction and follow direct confrontation tactics. They move towards the blue surveillance area by maintaining distance with friendly boats. Simultaneously, some red boats narrowly escaped by avoiding confrontation and moving through areas in which there was no surveillance.

The evolved tactics indicate that the red boats should follow the flanking strategy to increase its success, when there are a smaller number of red boats involved. However, a mixed strategy is recommended when there are a larger number of attackers involved in breaching 3 patrolling defenders. The result also demonstrated that there are different levels of internal cooperation among the red agents when the number of the red boats is varied. This cooperation is stronger when a larger number of red agents are involved in the penetration process. Conversely, they maintain their distance if there is a smaller number of agents involved, which leads them to follow flanking strategies to avoid casualties. As the number of the red agents increases, their tactics change from flanking to direct confrontation.

Table 3.4 shows the evolved personality values for the red team with two, three, four and five agents (the descriptions for variables shown in the columns in Table 3.4 are detailed in Table 3.2). These values indicate that the red agents stay away from the blue boats while they also maintain a distance between friendly agents. The flanking tactics and increased speed help the red agents to avoid confrontation with the blue agents and reach their goal. The red teams with the given characteristics succeed almost entirely in achieving their goal while minimizing their casualties. The negative value under „Enemy" for the red team shows the repulsion with their competing team, the blue. The positive and negative value in the „Friend" rows show closeness with, and distance from, the friendly boats.

Scenarios with agent personalities, as listed in Table 3.4, were further analysed to evaluate their effectiveness. For this, an additional 50 repetitions of each scenario were run in MANA. Table 3.5 tabulates the mean MOEs and fitness values for different numbers of red agents.

Table 3.4: Personality suggested by OT for a red team

Personalities/ Number of Boats			2	3	4	5
Distance from Enemy	Agent SA	Enemy	-90	-60	-83	-93
		Enemy Threat 3	-95	-98	-99	-98
	Squad SA	Enemy Threat 3	-85	-75	-90	-87
Distance from Friends	Agent SA	Uninjured Friend	-96	-35	30	50
	Squad SA	Friend	-65	-20	22	35
	Inorganic SA		0	0	0	0
Movement Speed			100	100	100	100

Table 3.5: Mean casualties and success rate of optimized red team.

Red Agents	Mean Casualties	Std. Dev. (+/-)	Mean Success Rate	Std. Dev. (+/-)	Fitness
2	0.38	0.07	0.95	0.02	5.54
3	0.65	0.19	0.96	0.05	11.03
4	0.7	0.10	0.97	0.02	19.04
5	1.24	0.14	0.98	0.02	28.26

The results in Table 3.5 indicate that there is a direct relation between the number of agents involved in penetration and their success rates. Additionally, as the number of agents involved in the scenario increased, their attrition also increased.

3.4.2 Evolving Blue Boats

To explore the strategy options further, another experiment was devised to optimize blue agents against the already optimized red agents. In this case, only the scenario with two red boats was considered in which the blue strategies are evolved against the red strategy. The default personality values for the red boats are shown in the second

column of Table 3.4. GA parameters were the same as in the previous experiments, as depicted in Table 3.3. Two factors are considered as MOEs to evaluate individuals: maximizing the red casualties and stopping the red boats from passing through the patrolled area. The formula used in the fitness function is depicted in Equation (3.2):

$$Fitness = Mean\ red\ casualties - Red\ goal\ success\ proportion + 3 \quad \textbf{(3.2)}$$

With two red boats in the scenario, the theoretical maximum for the blue team fitness function (mean red casualties = 2 and red goal success proportion = 0) is 5. After the application of OT, the evolved characteristics for the blue team for defeating the red boats are shown in Table 3.6. The evolved tactics for the blue boats in response to the optimized red boats strategy is more aggressive and active behaviours. Despite the use of flanking tactics by the optimized red boats, the optimized blue boats were capable of taking action against them. Against the default blue strategy, optimized red boats would reach their destination almost all the time. However, when the blue team was optimized, the red team winning ratio was reduced by one third. The fall in the red winning ratio after blue optimization indicated that improved tactics could address the weaknesses of the plan if they were identified in advance.

Table 3.6: Optimized personality of the blue agents

Personalities			Evolved value
Distance with Enemy	Agent SA	Enemy	100
		Enemy Threat 3	100
	Squad SA	Enemy Threat 3	25
Distance with Friends	Agent SA	Uninjured Friend	-84
	Squad SA	Friend	-71
Movement Speed			100

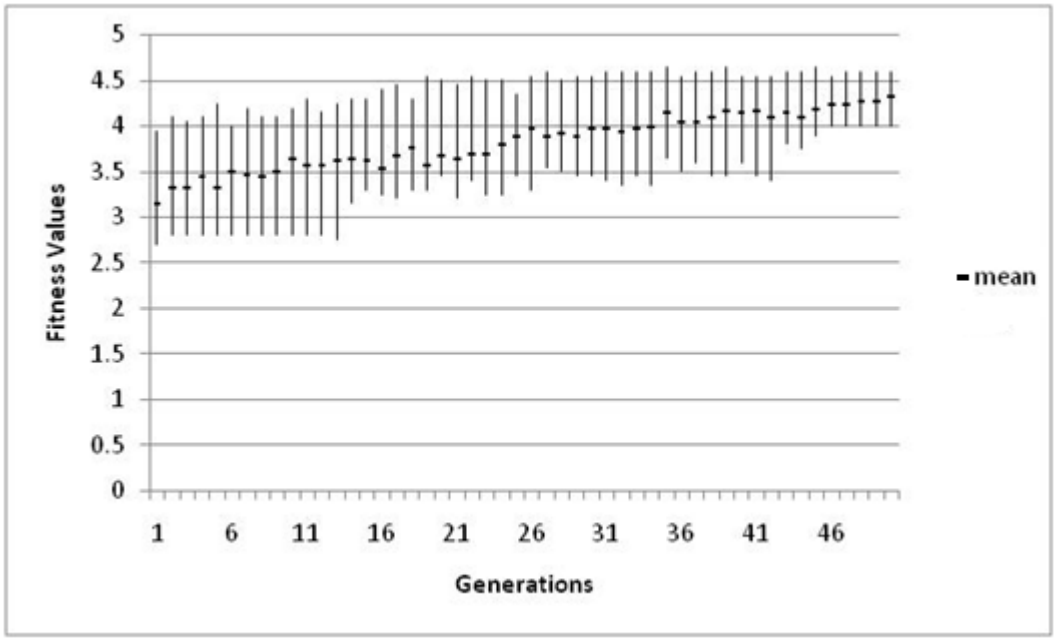

Figure 3.12: Mean fitness values, along with an indication of the range of fitness values, of the blue team while considering two red boats trying to penetrate three blue patrolling boats in the scenario.

Figure 3.12 depicts the progress of GA via the fitness values of the blue teams in each generation. The graph indicates that the gaps between maximum and minimum values are wide in every generation and convergence is hard to attain when optimizing the blue team against an already optimized the red team.

Table 3.7: Mean casualties and success rate of red boats after optimizing the blue boats.

Red agents	Mean Casualties	Std. Dev. (+/-)	Mean Success Rate	Std. Dev. (+/-)	Fitness
2	1.46	0.09	0.32	0.07	4.14

It was observed that when one team was optimized by keeping another team‟s strategy fixed, the optimized team identified various strategies that defeat the opponent team. Subsequently, as the first team‟s optimal strategy was kept fixed and the second team was then optimized against it. The second team then found some good strategies to defeat the first team‟s strategies. This indicated that even optimal strategies may be defeated when the opponent team is optimized against them. This provided sufficient

evidence that the tool was capable of generating an effective strategy to address the opponent‟s strategy. However, it also illustrated that taking turns in terms of evolving one team at a time against their opponent‟s specific optimal strategy is not practicable. Therefore, a comprehensive analysis would have to consider the range of possible red tactics and their likelihood of success but the fact is that GAs only support the optimization of a single population at a time. However, CEAs are capable of evolving two populations simultaneously.

3.5 Conclusion

This chapter presented different tactics which emerged in response to the blue patrolling boats while different numbers of red attackers were involved. When the blue boats were optimized against the already optimized strategy of the red boats, the blue boats again found a better strategy to respond to the red boats‟ optimal strategy. While the simple approach illustrated here can be used to gain valuable insights into a scenario, in general, the situation is very complicated and CEAs may be better approach as they can evolve multiple populations simultaneously.

4 Problems, Algorithms and Performance Measures

Chapter 3 presented a pilot study using GAs. The problem investigated in the pilot study, in which the RT scenario was optimized, showed that the optimization of RT applications can be enhanced by evolving two teams simultaneously. This task cannot be performed by using GAs as they can evolve only one population against a fixed strategy. CEAs are capable of evolving multiple populations at the same time. However, there are some pathologies which may be problematic for the performance of CEAs in finding good solutions when applied to RT applications. Additionally, there are some RT characteristics which are associated with the manifestation of these pathologies. Therefore, the first section of this chapter discusses RT characteristics. The second section presents CEAs, with and without variants designed to address CEA pathologies. The final section discusses various techniques that measure the performance of CEAs.

4.1 RT Characteristics

RT and its strengths and limitations have been reviewed in Chapter 2. This section presents two characteristics, intransitivity and multimodality, of RT applications that are investigated by this study. Details of these characteristics are explained in the following sub-sections.

4.1.1 Intransitivity in RT

A problem domain is intransitive when a simple ranking of solution strength cannot be performed. For example, intransitivity occurs if a solution *A* is better than solution *B*, and *B* is better than solution *C*, yet *C* is better than *A*, as in the example of the rock-paper-scissors game. The literature (section 2.2.4.2) showed many CEA domains demonstrate intransitivity characteristics (DeJong et al., 2004). With regard to RT, a strategy that is considered ineffective against one opponent may turn out to be a winning strategy for another (Sidran, 2004). This indicates that a simple ranking of good strategies may not be meaningful, which suggests that intransitivity also occurs in RT applications.

4.1.2 Multimodality in RT

A multimodal problem demonstrates a number of local optima which are better than all their neighbouring solutions (however, their fitness may be below the globally optimal solution). Eiben and Smith (2003) suggested that most domains demonstrate multimodal characteristics. However, to the knowledge of this author, no literature has been found that investigated issues of multimodality associated with RT. Based on the assumption that there may be more than one high quality solution to address strategies of the opponents, a further investigation was conducted on the output produced by the pilot study in chapter 3. It was designed to find whether the domain demonstrated multimodality. A "peak detection technique" was developed to carry out this investigation. A description of the technique is presented in the following section.

4.1.2.1 Peak Detection Technique

This technique is used to check whether a point (i.e. solution) in an evolved population is a local optimum. The technique involves taking a point into consideration and a number of points that are similar to it are randomly generated. A radius value is used to limit how far away these points can be generated from the specified point of interest. Figure 4.1 depicts the area of the „small value", the radius and the randomly generated points (i.e.18 points in Figure 4.1). If a generated point falls within the range of the „small value" (red stars in Figure 4.1) then that point is removed and a substitute point is then generated. The rationale is that we are satisfied thatwe have found a local optimum if the point in question is "close enough" to a real local optimum.

Subsequently, fitness values are calculated for all those points that are within the radius value (blue stars in Figure 4.1) by evaluating against a fixed test set (see Table 4.4). The differences between the fitness value of each generated point and the original point (white star) is compared with an assigned epsilon value. If the fitness differences are not bigger than the epsilon value then each of those points is regarded as being approximately the same fitness as the original. If the fitness of the original point is higher than or equal to all other points, then the original point is a local optimum. If one of the randomly generated points has a higher fitness value than the original point of interest, then it indicates that the original point is not a local optimum.

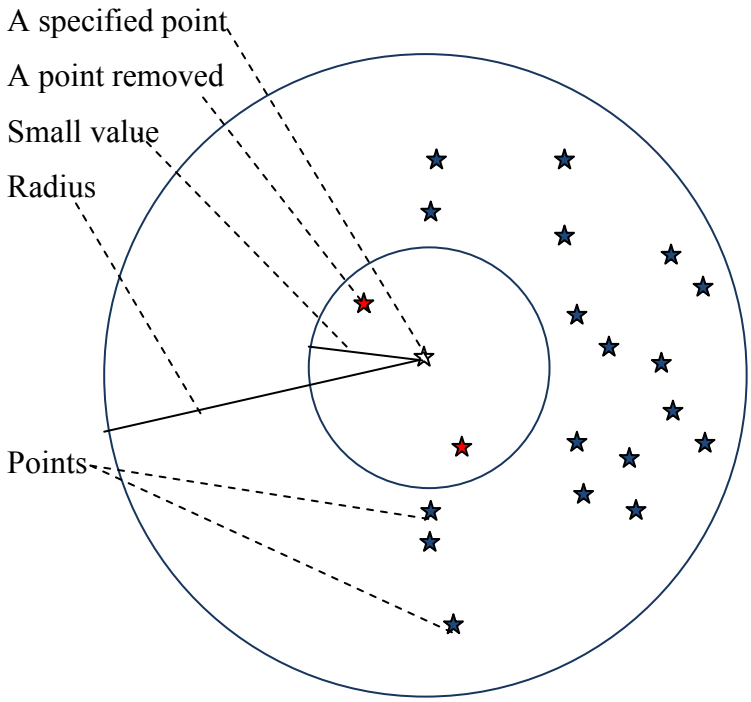

Figure 4.1: Randomly generated points around the specified point with radius and small value

4.1.2.2 Multimodality test in RT strategies

To explore if multimodality exists in the solution space of RT, an analysis using the technique described in section 4.1.2.1 conducted on data generated from the pilot study technique with similar parameters. Since the population size was 20 in the pilot study (see Chapter 3), the final evolved population consists of 20 strategies (after 50 generations) which is depicted in Table 4.1 (this data was generated from the case of involving 5 red boats competing with 3 blue boats). The descriptions for variables shown in the columns in Table 4.1 are described in Table 4.2 with the minimum and maximum range of each variable. Each individual in the population consists of a genome with the length of 12. Each gene is a real number ranging from 0 to 100. The peak detection technique was applied to these 20 strategies (individuals) to test for local optima. In their evaluation, for each evolved strategy (i.e. point of interest), 20 other similar strategies were generated randomly within a radius value of 5 and using a "small value" of 0.2. To calculate the fitness value of each strategy, simulations each with 40 iterations were executed. As detailed above, the fitness values associated with the points of interest (i.e. one of the evolved strategies) and those randomly generated strategies were compared using an epsilon value of 0.025. Of the 20 strategies tested, first 5 strategies were identified as local optima.

Table 4.1: Strategies studied

Alive Enemy	En Threat 1	Alive Friend	Cover	Conce-alment	Move-ment Speed	Min Dist to Enemy	Min Dist to Friend	Org Threat 3	Org Squad Friend	Min Dist to Org Threat	Min Dist to Squad Friend
39.17	**-81.46**	**100**	**11.91**	**-59.18**	**100**	**2705.03**	**7904.84**	**-95.88**	**45.71**	**0**	**4217.46**
70.98	**-91.51**	**100**	**11.91**	**-61.58**	**100**	**6931.15**	**9706.83**	**-71.66**	**38.53**	**0**	**961.04**
39.17	**-81.46**	**100**	**11.91**	**-61.58**	**100**	**6931.15**	**9706.83**	**-94.73**	**38.53**	**0**	**4217.46**
33.88	**-100**	**87.64**	**77.19**	**-45.77**	**97.2**	**7015.86**	**10000**	**-94.73**	**38.53**	**0**	**1821.49**
51.61	**-81.46**	**100**	**11.91**	**-61.58**	**100**	**6931.15**	**9706.83**	**-94.73**	**38.53**	**0**	**961.04**
51.61	-100	98.99	100	-61.58	100	6931.15	9706.83	-71.66	38.53	0	961.04
100	-100	-10.87	-82.08	-61.58	100	6931.15	9706.83	-56.07	42.89	0	216.95
46.05	-100	81.84	45.9	-38.14	100	6308.67	8425.06	-100	100	0	1277.5
51.61	-100	98.99	100	-61.58	100	6931.15	9706.83	-94.73	38.53	0	961.04
39.17	-81.46	41.36	64.35	11.83	90.79	8906.94	7044.59	-100	-38.52	0	664.28
51.61	-81.46	100	11.91	-61.58	100	8254.66	9706.83	-94.73	38.53	0	961.04
51.08	-100	86.45	11.91	-61.58	100	6931.15	9706.83	-94.73	38.53	0	4217.46
39.17	-100	86.45	75.33	-93.65	98.45	6795.49	6652.5	-79.31	17.54	1782.48	10000
70.98	-91.51	65.22	44.05	-45.27	95.62	7649.54	8862.75	-64.78	29.81	0	628.85
84.31	-58.12	12.66	38.76	-66.01	100	5756.91	7458.15	-100	45.75	1130.99	871.56
73.85	-71.2	100	30.76	-13.24	87.51	7523.71	10000	-67.4	51.95	18.33	965.64
43.95	-77.05	29.71	72.57	-49.07	87.3	7123.05	8762.69	-88.54	28.72	934.4	3161.41
33.66	-82.77	100	3.19	-44.36	73.86	6939.74	9165.11	-97.94	34.25	1193.31	5900.37
91.52	-53.78	99.59	49.12	-79.94	67.28	7339.83	8757.65	-82.39	36.56	0	0
72.83	-100	100	-72.63	-27.52	84.91	6798.23	10000	-100	100	2160	0

When utilizing the peak detection technique described in section 4.1.2.1 in evaluating the strategies found in the pilot study, associated with a specific RT scenario, it showed that there is more than one local optimum. This means that, within a population, there exists more than one high quality strategy to address the opponent‟s plan. This result provided some evidence that the solution space associated with RT scenarios may possess multimodal characteristics.

Table 4.2: Properties description

Properties	Description	Min Range	Max Range
Alive Enemy:	Attraction or repulsion with the agent which consists a property called enemy allegiance	-100	100
En Threat1:	Attraction or repulsion with the agent with a property called enemy allegiance Threat Level 1	-100	100
Alive Friend:	Attraction or repulsion with the agent with same allegiance	-100	100
Cover:	Determine the distance of shooting by direct fire	-100	100
Concealment:	Determine the visibility of agents in the terrain	-100	100
Movement Speed:	Determine the number of cells agents move in a given time step	0	100
Min Dist to Enemy:	Minimum distance the agent can spot their enemy. 0 value indicates the agent cannot spot the enemy agent and 100000 means the agent is very active in spotting enemies	0	10000
Min Dist to Friend:	Minimum distance the agent can spot their friends	0	10000
Org Threat 3:	Attraction or repulsion with the agent with enemy allegiance with threat level 3 from organic squad	-100	100
Org Squad Friend:	Attraction or repulsion with the agent with friend allegiance from organic squad	-100	100
Min Dist to Org Threat:	Minimum distance the agent can spot their enemy agents.	0	10000
Min Dist to Squad Friend:	Minimum distance the agent can spot their friends from the same squad	0	10000

4.2 Problems Studied

The previous sections of this chapter presented results for some investigations for the pilot study, specifically multimodality in RT. In order to test CEA for RT characteristics, their performance is evaluated on carefully-designed test problems. For testing intransitivity, the problem developed by Watson and Pollack (2001) was utilised as the basis for further investigation. For testing multimodality, a new test problem (Ranjeet, et al., 2012) was introduced.

4.2.1 An Intransitive Number Problem

In order to verify that the CEA examined in this thesis can address intransitivity, which occurs in RT scenarios, a test problem called "an intransitive number problem", proposed by Watson and Pollack (2001) was used in this study. This problem was

designed to test the behaviour of CEAs in a situation where there is an intransitive relationship between the strength of the evolved strategies. Due to the complexities of the intransitive number problem, there is no single solution that always wins. In situations where both opponents are evolving, a cyclic pathology may result, and the opposing sides may cycle between solutions. The investigation involving CEAs and the intransitive number problem is described in chapter 5.

4.2.2 A Multimodal Problem

To test the performance of CEAs in handling multimodal problems, a test problem is presented in this study called the *n-peak* problem. The *n-peak* problem was introduced for CEAs with two competing sides. Since there are *n* equally good strategies in this problem, the challenge for CEAs is to locate as many as possible of these peaks. As this is a symmetric problem, both sides are evaluated using the same method. The investigation involving CEAs and the multimodal problem is described in chapter 6.

4.2.3 RT Scenarios

The third domain is the RT application in which two scenarios are tested. Each scenario consisted of different environments in which each team tries to achieve its own objective using specific strategies. The aim was to ensure that the algorithms can effectively optimize and provide good strategies as outcomes in RT scenarios. The investigations employing CEAs in RT scenarios are described in chapter 7.

4.3 Experimental Environment

When optimizing a scenario in the RT application, a simulation run does not require a long computation time. However, in the CEA a high number of interactions were required, which then extends the time needed for the overall optimization process. In order to shorten the simulation running time, an environment was built by establishing a cluster using Shoal (see section 3.1.5). Experiments on the intransitive number problem and an RT scenario were conducted, with and without the use of Shoal to validate the

outcomes. The results produced with and without Shoal were consistent. In addition, the use of Shoal significantly lowered the computational time for optimizing the RT application. Shoal was not used in the investigations of the intransitive number problem and the multimodal problem. However, it was used in optimizing RT scenarios when CEAs, with and without variants, were used.

4.4 CEAs and Variants

The pilot study showed the optimization of each team in terms of evolving a better strategy against the opposing team‟s plans using GAs. In the pilot study evolution takes turns to fix the strategy of one team and evolving the competing team. Since the optimization takes place against one or few known strategies, the process may not be effectively applied when there are multiple strategies to be addressed. However, CEAs are capable of optimizing multiple strategies simultaneously for multiple populations.

Initially, the developed optimization tool (OT) used a GA which was subsequently modified to incorporate CEAs. The OT can be used with a GA or CEA on the basis of problems to be solved. The tool can be used to optimize single or multiple populations using the GA or the CEA respectively. However, CEAs suffer from various pathologies (Wiegand, 2003; Ficici, 2004) such as *cycling*, *disengagement* and *forgetting*. These pathologies may cause the CEA to become ineffective, as it may find the same solutions again and again (*cycling*), a population may outperform another population and cause its individuals to become indistinguishable in terms of fitness (*disengagement*), and during the optimization process it may forget the good solutions from the preceding generations (*forgetting*). Additionally, some RT characteristics including intransitivity and multimodality need to be addressed in order to improve the effectiveness of the optimization process.

Therefore, test problems associated with each characteristic are first investigated using a basic CEA and a number of variants that incorporated techniques for FS, HOF and a combination of both. The aim is to explore how well these algorithms will perform in these isolated problem domains (i.e. each problem with only one of the characteristics) before applying them to optimise RT scenarios

Figure 4.2 presents various CEAs that were created by integrating different variants in the naïve CEA to optimize the applications used in this thesis. Their naming conventions are included in Figure 4.2.

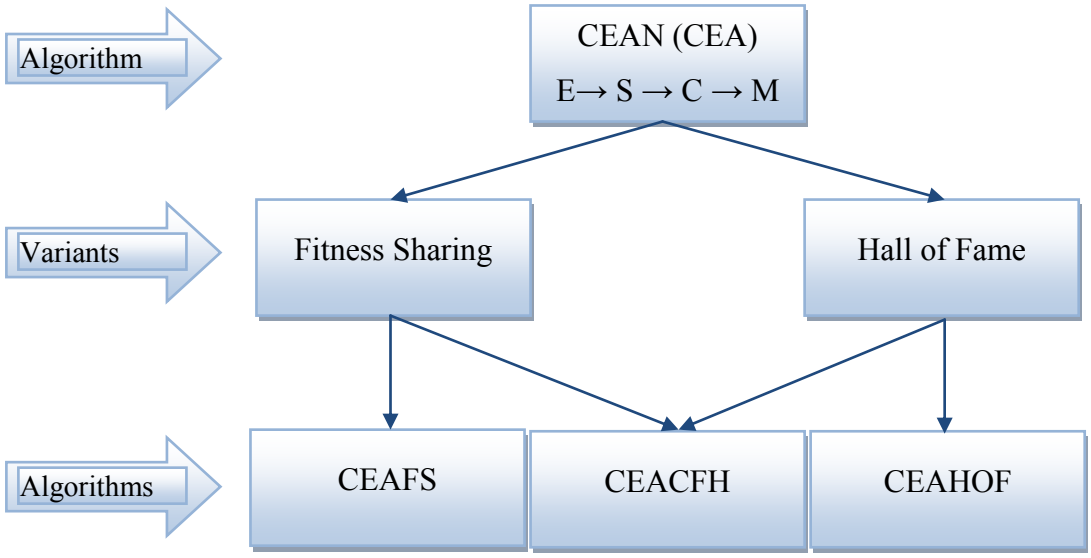

Figure 4.2: A naïve CEA and three variants that use evaluation (E), selection (S), crossover (C) and mutation (M) operators.

4.4.1.1 Naïve Coevolutionary Algorithm (CEAN)

The initial level of Figure 4.2 depicts the CEAN which is a naïve CEA; the adjective "naïve" indicates that no variants were used in this algorithm. Its characteristics are the same as those of the algorithm explained in Figure 2.6 (section 2.2.4). The specific CEA operators for Evaluation (E), Selection (S), Crossover (C) and Mutation (M) are now discussed. A stochastic uniform selection operator was used to select parents. A single point crossover was used as a crossover operator. For the intransitive number problem and RT scenarios domain, a polynomial mutation was used whereas a Gaussian mutation was used for the multimodal problem. The termination criterion of the optimization process was a specific number of generations, i.e. the process iterates until a specified number is completed. The procedures of those operators were explained in section 2.2.4.1.

In order to address CEA pathologies, the FS and HOF variants were integrated in the CEAN individually and in combination as depicted in the second level of Figure 4.2. The third level shows the algorithms that were generated by integrating variants. The descriptions of these algorithms are presented in the following sections.

4.4.1.2 Coevolutionary Algorithm with Fitness Sharing (CEAFS)

In order to enhance the CEAs‟ optimization capability, the FS technique described in section 2.2.4.3, proposed by Goldberg and Richardson (1987) was used in this thesis. FS is integrated into the CEAN to create an algorithm CEAFS. Researchers including Rosin and Belew (1997) and DeJong (2007) assert that FS helps to address various CEA pathologies mainly *cycling* and *disengagement*.

1. Randomly initialize Population1 and Population2
2. Evaluate each individual of Population1 with Population 2 to determine the values $score_{i,j}$ as in Equation (2.1)
3. Store each evaluated score in a matrix format as shown in Table 2.4
4. Calculate fitness values of Population1‟s individuals by averaging the column values. Likewise average row values from the matrix to calculate fitness values of individuals from Population2
5. Divide previously calculated fitness of each individual by niche count as in Equation (2.5) and receive new fitness values
6. Copy the fittest individual in each population (the *elite* individual) into the next generation of each population
7. Do until a new generation of Population1 is completed:
   (a) Select two parents from the old Population1 according to a selection function based on fitness (Detailed in Section 2.2.4.1)
   (b) Perform crossover according to the crossover operator described in section 2.2.4.1 and apply mutation as per in section 2.2.4.1 to obtain two new offspring
   (c) Add the new offspring to the new generation
8. Repeat step 7 for Population2
9. Repeat steps 2-8 for the required number of generations. The solution is the final two populations.

Note: The highlighted line is the only different step in this algorithm from the CEAN

Figure 4.3: Pseudo code of CEAFS

The optimization of any problem using this algorithm was similar to the CEAN except for the method of the fitness calculation. In the CEAN, the parents were selected on the

basis of the raw fitness (the fitness that individuals receive by evaluating against the other population members). However, in the CEAFS the raw fitness is divided by the niche count as shown in pseudo code in Figure 4.3

In this thesis, the value of τ in Equation (2.6) was set to 1, which was as suggested by Sareni and Krahenbuhl (1998). To determine a value for the niche radius, n_r, a method suggested by Horn and Nafpliotis (1994) was used. According to this method, the total surface area of frontier is divided by population size to get the approximate niche radius value. The total surface area of the frontier is calculated by using Euclidean distance between the upper and lower bounds of the attributes used in the search space.

When the approximate niche radius value was calculated using Horn and Nafpliotis (1994)‟s method, again a series of experiments were conducted using a suggested niche radius value and its smaller or bigger nearest value. Subsequently, the best fit niche radius value was selected to use in the experiment. The analyses for finding a suitable niche radius were conducted in all domains studied in this thesis (an intransitive number problem, a multimodal problem and RT scenarios). The values used for the niche radius in each of these domains are discussed in the associated chapters.

4.4.1.3 Coevolutionary Algorithm with Hall Of Fame (CEAHOF)

The CEAHOF used in this study was created by integrating a memory mechanism, the HOF, in the CEAN. The HOF introduced by Rosin and Belew (1997) was used in this study to address the CEAs‟ forgetting pathology. Rosin and Belew (1997) recommended populating the HOF by choosing one best solution from each generation. Nolfi and Floreano (1998) found that the concept of adding one elite individual from every generation may be problematic, as the HOF size also increases, requiring a large amount of memory. Also, in subsequent generations, evaluation of individuals will focus more upon non-evolving past strategies than the evolving competing population.

1. Randomly initialize Population1 and Population2
2. Create memory spaces for HOF1 and HOF2 (HOF1 Size = Population1 size and HOF2 Size = Population 2 Size)
3. Evaluate each individual of Population1 with Population 2 and also with HOF2 individuals (if the size of HOF2 != 0) to determine the values $score_{i,j}$ as in Equation (2.1)
4. Yet again evaluate each individual of Population2 with HOF1 individuals (if the size of HOF2 != 0)
5. Store each evaluated score in a matrix format as shown in Table 4.3
6. Calculate fitness values of Population1‟s individuals by averaging the column values. Likewise average row values from the matrix to calculate fitness values of individuals from Population2
7. Copy the fittest individual in each population (the elite individual) into the next generation of each population and also in the HOF. If the HOF is full (i.e. the number of individuals in the HOF = HOF size) then compare the fitness value of the HOF individuals and remove the individual with lesser fitness value
8. Do until a new generation of Population1 is completed:
   (a) Select two parents from the old Population1 according to a selection function based on fitness (Detailed in Section 2.2.4.1)
   (b) Perform crossover according to the crossover operator described in section 2.2.4.1 and apply mutation as per in section 2.2.4.1 to obtain two new offspring
   (c) Add the new offspring to the new generation
9. Repeat step 8 for Population2
10. Repeat steps 2-9 for the required number of generations. The solution is the final two populations.

Note: The highlighted lines are the only different steps in this algorithm from the CEAN

Figure 4.4: Pseudo code of CEAHOF

In this study, a limitation of the HOF size was applied, i.e. the number of HOF individuals cannot exceed the pre-defined maximum size of the HOF. The maximum HOF size was set to be the same as the current population size; to ensure that the HOF had sufficient room to store past strategies and also to minimize the memory consumption. When the allotted HOF space is full, individuals selected later for insertion into the HOF are compared with existing HOF individuals. On the basis of their fitness values, the weakest individuals are replaced from the HOF. It was noticed that when the best individuals are repeatedly selected, all individuals may become genotypically similar in the HOF. In future research, removing the most genotypically similar individual from the HOF can be considered.

The CEAHOF behaves in a similar manner to the CEAN, except for the method of the fitness calculation. In this algorithm, steps shown in Figure 4.4, the individuals‟ fitness was calculated not only by evaluating them against all individuals from competing populations, but also against the HOF members. The process of fitness calculation used by this algorithm is depicted in Equation (2.8) which is similar to Equation (2.1) except for the use of combine population size (population size + HOF members). The symbol *n* is the population size of the competing population; *m* is the size of the HOF archive, which starts from 0 and increases by one in every generation until it reaches the maximum size defined by the user.

Table 4.3: Matrix format showing a fitness calculation method in the HOF.

		Population			HOF			
		P1I1	P1I2	P1I3	P1HI1	P1HI2		
Population	P2I1	1	1	5	3	3	2.6	Row Average (Fitness of population 2)
	P2I2	8	9	2	4	9	6.4	
	P2I3	4	0	10	9	3	5.2	
HOF	P2HI1	7	1	2	X	X	X	
	P2HI2	7	4	6	X	X	X	
		5.4	3	5	X	X	X	
	Column Average (Fitness of Population1)							

In Table 4.3, the symbols P, I and H represent Population, Individual and Hall Of Fame respectively.

4.4.1.4 Coevolutionary Algorithm with Combined HOF and FS (CEACFH)

The CEACFH used in this thesis was created by integrating the combine approaches of FS and HOF into the CEAN as depicted in Figure 4.5. FS is a well known diversity maintenance technique which has been used in CEAs to address the cycling and disengagement pathologies and the HOF preserves the old strategies and introduces them at the time of evolving a new generation, which helps to avoid the CEAs‟ forgetting pathology. An interesting exploration is to integrate these two proven

techniques into the CEAN and then to examine the algorithms‟ capability to address various pathologies of CEA as well as intransitivity and multimodality.

1. Randomly initialize Population1 and Population2
2. Create memory spaces for HOF1 and HOF2 (HOF1 Size = Population1 size and HOF2 Size = Population 2 Size)
3. Evaluate each individual of Population1 with Population 2 and also with HOF2 individuals (if the number of HOF2 != 0) to determine the values $score_{i,j}$ as in Equation (2.1)
4. Yet again evaluate each individual of Population2 with HOF1 individuals (if the number of HOF2 != 0)
5. Store each evaluated score in a matrix format as shown in Table 4.3
6. Calculate fitness values of Population1‟s individuals by averaging the column values. Likewise average row values from the matrix to calculate fitness values of individuals from Population2
7. Divide previously calculated fitness of each individual by niche count as in Equation (2.5) and receive new fitness values
8. Copy the fittest individual in each population (the *elite* individual) into the next generation of each population and also in the HOF. If the HOF is full (i.e. the number of individuals in the HOF = HOF size) then compare the fitness value of the HOF individuals and remove the individual with lesser fitness value
9. Do until a new generation of Population1 is completed:
   (a) Select two parents from the old Population1 according to a selection function based on fitness (Detailed in Section 2.2.4.1)
   (b) Perform crossover according to the crossover operator described in section 2.2.4.1 and apply mutation as per in section 2.2.4.1 to obtain two new offspring
   (c) Add the new offspring to the new generation
10. Repeat step 9 for Population2
11. Repeat steps 2-10 for the required number of generations. The solution is the final two populations.

Note: The highlighted line is the only different step in this algorithm from the CEAHOF

Figure 4.5: Pseudo code of CEACFH

FS implicitly influences the algorithms to maintain diversity by giving a high priority to unique solutions; however, the HOF may reduce the diversity of the population. The HOF collects a best solution from each generation due to which the HOF members may be similar in structure. When individuals are evaluated against the HOF members, individuals with similar structure get equal response from the HOF which reduces the diversity of the population. Despite the contradicting characteristics (FS increases and HOF decreases diversity) of these two techniques in terms of diversity maintenance, a

question of interest would be: if these techniques were capable of addressing CEA pathologies individually, how will the combination perform?

The pseudo code in Figure 4.5 shows the steps in this algorithm. They are similar to CEAHOF except for the additional calculation for fitness sharing. As in CEAHOF, individuals were evaluated not only against the individuals from the opposing population but also against the HOF members. The score gained by individuals in each interaction were stored in a matrix format as shown in Table 4.3. The fitness of individuals was calculated by averaging the scores by column or row wise. Subsequently, the calculated fitness was processed for FS as in the CEAFS.

The fitness calculation of this algorithm is given in Equation (2.5) in which the raw fitness f_i was calculated by using Equation (2.8), the n and m are the population size of the competing population and HOF population respectively. The niche count c_i was calculated according to Equation (2.6). The symbol n and m is a total population size and the HOF size respectively and i and j are two competing individuals.

4.5 Performance Measures

Now that the test problems and CEA algorithms that will be employed have been described, attention turns to the evaluation of the performance of these chosen CEAs. All chosen test problems were used to evaluate all four algorithms, CEAN, CEAFS, CEAHOF and CEACFH. The performance of these algorithms was measured using a number of evaluation techniques. The quality, based on each individual‟s fitness value, and diversity (measured on the basis of genome and fitness value diversity) of the population are measured. The performance measures studied in this thesis are: the generalisation performance (GP), Peak Ratio, Success Ratio and Circular Earth Movers‟ Distance (CEMD). The following sub-sections present descriptions of these techniques.

4.5.1 Estimated Generalisation Performance

In CEAs, the performance of one evolving population, on the basis of their fitness, against its evolving opponents cannot be used as an objective measure of quality

because both these populations evolve simultaneously. In such situations, a technique called estimated generalisation performance can be used that measure the algorithms‟ performances by testing the evolving population against the fixed test population to monitor their progress in subsequent generations (Chong et al., 2008, 2009).

Chong et al. (2008, 2009) begin by defining estimated GP as the mean score of a solution in all possible test cases. This intuitively appealing definition poses several practical difficulties. First, for many problems of interest, the space of all possible test cases could be very large, or even infinite, and there may be no way to compute a mean score analytically. Therefore, they propose a statistical approximation approach, in which a mean score is computed for a suitable sample of test cases. The second difficulty is what probability distribution should be used over the space of test cases. In many cases, scores against "high quality" test cases might be considered more important. They therefore proposed two different methods for sampling the space of test cases: unbiased sampling (which is purely random) and biased sampling (which favours higher quality test cases).

Chong et al. (2008, 2009) used GP to measure the performance of a single population CEA whereas this technique was adapted for use involving multiple (two) populations CEAs in this thesis. There were some modifications from the original approach to account for the fact that two populations were used rather than one. The test cases were created for each participating population based on biased sampling. The procedure to obtain a biased test set for each population is outlined in Table 4.4.

Table 4.4: Procedures to generate a test set population

1. Generate *n* random test solutions for each population 2. Calculate fitness values by scoring these against each other 3. Select and save the fittest individual from each population 4. Repeat steps 1-3 until the desired number of test cases have been collected.

In this procedure, the larger the value of *n,* the more biased towards quality the test set becomes. In this thesis the value of *n* used was three times of population size in each of the domain problems (an intransitive number, multimodal problem and RT scenarios).

After generating these test sets, the quality of the evolving population can be estimated by evaluating each solution against the test set solutions. Chong et al. (2008, 2009) proposed three ways of measuring estimated GP: average, best and ensemble which are described below.

- **Estimated Average GP**

Although in an EA a population of solutions are evolved, usually the top few, on the basis of fitness, are of most interest. Thus, when calculating the quality of a population, the population is first sorted according to fitness values (since this is the only quality information available to the algorithm), and then only the best few, in term of highest fitness value, *nBest*, in the sorted list are considered. *Estimated average GP* is then estimated as in the Equation (4.1).

$$Estimated\ average\ GP = \frac{1}{nBest} \sum_{i=1}^{nBest} E_i \qquad \textbf{(4.1)}$$

Where $E_{i,}$ is the average score of solution *i* against the test set as defined in Equation (4.2).

$$E_i = \frac{1}{nTest} \sum_{j=1}^{nTest} score_{i,j} \qquad \textbf{(4.2)}$$

where*, nTest* is the size of the test set. $score_{i,j}$ is a fitness that individual *i* receives when competing against *j* individual. For calculating the *estimated average GP*, *nBest* is set to 5 and *nTest* is set equal to the population size in each domain studied in this thesis.

- **Estimated Best GP**

Another measure of estimated GP is *estimated best GP*. This is a quality of the best solution amongst the top *nBest* solutions in the population. The average score of each solution against the test set E_i is calculated as defined in Equation (4.2). The highest score of the *nBest* solutions will be considered as the *estimated best GP*. For calculating

the *estimated best GP*, as in the estimated average GP, *nBest* is set to 5 and *nTest* is set equal to the population size in each domain studied in this thesis.

$$Estimated\ best\ GP = \max_{i<nBest} E_i \tag{4.3}$$

- **Ensemble**

Ensemble examines the individuals in an evolving population that best defeat the competing population. In this approach selected individuals, *nBest*, compete against a number of fixed test set opponent individuals, *nTest*. Each *nTest* individual is tested against all *nBest* and each time the highest score when defeating an *nTest* individual is collected. The collected value is an ensemble for the specific evolving population. Equation (4.4) shows the calculation for ensemble.

$$Ensemble = \sum_{j=1}^{nBest} B_j \tag{4.4}$$

$$B_j = \max_{i=1\ to\ nTest} Score_{i,j}$$

All ways of measuring the estimated GPs are between the selected number of individuals from the fixed test population, *nTest,* and top individuals from the evolving population, *nBest.* In the entire thesis the size of *nTest* is equal to the population size and *nBest* is always 5 regardless of the problem, which was a number chosen based on the work of Chong et al (2008, 2009).

Since the Ensemble GP gathers only the highest score, the ensemble value often appears to be a maximum value. The *estimated best GP* chooses the best score among the top individuals; thus, the highest score is returned as the best GP-value. The *estimated average GP* is an average score of *nBest* individuals while competing against *nTest* individuals. Thus, the average GP‟s value is smaller in comparison to the ensemble and the best GP as all individuals may not be equally good in their performance. All GP-values are expected to be higher in subsequent generations in comparison to earlier generations.

4.5.2 Objective Quality Measurement

This performance measure was used only in the intransitive number problem as the domain"s objective quality can be measured according to the gene values. In this domain each individual contains two genes, and each represents a coordinate value. The coordinate value can be within a range of 0 to 100. The quality of solution *i* is calculated as Equation (4.5), the average of the solutions *x* and *y* values in which x and y are coordinate values. The measures for the objective quality of a population are then defined in a similar way as for estimated GP.

$$A_i = \frac{x + y}{2} \tag{4.5}$$

- **Objective Average Quality Measurement**

Similar to Equation (4.1), a specific number of top ranked (based on fitness value) individuals, *nBest,* are evaluated and their scores are averaged to calculate the *objective average quality*. The difference is that instead of evaluating against a test set the objective quality measure in Equation (4.5) is used. The average objective quality measure is depicted in Equation (4.6). For calculating the *objective average quality*, the *nBest* is set to 40% of the population size and *nTest* is set equal to the population size in the intransitive number domain.

$$Objective\ average\ quality = \frac{1}{nBest} \sum_{i=1}^{nBest} A_i \tag{4.6}$$

- **Objective Best Quality Measurement**

The *objective best quality* is calculated by considering the *nBest* top individuals (based on fitness value) and selecting the highest objective quality value from that set. The *objective best quality* is derived in Equation (4.7). For calculating the *objective best quality*, the size of the *nBest* is set to 5 and *nTest* is set equal to the population size in the intransitive number domain.

$$Objective\ best\ quality = \max_{i<nBest} A_i \quad \textbf{(4.7)}$$

4.5.3 Circular Earth Mover's Distance

This performance measure was used only in the multimodal problem. Since multimodality is one of the RT characteristics, a multimodal problem was proposed to check capabilities of CEAs in finding multiple optimal solutions. In the multimodal domain, a number of peaks can be specified and depending on the algorithms‟ performance the peaks are detected. The algorithms‟ performance (in finding multiple peaks) cannot be measured by only the GPs. The circular earth movers‟ distances (CEMD) (Rabin, et al., 2008) is proposed to specifically measure the algorithms‟ performance in detecting multiple peaks.

Earth mover‟s distance (EMD) (Rubner, et al., 2000) is a technique that measures the minimum total number of movements that would be required to make the two histograms identical. This technique has previously been used for comparing two histograms in image processing related research such as image retrieval (Rubner, et al., 2000), measurement of texture and colour similarities (Levina & Bickel, 2001), image matching (Ling & Okada, 2006) and image comparison (Rabin, et al., 2008). Rabin, Delon and Gousseau (2008) introduced the concept of CEMD, based on the EMD designed to compare one-dimensional circular histograms.

In the multimodal problem studied in this thesis, the CEMD is used to measure the performance of CEAs in identifying the multiple optima. The multimodal problem proposed in this thesis allows a user to specify the number of peaks. Since the true location of the peaks in the problem is known, an "ideal" distribution can be created in the form of a histogram for both the populations. In the histogram, all buckets that include the peak contain equal number of solutions and the remaining buckets which do not contain any peak are empty. Likewise, an actual histogram can be created for each evolving population which can be compared with the ideal histogram.

On the basis of similarities between the two histograms, the CEMD value is assigned. A CEMD value close to zero is considered to be good because a zero difference indicates

that the histogram for the evolving population is identical to the ideal histogram. Equation (4.8) is used to calculate circular earth movers" distance in which the evolving population"s histogram and the ideal histogram is represented by F_k and G_k respectively in which N number of intervals are allocated. The value of F_k and G_k can be calculated according to Equation (4.9)

$$CEMD(f,g) = \min_{k \in \{1,\ldots,N\}} \left\{ \frac{1}{N} \sum_{i=1}^{N} |F_k[i] - G_k[i]| \right\} \quad \textbf{(4.8)}$$

Where, $\forall\, k \in \{1, \ldots, N\}$, G_k can be calculated exactly the same way as F_k.

$$F_k[i] = \begin{cases} \sum_{j=k}^{i} f[j] & if\ i \geq k \\ \sum_{j=k}^{N} f[j] + \sum_{j=1}^{i} f[j] & if\ i < k \end{cases} \quad \textbf{(4.9)}$$

Table 4.5: Distribution for the ideal histogram with 40 buckets and 5 peaks

Peaks	P1		P2		P3		P4		P5		P1
Value	0.1	0 0 0 0 0 0 0	0.2	0 0 0 0 0 0 0	0.2	0 0 0 0 0 0 0	0.2	0 0 0 0 0 0 0	0.2	0 0 0 0 0 0 0	0.1
Bucket Number	1	2 to 8	9	10 to 16	17	18 to 24	25	26 to 32	33	34 to 40	1
0.2 0.1											
Range	0.0 - 0.0125		0.1875 - 0.2125		0.3875 - 0.4125		0.5875 - 0.6125		0.7875 - 0.8125		0.9875 - 1.0
	Peak at 0.0	Increase by 0.025 in each bucket	Peak at 2.0	Increase by 0.025 in each bucket	Peak at 4.0	Increase by 0.025 in each bucket	Peak at 6.0	Increase by 0.025 in each bucket	Peak at 8.0	Increase by 0.025 in each bucket	Peak at 0.0

For example, if the ideal histogram *(G_k)* has forty buckets (*N*=40) and the multimodal problem has five optima, each peak is positioned with 7 bucket intervals for an equal distribution as shown in Table 4.5. The bucket value is a distribution of 0 to 1. The first

and last buckets have 0.0125 intervals whereas remaining buckets have 0.025 intervals. Since it is circular in nature, the first peaks shares the bucket at the first half and last half as shown in Table 4.5 (P1 at the first and after the last bucket). These two half buckets are assigned 0.1 and 0.1 values respectively for the first half and last half buckets. Other peaks which are located at the 0.2, 0.4, 0.6 and 0.8 buckets are assigned with 0.2 values each. The remaining buckets are empty.

The actual histogram (F_k) can be calculated in the same way as G_k. Unlike in the ideal histogram distribution, individuals in an evolving population (only one gene in each individual in the proposed multimodal problem) are assigned to different buckets according to their gene values. Subsequently, the resulting distance is compared with the ideal histogram using Equation (4.9). A small gap between the actual and ideal histogram represents a success of algorithms in detecting all the mentioned peaks in the problem.

4.5.4 Peak Ratio and Success Ratio

To measure the performance of CEAs in detecting multiple optima, two additional measurement methods were used; the peak ratio and success ratio. These performance measures have previously been used for GAs (Beasley, et al., 1993) and differential evolution (DE) (Thomsen, 2004). Subsequently, Epitropakis, Plagianakos, Vrahatis (2011) and Otani, Suzuki and Arita (2011) also used these same techniques to measure the capabilities of DEs for multimodal optimization.

The peak ratio (PR) measures the percentage of global optima identified within a number of known global optima. In both evaluation methods, a tolerance level needs to be assigned (such as 10^{-2}, 10^{-3}, 10^{-6} and 10^{-7}). The tolerance level is an acceptable computed value to be considered as an optimum (for example the value 2.001 and 4.008 are considered as 2 and 4 if 10^{-2} tolerance level is set). The peak ratio is calculated according to Equation (4.10).

$$PR = \frac{identified\ peak\ numbers}{total\ peaks} \quad \textbf{(4.10)}$$

Another method, success ratio (SR), measures the number of times that all global optima are found in a specific number of runs. The SR is calculated according to Equation (4.11).

$$SR = \frac{Number\ of\ times\ all\ global\ optima\ identified\ in\ N\ runs}{N\ runs} \quad \textbf{(4.11)}$$

4.5.5 Diversity Evaluation Techniques

The purpose of measuring diversity was to examine whether the population remain equally diverse from the initial to final generations. In addition, diversity evaluation helps to identify whether the diversity influences the characteristics of the population such as the population‟s performance in achieving optimal solutions. In this thesis, two diversity measures were used, namely genotypic and phenotypic. These two diversity measures are described in the following sub-sections.

4.5.5.1 Genotypic Diversity Measurement

According to Herrera and Lozano (1996), the diversity of the population can be measured based on Euclidian distance between chromosomes. The authors have proposed this diversity measure, called "*genotypic diversity* measures based on Euclidian distance" for real coded GAs. However, this diversity measure is used for the multiple population CEAs in this thesis.

In this diversity measure approach, the distance between each gene from one individual to other individuals are measured. The diversity of the population is calculated on the basis of individuals‟ gene values within a population. In Equation (4.12), the value of D is the average of the genotypic variation d_j. The symbol n represents the number of individuals in the population.

$$D = \frac{1}{n}\sum_{j=1}^{n} d_j \quad \textbf{(4.12)}$$

d_j is calculated as the Euclidian distance between individuals‟ genes which is depicted in Equation (4.13). u is a length of the genome. The length of the genome in the intransitive number problem was two. The genome length in the multimodal problem was one. With regard to RT scenario, coastline protection, the length of genome for the blue was 15 and the red was 12. x_m is a gene from an individual which is being evaluated and $y_{j,m}$ is a gene from j individual from the same population.

$$d_j = \sqrt{\sum_{m=1}^{u} (x_m - y_{j,m})^2} \tag{4.13}$$

4.5.5.2 Phenotypic Diversity Measurement

While *genotypic diversity* was measured in the population, the *phenotypic diversity* was also calculated. The *phenotypic diversity* is a measure based on fitness of the population in this study. This diversity measurement method evaluates the population according to the similarity or dissimilarity of the fitness value that individuals receive.

In order to calculate *phenotypic diversity* of the population, this study used the entropy concept introduced by Ray (1993). The process of calculating entropy begins with the creation of a number of clusters (N) (in this study, on the basis of individual‟s fitness value). The interval between the clusters (R) is calculated by using the formula, $R = \frac{S_{max} - S_{min}}{N}$, so that N evenly spaced clusters are created between maximum score (S_{max}), and the minimum score (S_{min}). All individuals in the population are categorized into specific clusters (based on their fitness value).

The numbers of individuals belonging to the specific cluster are counted (p_k) then the following formula is applied to calculate the entropy of the population (E) as in Equation (4.14). If the cluster is empty, the value of p_k is zero.

$$E = -\sum_{k}^{N} p_k . log p_k \tag{4.14}$$

4.6 Summary

The first section of this chapter presented brief descriptions of test problem domains considered in this thesis. More details and experiments for these domains are presented in the subsequent chapters. Chapter 5, 6 and 7 present the analysis of the intransitive number problem, multimodal problem and RT scenarios respectively. Subsequently, a description of CEAs (with and without the use of variants) used in this thesis was explained in the second section. Some well known variants such as FS and the HOF were integrated in a CEA to enhance the algorithms capabilities. These algorithms will be utilized in subsequent chapters to optimize the considered domains.

In addition, descriptions of some techniques that measure the performance of algorithms were also explained. In this thesis, the performance measures evaluate two factors: the quality and diversity of the population. The performance measures including GP, CEMD, PR and SR measures quality. The diversity of the population is measured on the basis of their genotype and phenotype. These performance measures are also used in the subsequent chapters to measure the effectiveness of the algorithms in addressing specific problems. Now, the next two chapters present how a naïve CEA with three variants, address intransitivity and multimodality.

5 Testing CEAs on an Intransitive Problem

Chapter 4 provided descriptions of the problems, algorithms and the performance measures that are studied in this thesis. This chapter describes the evaluation of four algorithms in terms of their generalisation ability and the objective quality of solutions using an intransitive number problem. A naïve CEA and three variants were employed in solving this problem. This chapter begins with a definition of intransitivity in section 5.1 which is followed by the description of the test problem considered in section 5.2. Section 5.3 explains the parameters used for the experimental setup and section 5.4 shows the results of this empirical study.

5.1 What is Intransitivity?

An example of a transitive relationship is in the ordering of numbers in a set of integers a = 10, b = 5 and c = 3. For these integers, it can be seen that if a > b and b > c it can be inferred that the relationship of a > c holds. As a result of this transitive relationship, the problem of determining the largest or smallest number in a set of integers is called a transitive problem. In contrast, an example of an intransitive relationship is that of finding a „winning move" in the game rock, paper and scissors. In this game, there is no optimal move that will always win. For example, paper beats rock and rock beats scissors but that does not imply that paper beats scissors; in fact, scissors beats paper. Owing to such relationships, it is not possible to find a superior solution. In this situation both opponents are evolving; a cyclic pathology can result, and the opponents may cycle between the solutions.

As mentioned in section 4.2.1, the assumption was that intransitivity exists in RT. When two teams try to evolve dominant strategies against each other, eventually, the process evolves a counter strategy that has been a solution previously. For example, if the blue team evolves a *strategy A* to counteract the red team's *strategy B*, the red team could evolve a *strategy C* that defeats *strategy A*. Subsequently, the blue team could evolve a *strategy D* in response to *strategy C.* In order to defeat *strategy D*, the red team could evolve another strategy which could be *strategy B* again. Such a reappearance of the same strategy to counter an evolved strategy is a symptom of intransitivity. In such

situations, a cyclic pathology occurs as the competing teams cycle between the previously evolved solutions. RT is a very complex domain in which to test whether the optimization algorithm can address intransitivity. Therefore, in order to see whether the CEA can address intransitivity, a carefully designed test problem was studied.

5.2 Problem Domain

In this section, a problem domain that was chosen to study intransitivity using CEAs is explained. A test problem introduced by Watson and Pollack (2001) was chosen for this study. Researchers have utilized various test problems to analyse the performance of CEAs, for example, Chong et al. (2008, 2009) used Iterated Prisoner‟s Dilemma (IPD) and Rosin and Belew (1997) used the games of Nim and 3-D Tic-Tac-Toe. IPD is a widely studied problem; however, it is an extremely difficult problem for a CEA. The game contains complex evolutionary dynamics and an enormous search space. In fact, researchers always limit their search to solutions that use some restricted representation, such as a finite state machine. The intransitive number problem that was chosen for this study has some advantages over the other test problems mentioned above. It has the specific feature of intransitive relationships that makes the problem difficult, yet it can be represented in a simple manner. The problem is suitable for evaluating CEAs as it is possible to define an objective quality criterion since the theoretical optimum is known.

An intransitive number problem introduced by Watson and Pollack (2001) is adapted into a version with two populations. The problem is symmetric in which both sides are evaluated using the same method. In this problem, individual solutions in both populations consist of pairs of real numbers each ranging from 0 to 100. The score when solution $a = (a_x, a_y)$ from one population competes with solution $b = (b_x, b_y)$ from the other population is given in Equation (5.1). *Score (a, b)* has three possible outcomes: 1 if $a > b$; 0 if $a < b$ and if $a = b$, the score is assigned a value of 1 or 0 randomly.

Consider three solutions, P = <10:90>, Q = <11:88> and R = <8:89>. When the values of P and Q are evaluated according to Equation (5.1), the value of *score* $((a_x, a_y),(b_x, b_y))$ will be score (a_x, b_x) because the condition $|a_x - b_x| < |a_y, b_y|$ is met. Then the value

of score *(a, b)* will be 0 because $a < b$. The basic idea is to check which two of the axes‟ values are closer, and check whether the first axis‟ value is greater than its second value. One example is when score *(a, b)* is 0 (for example, *Q* beats *P*), because 10 and 11 are closer than 90 and 88, so the score is determined by which solution has the larger *x* value. Similarly, *R* beats *Q* (based on *y* being larger than *x*), and yet *P* beats *R*. Thus, the relationships between these solutions are intransitive.

$$score\left((a_x, a_y), (b_x, b_y)\right) = \begin{cases} score(a_x, b_x), & if\,|a_x - b_x| < |a_y - b_y| \\ score(a_y, b_y), & if\,|a_x - b_x| > |a_y - b_y| \\ random\ choice\ of\ above, & if\,|a_x - b_x| = |a_y - b_y| \end{cases} \quad \textbf{(5.1)}$$

$$where, score(a, b) = \begin{cases} 1, & if\ a > b \\ 0, & if\ a < b \\ 0\ or\ 1, & if\ a = b \end{cases}$$

To see how intransitivity could cause convergence problems for a CEA, suppose there are two competing populations and that at some point one population (α) contained a high proportion of *P*-like solutions. This might lead to an increase in the number of *Q*-like solutions in the other population (β). In turn, this would favour *R*-like α solutions, which would then favour *P*-like β solutions. A repeat of a cycle, described above is repeated with *Q*-like α solutions being favoured, followed by, favouring of *R*-like β solutions, finally returning to favouring *P*-like α solutions. This would be a return to the situation that was 6 generations ago. That is, there is the potential for a cyclic dynamic to be established.

5.3 Experimental Setup

This section presents the experimental design and the parameter settings for the experimentation of this problem domain. The intransitive number problem is used to evaluate the performance of a basic competitive CEA and variants (CEAN, CEAFS, CEAHOF and CEACFH), as introduced in chapter 4, in handling intransitivity. The performance of these algorithms is measured via two ways: generalisation performance and objective quality. The generalisation performance tests how well solutions found

for one side in a contest, learned via a CEA, generalise to compete against arbitrary strategies for the other side. In this intransitive number problem, although the superiority relationship is complex, the closer the solution is to the theoretical optimum i.e. <100:100>, the higher quality the solution is, that is the more likely it is to be better than a randomly chosen solution. The method of calculating objective quality of the solution is defined in section 4.5.2. For each algorithm tested, the *mutation rate* was varied from 2.5% to 100% in 2.5% increments. The crossover and mutation operators and values of the parameters used in these experiments are listed in Table 5.1. For each execution, in each generation, diversity (genotypic and phenotypic), generalisation performance (best GP, average GP and ensemble), and objective quality (best and average quality) were calculated. Although Chong et al. (2008, 2009) found the ensemble measure to be more interesting, for this specific problem, the ensemble measure this case provided similar results in every generation for each of the four algorithms. Thus, this measure is excluded for further analysis.

Table 5.1: Parameters used in the experiments

Properties	**Algorithm/Values**
Population size	25 in each population, as recommend by Watson and Pollack (2001)
Gene value	0 to 100
Crossover	Single point
Crossover rate	60%, as recommend by Watson and Pollack (2001)
Mutation	Polynomial
Mutation rate	2.5% to 100% stepwise increments of 2.5%
Selection	Stochastic universal sampling
Generations	300
Number of runs	60
Niche radius	5 (best value suggested by experimentation)
HOF sample size	25 (equal to population size)

Initial gene values were randomly generated values between 0 and 100. The interaction between two individuals produces a score for each individual involved. As the intransitive number problem is symmetric, two competing individuals were evaluated using the same method. The score was calculated using Equations (5.1). As seen in Equation (5.1), the *score (a, b)* has 3 possible outcomes; 1 if $a > b$; 0 if $a < b$ and in the case of $a = b$, the score will be randomly assigned a value of 0 or 1, resulting in non-

deterministic outcomes. To address this, each pair of individuals are evaluated for 20 times and the average over the 20 iterations was considered as an interaction score for each of the individuals involved in the interaction.

For calculating fitness, as shown in Equation (2.1), when the individual is evaluated against all the members from the competing population the fitness values received by an individual are also averaged over the number of individuals in the competing population. In terms of FS, an empirical study was carried out to investigate suitable values for niche radius. The value of 5 was found to be the best one for this domain. Each run of an algorithm was repeated 60 times to allow for statistical variation.

5.4 Results and Analysis

The results of the experiments are analysed using the quality and diversity of the populations that were produced by the CEAN, CEAFS, CEAHOF and CEACFH. The quality of the populations was measured in two ways: using the GP (discussed in section 4.5.1) and the objective quality (section 4.5.2). Diversity of the population was also measured in two ways: the genotypic and phenotypic (sections 4.5.5.1 and 4.5.5.2 respectively). The data for each measure (diversity and quality) were examined in four ways: (1) a profile (interaction) plot of each measure against *mutation rate,* (2) ANOVA test, (3) Correlation analysis for measures associated with each of the four algorithms and (4) Scatter plots are presented to visualise the relationship between quality and diversity.

5.4.1 Evaluation of Four Algorithms via GPs

This section presents an analysis of the quality of two competing populations (evolved by each of the four algorithms) using one of the measures for generalisation performance, the *estimated best GPs*. The two competing populations were named the blue and red respectively. Each of the four algorithms was run 60 times using the parameters shown in Table 5.1. The average of these 60 runs for each of the corresponding 5000 generations is presented as convergence plots depicted in Figure

5.1 when a *mutation rate* of 2.5% was used. The x-axis represents the number of generations and the y-axis represents the *estimated best GPs*.

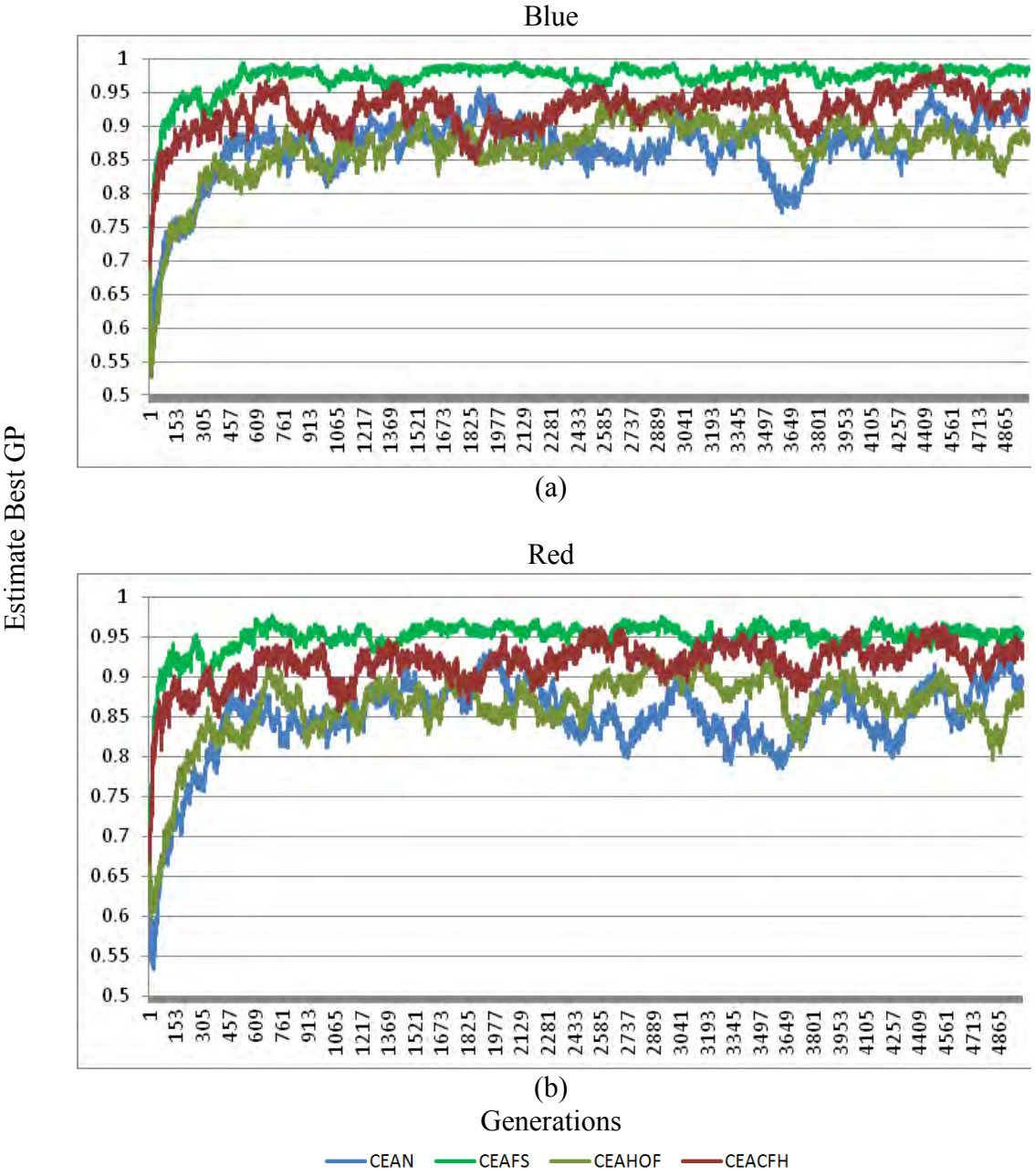

Figure 5.1: Convergence plots showing the estimated best GPs of the four algorithms for (a) the blue and (b) the red team using mutation rates of 2.5%

Figure 5.1 (a) and (b) show that the performance of each of the algorithms in terms of receiving the *estimated best GP* was similar in both, the blue and red, teams. The performance of the solutions associated with all four algorithms suffers from fluctuations. Although CEAFS received the highest *estimated best GP* value in both

teams, there still exists some amount of fluctuation. In comparison to CEAFS, the other three algorithms fluctuated for *mutation rate* of 2.5%. This is presumably due to the intransitive nature of the problem in which each team evolve superior solutions in every generation making it hard to converge at very low *mutation rate*.

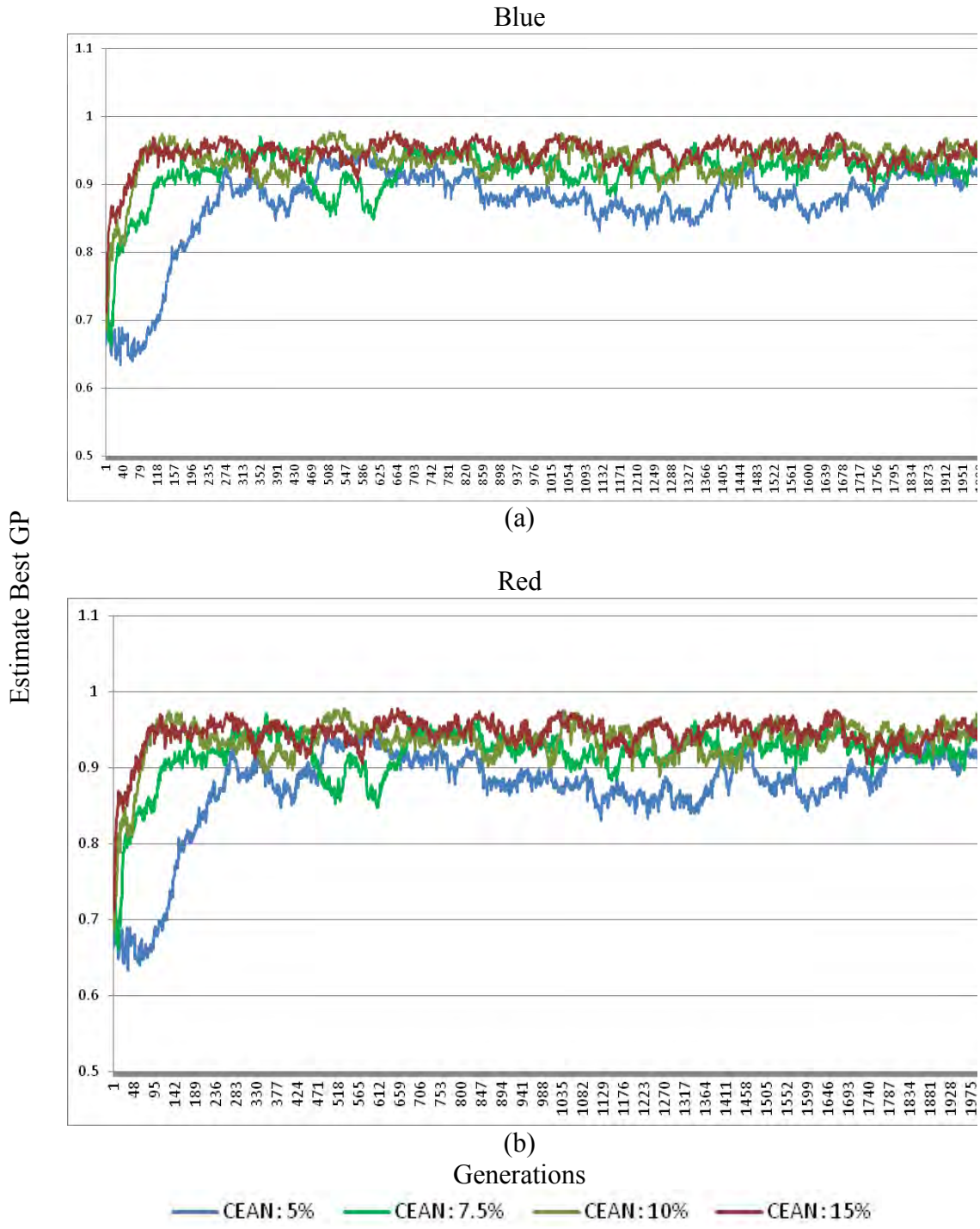

Figure 5.2: Convergence plots showing the estimated best GPs of CEANs for (a) the blue team and (b) red team using mutation rate of 5%, 7.5%, 10% and 15%

Since CEAN, CEAHOF and CEACFH were not converging even after 5000 generations for a *mutation rate* of 2.5%; a further analysis was conducted to evaluate CEAN for a number of mutation rate using the *estimated best GP* and *mutation rate* of 5%, 7.5%, 10% and 15% over 2000 generations. The results are shown in Figure 5.2 in which x-axis represents generations and y-axis the *estimated best GP*. The *estimated best GP* starts to converge in about 100 generations with some fluctuations when the *mutation rate* is 10% and 15%.

Based on the above experimentation each run in subsequent investigations involved 300 generations, as it was shown that these algorithms had relatively stabilised when a *mutation rate* of 10% and above was applied. In order to see the effect of mutation rates, as small, medium and large, mutation rates of 2.5%, 50% and 100% were chosen respectively.

Figure 5.3 (a) and (b) show the plot with a 2.5% *mutation rate*, for both the blue and red team. The solution associated with CEAFS (the green plot) scored the highest *estimated best GP* value with the algorithm starting to converge at around 150 generations. The solution associated with CEACFH achieved the second highest *estimated best GP* value and the algorithm also converged in a similar manner as the CEAFS. These two algorithms incorporate the FS method. The CEAN was third and the CEAHOF achieved the lowest *estimated best GP* value in comparison to the other three algorithms. For both CEAN and CEAHOF, their generalisation performance appeared to improve as the *mutation rate* increases. Throughout this chapter, the phrase "performance of the algorithm" implies the "performance of the evolved solutions generated by the algorithm".

When 50% *mutation rate* was applied, unlike in the case of 2.5% *mutation rate*, the performance of all four algorithms was not very different from each other. These algorithms converged at around 10 generations for both teams (Figure 5.3 (c) and (d)). However, the *estimated best GP* value received by the CEAN and CEAHOF was

slightly higher than two other algorithms. The performance of the CEACFH was relatively low.

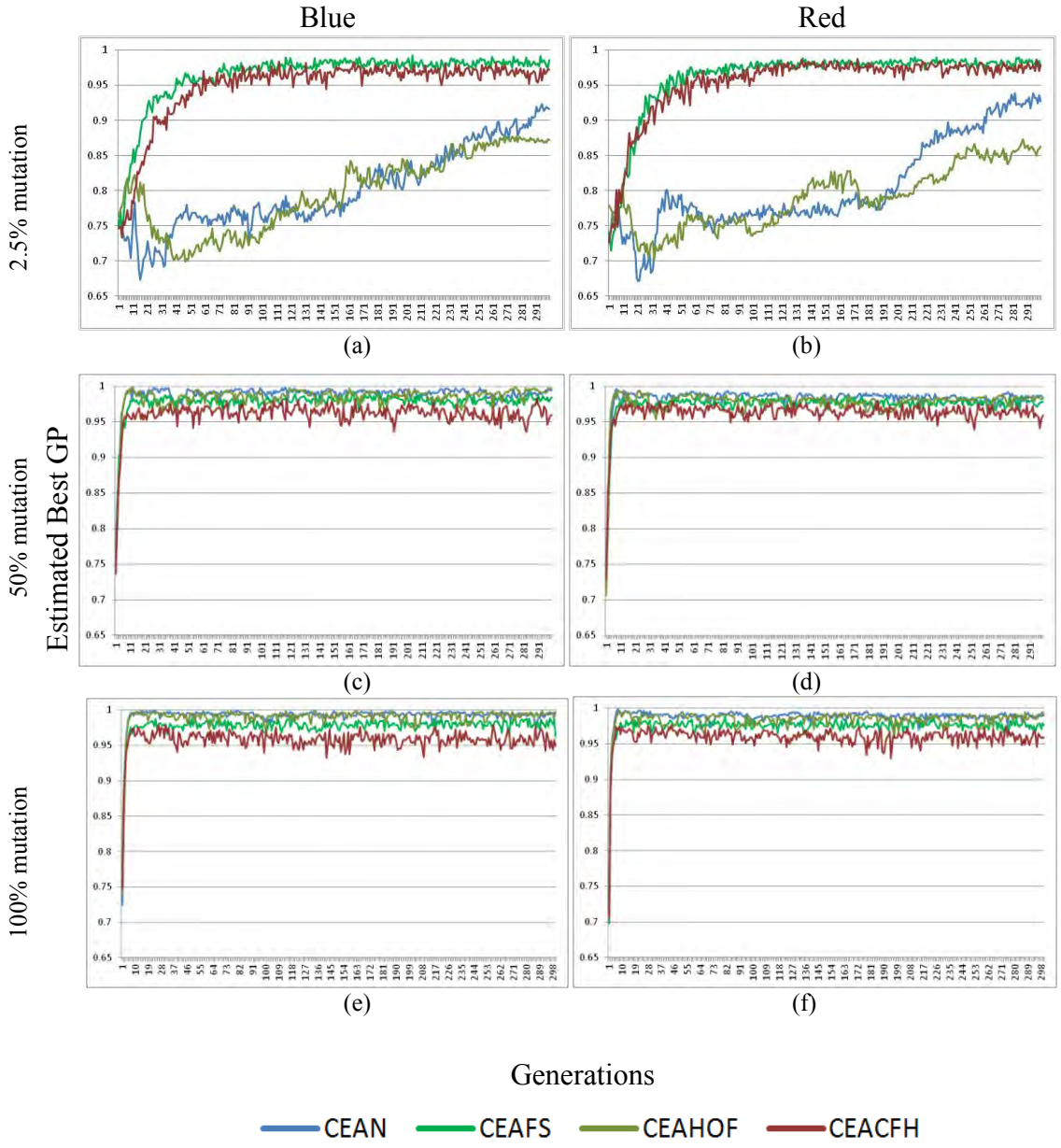

Figure 5.3: Convergence plots showing the estimated best GPs of the four algorithms for the blue and red team using mutation rates of (a) 2.5% blue (b) 2.5% red (c) 50% blue (d) 50% red (e) 100% blue and (f) 100% red

Similar results, as in those associated with 50% *mutation rate*, can also be seen for both teams when *mutation rate* of 100% was applied (Figure 5.3 (e) and (f)). The performance of the CEAN and CEAHOF appeared to be slightly higher than the CEAFS and CEACFH. The graph for CEAFS was relatively smoother but the graph for

CEACFH were noisy, with values fluctuating up and down. The convergence plots for *mutation rates* of 2.5%, 50% and 100% showed that the blue and red teams‟ performances were similar in achieving the *estimated best GP*. This was expected as the intransitive number problem was symmetrical. These similarities imply that it is sufficient to use data from only one team in subsequent analyses.

It was observed that the performance of the four algorithms in terms of the *estimated best GP* were different when using 2.5%, 50% and 100% *mutation rates*. Additionally, Chong et al. (2008, 2009) analysed the performance of algorithms involving various mutation rates to evaluate whether they perform equally well despite varying *mutation rate*. The authors found that algorithms performed differently, when *mutation rate* was varied. Therefore, in order to examine the impact of varying *mutation rate* on *estimated best GP* for each of the four algorithms, 40 different mutation rates ranging from 2.5% to 100% in stepwise increment of 2.5% were used in this study.

Univariate analysis of variance (ANOVA) was conducted using the International Business Machines‟ (IBM) *Statistical package for the social sciences (SPSS)* software version 19. The GP was measured in two different ways: the *estimated average* and *estimated best GP*. Effects of varying *mutation rates* on the *estimated best GP* and *estimated average GP* are explained in sections 5.4.1.1 and 5.4.1.2 respectively.

5.4.1.1 Analysing Estimated Best GP

Figure 5.4 shows the interaction plot in which each of the four algorithms‟ *estimated best GP* was investigated by varying the *mutation rates* from 2.5% to 100% with a stepwise increment of 2.5%. The x-axis represents the *mutation rate* and y-axis represents the *estimated best GP*. For each mutation rate the average of the *estimated best GP* for the final 60 generations is plotted. In CEAN and CEAHOF, there was a rapid increase in the *estimated best GP* between *mutation rate* of 2.5% and 17.5%. However, an increase of *mutation rate* above 17.5% increases the value of *estimated best GP* steadily. This indicated that even the naïve CEA may improve its performance in achieving higher *estimated best GP* values when higher *mutation rate* is applied.

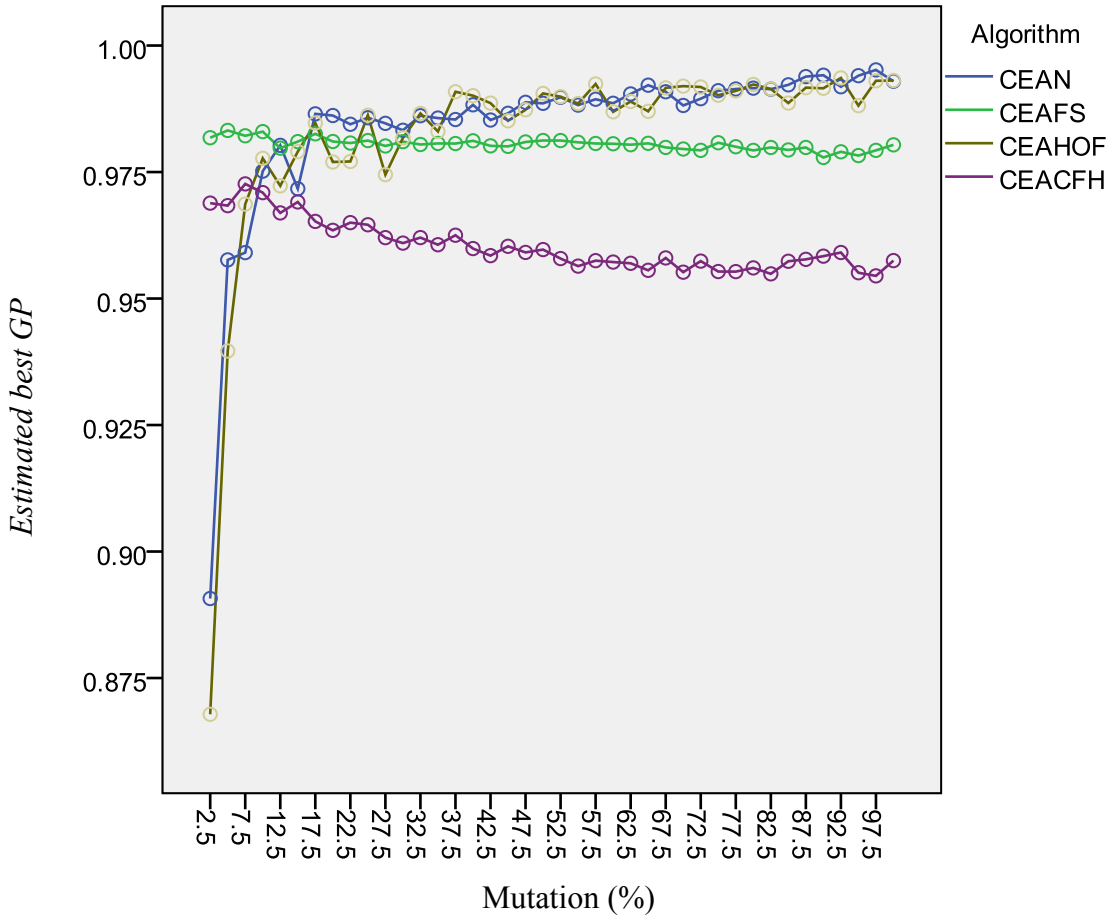

Figure 5.4: Interaction plots of the estimated best GP (mean over the final 60 generations) versus mutation rate for each of the 4 algorithm variants

From the plot, it can be seen that increasing the *mutation rate* generally has no impact on CEAFS. In CEACFH, the value of *estimated best GP* gradually decreased as *mutation rate* increased (i.e. slope of the line passing through the data points have a negative gradient). This requires further investigation to establish the reasons for this effect. The *estimated best GP* associated with CEACFH was relatively low compared to that in the CEAFS. Intuitively, it was expected that the HOF might combat intransitivity by preserving past solutions in its archive and FS might combat by ensuring that a mix of diverse solutions remain in the population, the combination of these two techniques might further improve the performance of algorithms. However, the result indicated that the combination seems to be negating each other. FS aims to increase diversity of the population and at the same time the HOF decreases diversity. Additionally, it was noticeable that the effect on each of the four algorithms is relatively small for increasing

mutation rate above 17.5 % as indicated by the respective gradients of lines ("best fitted" over the points) being close to zero.

Two-way ANOVA was conducted to analyse *estimated best GP* with 40 levels of *mutation rate* and *algorithm used*. The overall model and all effects were statistically significant. The interaction effect was ($F(117, 9440) = 5.775$, $p < 0.05$, Partial Eta Squared = 0.067). The table associated with ANOVA is depicted in Appendix A.1.

5.4.1.2 Analysing Estimated Average GP

This section presents the analysis of the *estimated average GP* of the populations, produced by CEAN, CEAFS, CEAHOF and CEACFH when the same series of experiments as those for *estimated best GP* were conducted. An interaction plot of the mutations rates and the *estimated average GP*s are presented in Figure 5.5. In CEAN, the *estimated average GP*s increases over first few mutation rates and subsequently fluctuates within the band of 0.83 to 0.88. The CEAHOF was also similar to CEAN but the average GP value gradually decreases after 50% *mutation rate*. The CEAHOF scored the highest value for *estimated average best GP* when the *mutation rates* range from 7.5% to 37.5%. Unlike these two algorithms, the CEACFH starts with the highest *estimated average GP value* at 2.5%. The *estimated average GP* associated with CEAFS decreases when *mutation rate* of 2.5% to 17.5% was applied and remained approximately at the same value at *mutation rates* higher than 17.5%. The slope of CEAFS and CEACFH flatten out when mutation rate is 30% and above, indicating that increasing *mutation rate* seems to have little impact on the GP of each of these two algorithms.

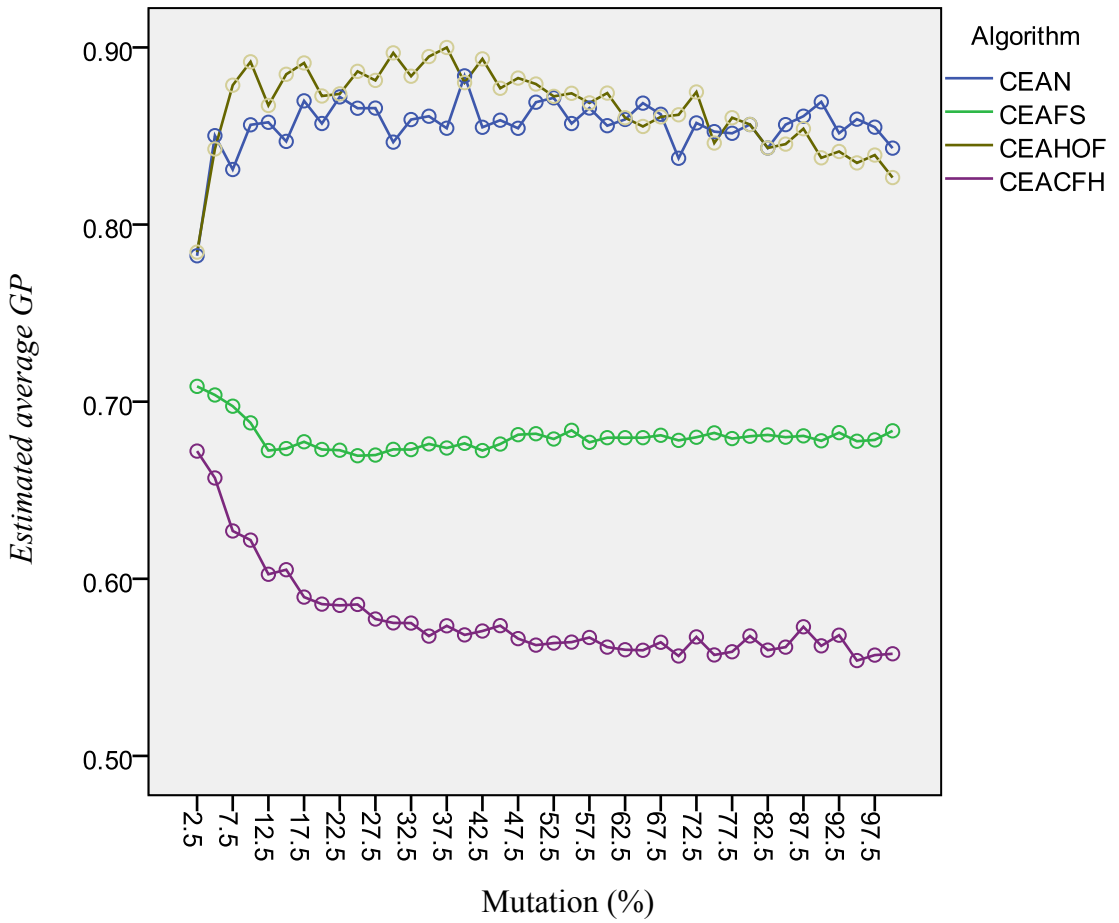

Figure 5.5: Interaction plots of the estimated average GP (mean over the final 60 generations) versus mutation rate for each of the 4 algorithm variants

Two-way ANOVA was conducted to analyse *estimated average GP* with 40 levels of *mutation rate* and four types of *algorithm used*. The overall model and all effects were statistically significant. The interaction effect was F ((117, 9440) = 6.216, $p < 0.05$, Partial Eta Squared = 0.072). The table associated with ANOVA is depicted in Appendix A.2.

On the basis of the analysis involving GPs, the above results showed that the CEAHOF and CEAN were equally good algorithms when used with higher *mutation rate* to address the intransitive number problem. This indicated that the integration of the HOF could enhance the capacity of the naïve CEA. Likewise, the use of higher *mutation rate* in the naïve CEA helped to increase the performance of the algorithm in scoring higher GPs value. After the evaluation of *estimated best GP,* the following section presents the

evaluation of a naïve CEA and three variants in terms of objective quality, a measure of how close the evolved solution is to the theoretical optimum.

5.4.2 Evaluation of Algorithms via the Objective Quality

This section explores the performance of the algorithms using the objective quality. Chong et al. (2008, 2009) argued that performance of algorithms could be enhanced by applying a suitable *mutation rate*. This evaluation of the performance in terms of the *objective best quality* and the *objective average quality* (as explained in section 4.5.2) are described below.

5.4.2.1 Analysis of the Objective Best Quality

This section describes the use of *objective best quality* in evaluating the performance of CEAN, CEAFS, CEAHOF and CEACFH when *mutation rate* was varied. An interaction plot of the *algorithms used* versus different *mutation rate* is presented in Figure 5.6 to show the effect of varying *mutation rate* in these algorithms. Increasing *mutation rate* adversely affects the performance of CEACFH, with a negative gradient for a line "best fitted" over the points. The *objective best quality* oscillates up and down with increasing *mutation rate* but there was still an upward trend for the CEAN and CEAHOF. These algorithms managed to produce relatively higher values for the *objective quality* at a higher *mutation rate*. CEAFS received the highest value of *objective best quality.* The slope associated with the points for CEAFS has a small positive gradient.

When evaluating the *estimated best GP* and *objective best quality*, Figure 5.4 and Figure 5.6, similar trends for the CEAFS and CEACFH can be seen. The performances of these algorithms are similar in both measurements. However, the performance of CEAN and CEAHOF is highly different in these two different measures. The *estimated best GP* associated with CEAN and CEAHOF rapidly increased initially (*mutation rates* of 2.5% to 10%) with some small fluctuations in the GP values at the higher *mutation rates*. However, these algorithms were not very effective to achieve higher *objective best quality*. The performance of the CEAHOF produced values that showed large

fluctuations. Figure 5.6 also shows that even in high mutations, the objective quality of the CEAN and CEAHOF remained far below than the best quality of the CEAFS.

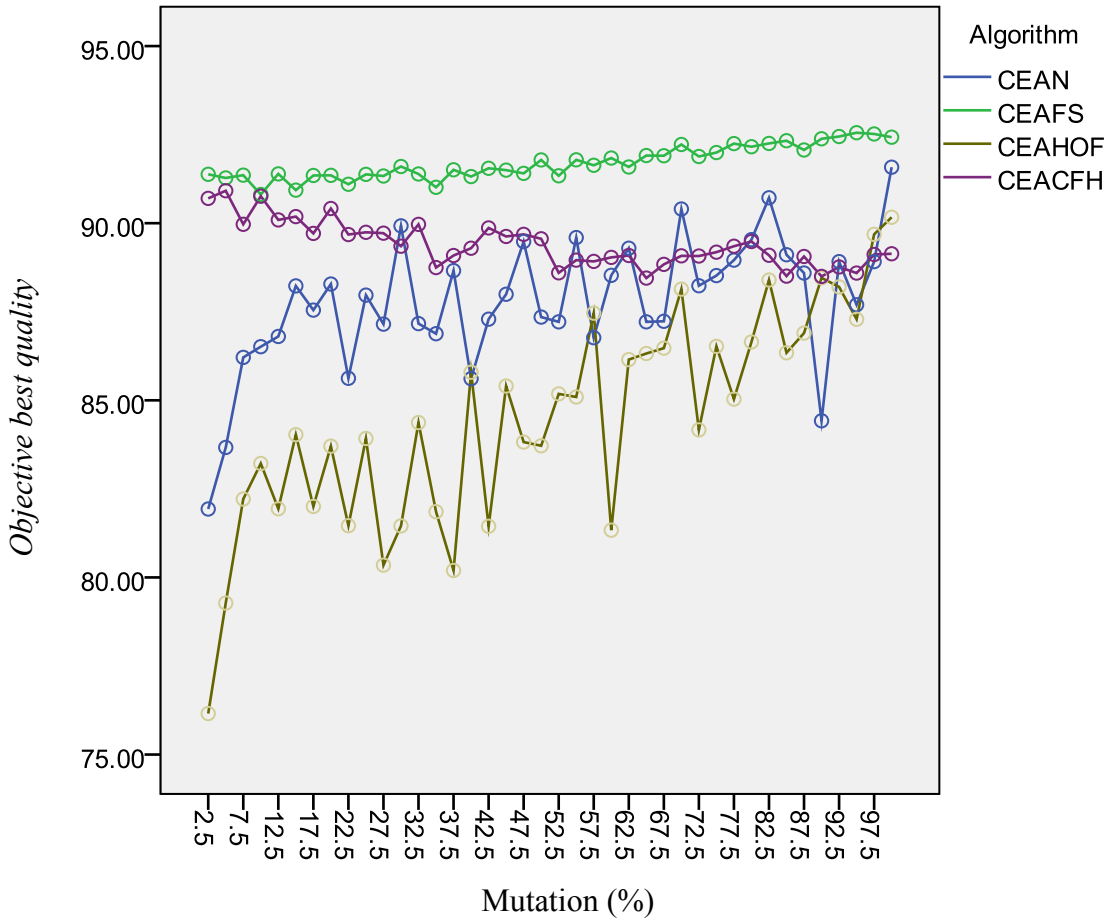

Figure 5.6: Interaction plots of the objective best quality (mean over the final 60 generations) versus mutation rate for each of the 4 algorithm variants

Two-way ANOVA was conducted to analyse *objective best quality* with 40 levels of *mutation rate* and four types of *algorithm used.* The overall model and all effects were statistically significant. The interaction effect was $F\ (117, 9440) = 2.391$, $p < 0.05$ (Partial Eta Squared = 0.029). The table associated with ANOVA is depicted in Appendix A.3.

5.4.2.2 Analysis of the Objective Average Quality

This section investigates the impact on the *objective average quality* of the populations from employing the CEAN, CEAFS, CEAHOF and CEACFH as *mutation rate* changes. The interaction plots indicating the *objective average quality* in these algorithms are

depicted in Figure 5.7. In the CEAN and CEAHOF, the *objective average quality* varies within a band value of 72-79 with a slight upward trend as *mutation rate* increases. In CEAFS, a lower *mutation rate* gave relatively higher *objective average quality* which dropped over the initial few values of *mutation rate* and again slowly increased as the *mutation rate* increased. In the CEACFH, the overall trend was that *objective average quality* decreases as *mutation rate* increased with some small fluctuations in the objective average quality values.

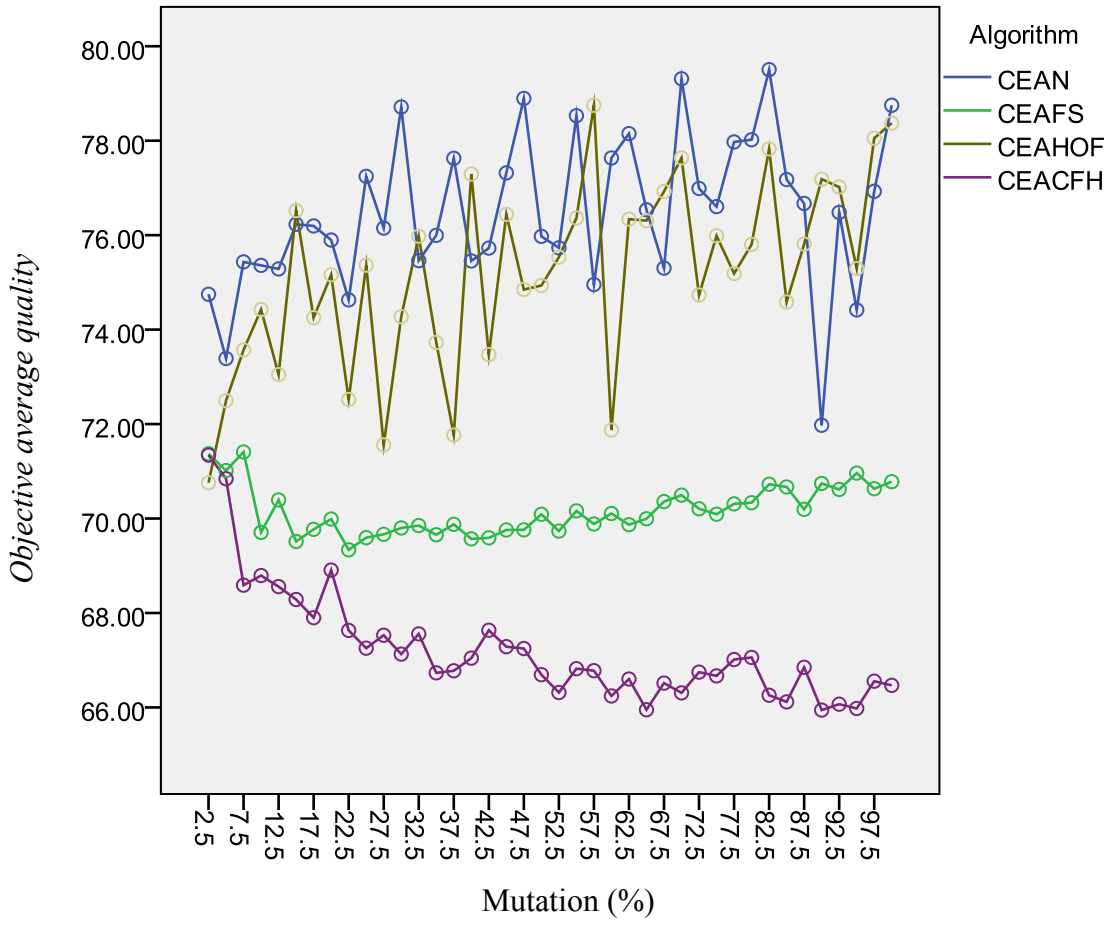

Figure 5.7: Interaction plots of the objective average quality (mean over the final 60 generations) versus mutation rate for each of the 4 algorithm variants

Two-way ANOVA was conducted to analyse *objective average quality* with 40 levels of *mutation rate* and four types of *algorithm used.* The overall model was statistically significant. The interaction effect was $F(117, 9440) = 1.564$, $p < 0.05$ (Partial Eta Squared = 0.019). The table associated with ANOVA is depicted in Appendix A.4

The above analysis showed the evaluation of the performances of each of the four algorithms in terms of the *objective quality*. Specific to this problem domain, the result indicated that the performance of CEAFS (when measured using the *objective best quality*) appeared to be the best out of the four algorithms tested in terms of evolving solutions that approach the theoretical optimum. Higher *mutation rate* helped to enhance the performance of algorithms except for CEACFH where increasing *mutation rates* had an adverse effect.

When evaluating the *objective best quality* and *objective average quality*; the performance of CEAFS, in terms of the *objective average quality*, was relatively poor. This was expected, as the average calculation involved all individuals from the population and the population may include individuals that are not so optimal in order to maintain diversity

Interestingly, each of the four algorithms performance on the basis of the *estimated GP* (performance of solutions when competing against arbitrary strategies) and objective quality (the theoretical optimum) was different. The CEAN and CEAHOF appeared to be a high achiever with higher *mutation rate* when the performance of the algorithms was measured in terms of *estimated GP*. However, these algorithms were not so effective in comparison to CEAFS when the algorithm‟s performance was measured on the basis of the *objective quality*. In addition, the above analysis suggests that an appropriate *mutation rate* can influence the performance of all four algorithms. Chong et al. (2009) also stressed the importance of diversity in order to enhance the performance of the algorithms. Therefore, experiments were conducted to investigate how diverse the evolved populations from these algorithms were when different values of *mutation rate* were applied. The following section presents details of this experiment.

5.4.3 Analysis of Diversity of the Population

In order to investigate diversity of the population associated with all four algorithms employed in this study, diversity was measured here in two ways: genotypic and phenotypic. This section presents the analysis of each of these diversity measures in sections 5.4.3.1 and 5.4.3.2 respectively.

5.4.3.1 Analysis of Genotypic Diversity

This section presents the analysis of *genotypic diversity* when *mutation rate* increased from 2.5% to 100% in stepwise increments of 2.5%. The *genotypic diversity* was measured according to the steps discussed in Section 4.5.5 against a fixed test population (see Table 4.4). The interaction plot for *genotypic diversity* versus *mutation rate* is depicted in Figure 5.8. Each point in the plot represents a mean value of 60 runs of experiment in which the averages of the last 60 generations were calculated. The *x* and *y* axes represent *mutation rate* and *genotypic diversity* respectively.

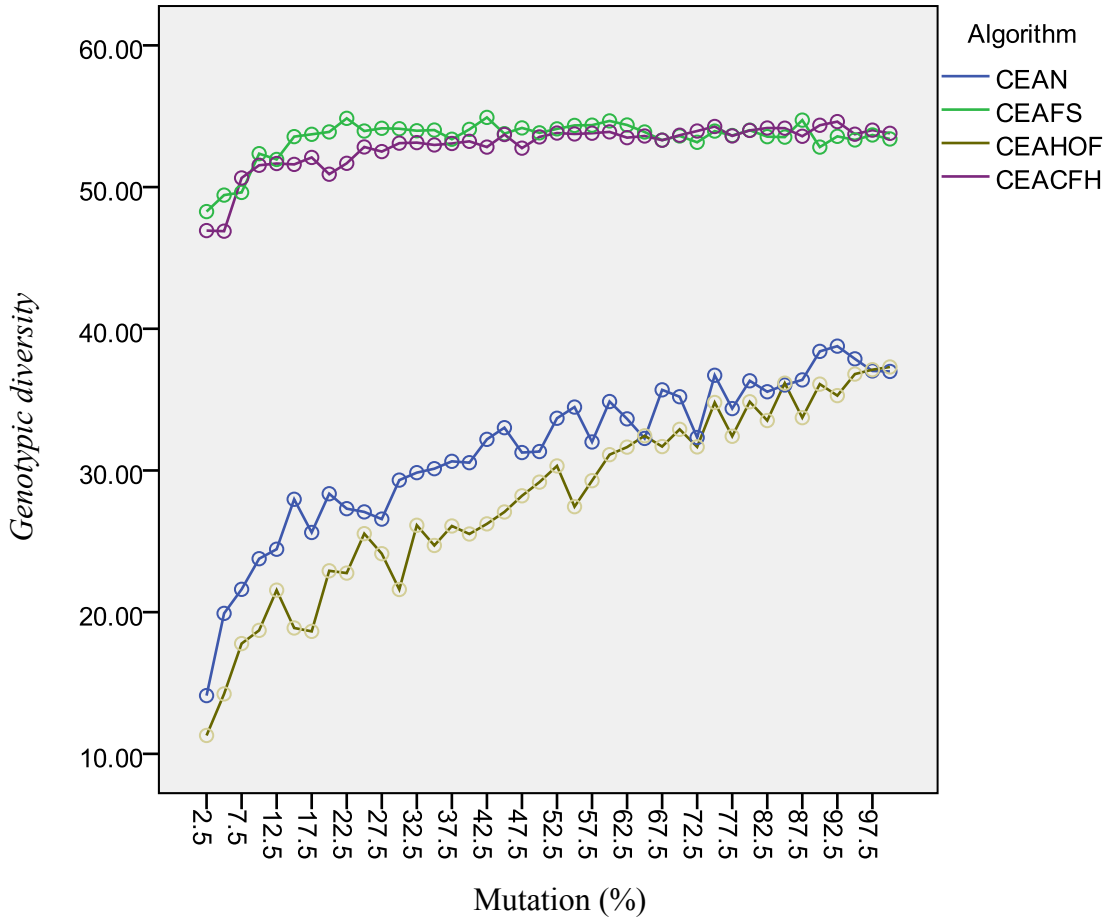

Figure 5.8: Interaction plots of genotypic diversity (mean over the final 60 generations) versus mutation rate for each of the 4 algorithm variants

For CEAN and CEAHOF, *genotypic diversity* increases as *mutation rate* increases, with some oscillations. However, the diversity of the evolved population from these algorithms did not reach the same level of the CEAFS and CEACFH. The *genotypic diversity* increases as *mutation rate* increased from 2.5% to approximately 20% in

CEAFS and CEACFH. Increasing the *mutation rate* beyond 20% appears to have little effect on increasing the *genotypic diversity,* as the slope, associated with the points for these two algorithms respectively, flatten out thus indicating that increasing *mutation rate* seems to have little impact on the *genotypic diversity* for each of these two algorithms. This may be due to the FS method that was integrated in these two algorithms which was an effective means of increasing diversity in the populations.

Two-way ANOVA was conducted to analyse *genotypic diversity* with 40 levels of *mutation rates* and four types of *algorithm used*. The overall model and all effects were statistically significant. The interaction effect was ($F(117, 9440) = 11.77$, $p < 0.05$, Partial Eta Squared = 0.127). The table associated with ANOVA is depicted in Appendix A.5.

It was noticed that the populations evolved by CEAFS and CEACFH were highly diverse in comparison to the populations evolved by CEAN and CEAHOF. It appeared to be fitness sharing is a good technique to maintain diversity in the population. However, the population associated with CEAN and CEAHOF maintain diversity well at higher *mutation rate*. When evaluating interaction plots, it seems that an increase in genotypic diversity increases quality of the populations (*best GP* and *objective best*) associated with CEAN and CEAHOF. In higher *mutation rate*, *genotypic diversity* also increases as well as the *best GP* and *objective best quality* for these two algorithms. However, for the populations associated with CEAFS and CEACFH, it seems like there is no impact of increasing genotypic diversity on their performance. Further analysis is required to evaluate the relationship between diversity and quality of the solutions and this will be described in section 5.4.4.

5.4.3.2 Analysis of Phenotypic Diversity

A similar analysis to that described in section 5.4.3.1 was conducted to analyse *phenotypic diversity* in all four algorithms. The steps involved in calculating *phenotypic diversity* were described in 4.5.2. As in the case of quality measures (GPs and objective quality) and other diversity measure (genotypic diversity), the phenotypic diversity was also calculated for each evolving population in every generation and the last 60 generations were averaged from each run. Note that there were 60 repetitions for every

algorithm execution at a specific *mutation rate*. The average of these 60 runs was taken into consideration for evaluation. The interaction plot between the *mutation rate* and *algorithms used* is shown in Figure 5.9. The *x* and *y* axes represent *mutation rates* and *phenotypic diversity* respectively.

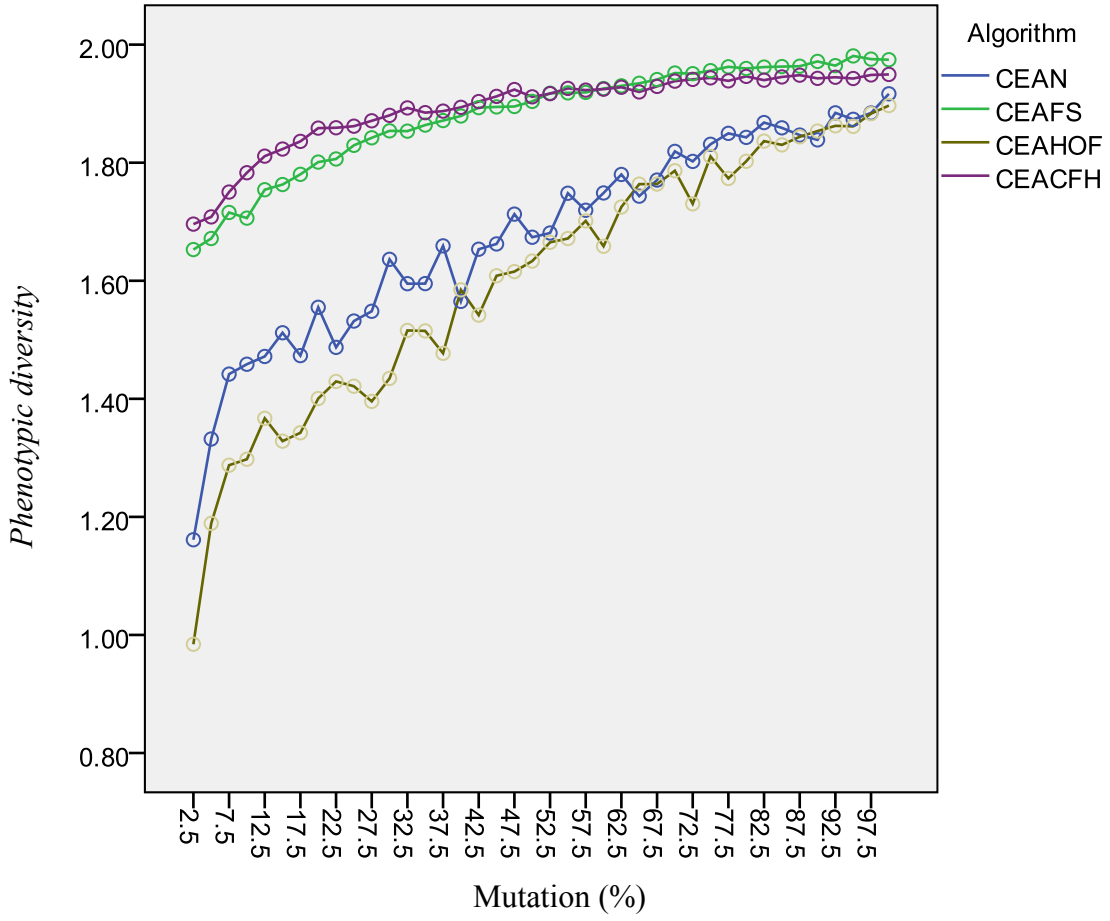

Figure 5.9: Interaction plots of phenotypic diversity (mean over the final 60 generations) versus mutation rate for each of the 4 algorithm variants

Figure 5.9 demonstrates that an increase in *mutation rate* also increased *phenotypic diversity* of the evolved populations associated with all four algorithms. In each of these algorithms, the increase was rapid for low *mutation rate*. However, for higher *mutation rate*, the increase was relatively slower in the CEAFS and CEACFH. The slopes associated with CEAFS and CEACFH are similar. The *mutation rate* continued to increase the *phenotypic diversity* at a consistent rate in the CEAN and CEAHOF. FS introduced higher *phenotypic diversity*, as was shown for *genotypic diversity* (section 5.4.1.1). In the CEAN and the CEAHOF, there was less diversity in small mutations but

diversity rapidly increased with the increase of *mutation rate*. The slopes associated with CEAN and CEAHOF are similar.

Two-way ANOVA was conducted to analyse *phenotypic diversity* with 40 levels of mutation rates and four types of *algorithm used*. The overall model and all effects were statistically significant. The interaction effect was ($F(117, 9440) = 13.985$, $p < 0.05$, Partial Eta Squared = 0.148). The table associated with ANOVA is depicted in Appendix A.6.

A point to note is that by examining Figure 5.8 and Figure 5.9, it can be seen that the plots for *genotypic* and *phenotypic diversity* are quite similar. The effect of *mutation rate* on CEAFS and CEAHOF was relatively higher when measuring *phenotypic diversity*. An increase in *mutation rate* increases both *genotypic* and *phenotypic diversity;* however, increase was slight in *genotypic diversity* for CEAFS and CEACFH. In order to further analyse the relationship, correlation analysis is conducted. Chong et al. (2008, 2009) also analysed the relationship between the diversity of a population with its GP and found that diversity maintained by the implicit and explicit method in the population highly influenced the GP. Bearing that in the mind, relationship between diversity and quality of the evolved populations produced by all four algorithms are analysed in the following section.

5.4.4 Relationship between Diversity, GPs and Objective Quality

To examine the relationship between variables involved in this study, correlation analysis was conducted. These variables were diversity measures (*genotypic* and *phenotypic diversity*), estimated GPs (*estimated average GP* and *estimated best GP*, *objective average quality* and *objective best quality*) and *mutation rate*. The correlations between these variables in the CEAN, CEAFS, CEAHOF and CEACFH are presented in Table 5.2, Table 5.3, Table 5.4 and Table 5.5 respectively.

Table 5.2: Correlation between factors in CEAN algorithm

	Genotypic	Phenotypic	Avg_GP	Best_GP	Obj_Avg	Obj_Best	Mutation
Genotypic	1						
Phenotypic	**.854****	1					
Avg_GP	**.127****	**.151****	1				
Best_GP	**.292****	**.297****	**.693****	1			
Obj_Avg	**.057****	**.043***	**-.347****	**.150****	1		
Obj_Best	**.122****	**.108****	**-.336****	**.231****	**.948****	1	
Mutation	**.513****	**.382****	.034	**.191****	**.042***	**.097****	1

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

N= 2400

Table 5.2 shows that *genotypic* and *phenotypic diversity* measures were positively correlated. The mutation was also positively correlated with *genotypic diversity* which supports the result presented in Figure 5.8. In addition, both these diversity measures were positively correlated with the *estimated best GP* and also with the *objective best quality.* Although, the correlations were not very strong this indicated that this is a link between obtaining solutions with good generalisation performance and those that are closer to the theoretical optimum <100,100>. Unexpectedly, the *estimated average GP* appeared to be negatively correlated with *objective average quality* and *objective best quality*. This requires further investigation to explore the reason of this effect. On the other hand, mutation is positively correlated with all the variables, supporting some of the finding from the interaction plots.

The correlation between variables for the CEAFS was depicted in Table 5.3. There was a positive but weak correlation between *genotypic* and *phenotypic diversity*; however, their relationship was significant at $p < 0.01$. These diversity measures are positively correlated with the *estimated best GP*. This indicated that an increase in diversity increases this quality measure. The correlation between the *estimated best GP* and *objective best quality* was also significant but weakly positive (0.048). *Mutation rate* has a negative correlation with all the variables except for *objective best quality*, showing that increasing *mutation rate* for CEAFS did not effect as for the case of CEAN where in the previous table it can be seen that it has a positive correlation with all variables.

Table 5.3: Correlation between factors in CEAFS algorithm

	Genotypic	Phenotypic	Avg_GP	Best_GP	Obj_Avg	Obj_Best	Mutation
Genotypic	1						
Phenotypic	**.111****	1					
Avg_GP	-.010	.032	1				
Best_GP	**.043***	**.059****	**.463****	1			
Obj_Avg	**-.065****	-.005	**.360****	**-.117****	1		
Obj_Best	**-.059****	**-.126****	**.120****	**.048***	**.712****	1	
Mutation	**-.139****	**-.584****	-.034	**-.123****	**-.060****	**.245****	1

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

N= 2400

Table 5.4: Correlation between factors in CEAHOF algorithm

	Genotypic	Phenotypic	Avg_GP	Best_GP	Obj_Avg	Obj_Best	Mutation
Genotypic	1						
Phenotypic	**.388****	1					
Avg_GP	.082	**.141****	1				
Best_GP	**.250****	**.243****	**.710****	1			
Obj_Avg	**.056****	**.042***	**-.235****	**.211****	1		
Obj_Best	**.107****	**.075****	**-.279****	**.248****	**.954****	1	
Mutation	**.458****	**.166****	**-.110**	**.181****	**.084***	**.183****	1

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

N= 2400

The correlation between variables for the CEAHOF was depicted in Table 5.4. The correlation between two diversity measures was positive and stronger than in the CEAFS. In addition, both these diversity measures have positive correlation with the *estimated best GP* and *estimated average GP* and also with the *best objective* quality and *average objective quality*. Unlike the case with CEAFS, there was a positive correlation between the *estimated best GP* and *objective best quality*. Interestingly, the correlation between the *objective average quality* and *estimated average GP* was negative; a fact not reflected for the related measures associated with "best quality" as

the *objective best quality* and *estimated best GP* were positively correlated. Similar to CEAN, both diversity measures are positively correlated with the quality measures and mutation is also positively correlated to all the other variables except the *estimated average GP*.

Table 5.5: Correlation between factors in CEACFH algorithm

	Genotypic	Phenotypic	Avg_GP	Best_GP	Obj_Avg	Obj_Best	Mutation
Genotypic	1						
Phenotypic	**.045***	1					
Avg_GP	**-.054****	**.270****	1				
Best_GP	-.025	**.206****	**.632****	1			
Obj_Avg	-.036	**.185****	**.512****	**.097****	1		
Obj_Best	-.032	**.133****	**.309****	**.128****	**.791****	1	
Mutation	**-.056****	**-.612****	**-.408****	**-.294****	**-.302****	**-.185****	1
**. Correlation is significant at the 0.01 level (2-tailed).							
*. Correlation is significant at the 0.05 level (2-tailed).							
N= 2400							

The correlation between variables for the CEACFH is depicted in Table 5.5. The correlation between genotypic and *phenotypic diversity* was positive but weak at 0.045. *Genotypic diversity* was negatively correlated with the *estimated best* and *average GP* and also with the *objective best quality* and *objective average quality*. However, *phenotypic diversity* was positively correlated with the *estimated average GP* and *best GP* and also with the *objective average quality* and *objective best quality*. Similar to CEAFS, *mutation rate* has a negative correlation with all the variables (including *objective best quality)*.

The correlation analysis shows that *mutation rate* was positively correlated with *genotypic* and *phenotypic diversity* for CEAN and CEAHOF. Interestingly, the *mutation rate* was negative correlated with both diversity measures. The reason may be that fitness sharing already maintained diversity and when mutation changed the genome, it generated solution more similar to each other.

As depicted in the interaction plots, the correlation between *genotypic* and *phenotypic diversity* was positive in all four algorithms; however, it was strong only in CEAN,

moderate in CEAHOF and weak in CEAFS and CEACFH. This was expected as the algorithms with FS were already diverse and there was a low space for further diversity, in terms of both genotypic and phenotypic.

Another interesting relationship was in between *genotypic diversity* and *estimated best GP*, the correlation between them was positive except in CEACFH. The correlation was weak in CEAFS and moderate in the other two algorithms. The expectation was that an increase in diversity increases quality but the results show that there was no strong correlation between diversity and quality. However, diversity helps to improve quality. Unlike the presumption made on the basis of interaction plot in section 5.4.2.1, the correlation between two quality measures (estimated best GP and objective best quality) was positive. This indicated that the solution close to the theoretical optimum can receive higher score against the arbitrary strategy (fixed population set).

5.4.4.1 Genotypic Diversity and Estimated Best GP

In order to further the examination on the relationships between *genotypic diversity* and *estimated best GP* and also to visualize the information, Figure 5.10 depicts scatter plots for each of four algorithms used in this thesis. The x-axis and y-axis represent the *genotypic diversity* and *estimated best GP* respectively. The x value of each data point is the mean value of *genotypic diversity* over the last 60 generations from each run with a specific *mutation rate*. There were 40 variations of *mutation rate* in this experiment. Each of the four algorithms was executed for 60 times with a specific *mutation rate*. Similarly, the y value of each point is the corresponding mean value of *estimated best GP*. Therefore there are 60 x 40 = 2400 points in each algorithm.

The relationship between the *estimated best GP* and *genotypic diversity* is shown in Figure 5.10. The scatter plots for CEAFS and CEACFH were similar in nature which was also shown in the interaction plots associated with each of the four algorithms and also from the correlation analysis. The points are concentrated in a region bounded by the *estimated best GP* value around 1.0 and the *genotypic diversity* in the range of approximately 0.3 to 0.4 which indicates that neither *genotypic diversity* nor the *estimated best GP* changes a lot based on *mutation rate*.

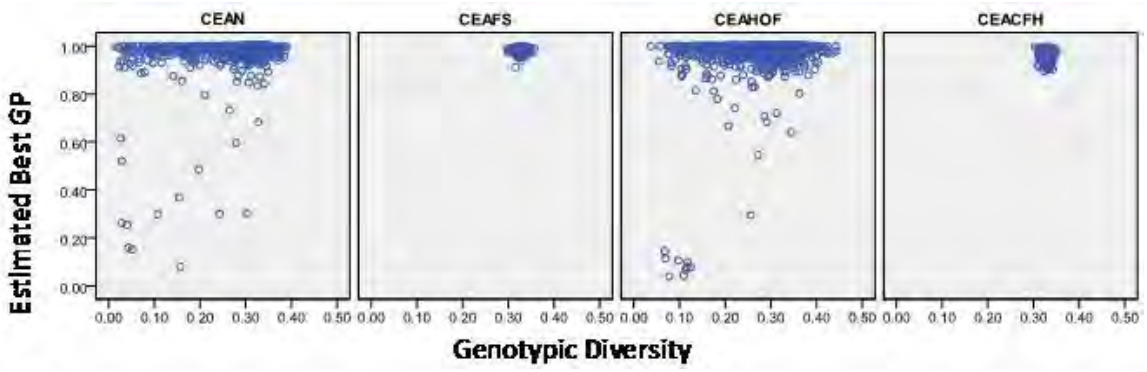

Figure 5.10: Scatter plot of estimated best GP versus genotypic diversity for the CEAN, CEAFS, CEAHOF and CEACFH algorithms. Each point is a mean value of particular mutation rate and there were 40 mutation variations, there were 60 runs for each mutation rate.

In contrast, the scatter plots for CEAN and CEAHOF show that despite changing *genotypic diversity*, the *estimated best GP* remained less changed. In both cases the points are mostly concentrated in a region bounded by the *estimated best GP* value ranging between 0.8 to 1.0 and the *genotypic diversity* in the range of approximately 0.05 to 0.4. Scattered points at the bottom part of the associated charts also indicated that sometimes the population received low *estimated best GP* and low *genotypic diversity* which did not occur in the CEAFS and CEACFH. The plots show that there was no strong correlation between the *estimated best GP* and *genotypic diversity* for all four algorithms.

5.4.4.2 Phenotypic Diversity and Estimated Best GP

Figure 5.11 shows the scatter plot for the *phenotypic diversity* and *estimated best GP* for these algorithms. The x-axis and y-axis represent *phenotypic diversity* and *estimated best GP* respectively. Similar to Figure 5.10, each point represents an interaction between *phenotypic diversity* and *estimated best GP*. In the CEAFS and CEACFH, scatter chart shows that the data points are concentrated in the upper right-hand quadrant. The *estimated best GP* is in the range of 0.9 to 1.0 and the *phenotypic diversity* ranges approximately from 1.4 to 2.0. They appeared to be positively correlated. A point to notice was that both these algorithms received the highest *phenotypic diversity*. In the CEAN and CEAHOF, the scatter plot shows that most of the data are mostly concentrated around a region where the *estimated best GP is 1.0 and phenotypic diversity* is between 0.3 and 1.9. In the case of CEAN, there are a small

number of points scattered randomly and in CEAHOF, there is also a small number of points with low *estimated best GP* values concentrated at the bottom of the chart.

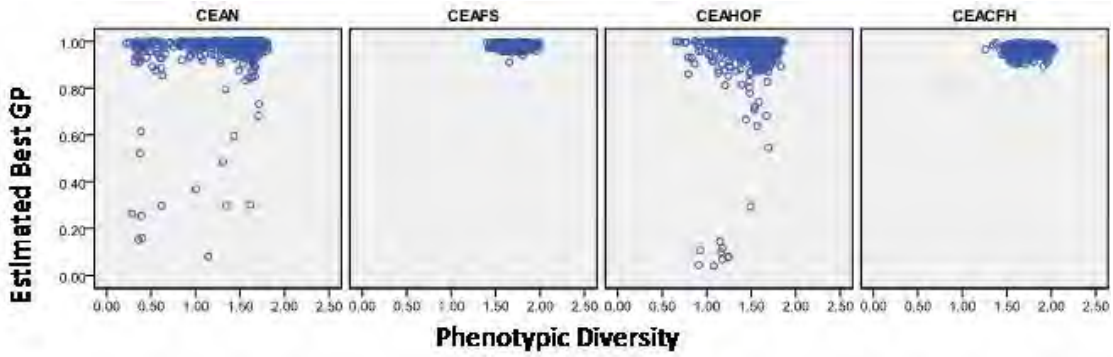

Figure 5.11: Scatter plot of estimated best GP versus phenotypic diversity for the CEAN, CEAFS, CEAHOF and CEACFH algorithms

5.4.4.3 Genotypic Diversity and Objective Best Quality

Figure 5.12 shows the relationship between *genotypic diversity* and the *objective best quality* in each of the four algorithms studied in this thesis. In all four algorithms, the data points appear to be concentrated towards the upper right-hand quadrant (i.e. high *objective best quality* and *genotypic diversity*). The points associated with CEAFS and CEAFCH are located densely in a very small region whereas the points are much more spread out for CEAN and CEAHOF with *objective best quality* ranging from 60 to 100 and *genotypic diversity* from 0.05 to 0.4. There are also many points having less *genotypic diversity* and high *objective best quality*.

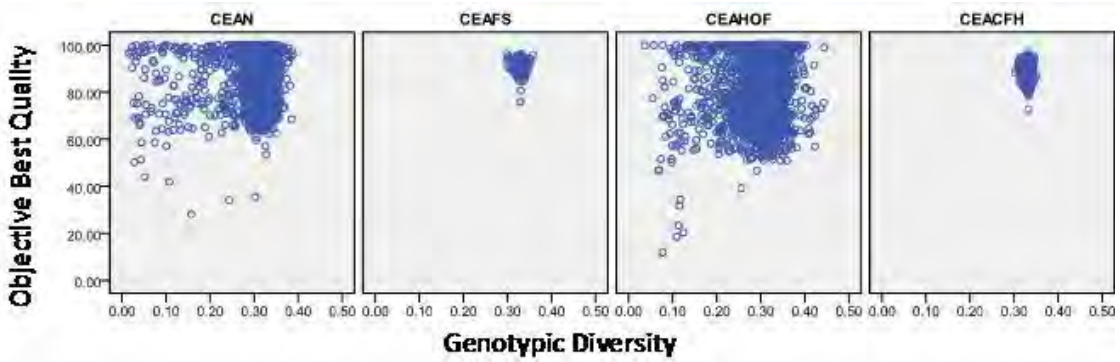

Figure 5.12: Scatter plot of objective best quality versus genotypic diversity for the CEAN, CEAFS, CEAHOF and CEACFH algorithms

5.4.4.4 Phenotypic Diversity and Objective Best Quality

Figure 5.13 shows the effect of *phenotypic diversity* on the *objective best quality*. The x-axis and y-axis represent *phenotypic diversity* and *objective best quality* respectively. The scatter plots here are quite similar (except for the area of the regions being less spread out along the *phenotypic diversity* axis) to those in the previous section which examined the relationship between *genotypic diversity* and *objective best quality*.

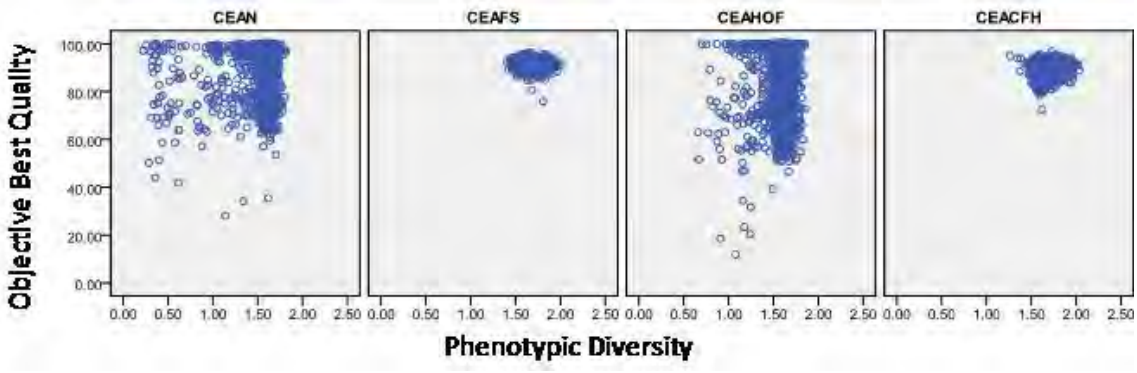

Figure 5.13: Scatter plot of objective best quality versus phenotypic diversity for the CEAN, CEAFS, CEAHOF and CEACFH algorithms

5.4.4.5 Genotypic Diversity and Estimated Average GP

Figure 5.14 presents the relationship between *genotypic diversity* and *estimated average GP*. The x-axis and y-axis of the chart represent *genotypic diversity* and *estimated average GP* respectively. For CEAN and CEAHOF, the data points appear to be concentrated towards the upper half quadrant (i.e. high estimated average GP) with *genotypic diversity* ranging from 0.05 to 0.4. The points associated with CEAFS and CEACFH are located densely in a very small region whereas the points are much more spread out for CEAN and CEAHOF with *objective best quality* ranging from 60 to 100 and *genotypic diversity* from 0.05 to 0.4. In the case of CEAN, there are a small number of points scattered randomly and in CEAHOF, there is also a small number of points concentrated at the bottom left-hand corner of the chart (i.e. with low *estimated average GP* and low *genotypic diversity* values). In the CEAFS and CEACFH, the points are lined concentrated around *genotypic diversity* of 0.34 and *estimated average GP* ranging from 0.5 to 0.9 and from 0.4 to 0.9 respectively.

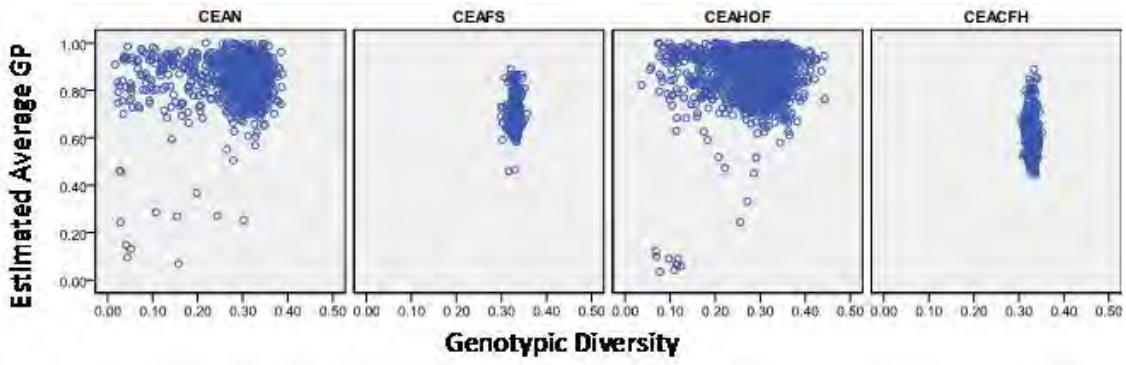

Figure 5.14: Scatter plot of estimated average GP versus genotypic diversity for the CEAN, CEAFS, CEAHOF and CEACFH algorithms

5.4.4.6 Genotypic Diversity and Objective Average Quality

Figure 5.15 shows the relationship between *genotypic diversity* and *objective average quality*. The x-axis and y-axis represents *genotypic diversity* and objective average respectively. The scatter plots here appear to be similar (except for the area of the regions being more spread out along the *objective average quality* axis, with values ranging from 50 to 100 for CEAN and CEAHOF and less spread out for CEAFS and CEACFH with values ranging approximately from 60 to 80) to those in the previous section which examined the relationship between *genotypic diversity* and *estimated average GP*.

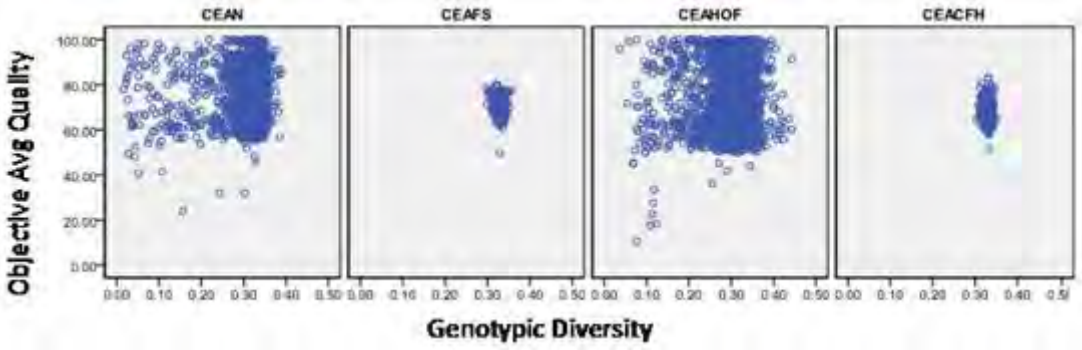

Figure 5.15: Scatter plot of objective average quality versus genotypic diversity for the CEAN, CEAFS, CEAHOF and CEACFH algorithms

5.5 Conclusion

Experiments were conducted with different variants of a naïve CEA by incorporating different techniques: FS, HOF, and a range of *mutation rates*. These variants were tested on a test problem called an intransitive number problem which was designed to be difficult for CEAs due to an intransitive superiority relationship between solutions. The effects of varying *mutation rate*s on the performance of the four algorithms were measured in terms of population diversity and solution quality. The diversity of the population was measured in two ways, genotypic and phenotypic, and the quality of the population was measured by using techniques such as the GP and objective quality. The data gathered from each measure was analysed in 4 ways using an interaction plots, ANOVA, correlation and scatter charts.

This study showed that the algorithms that incorporated fitness sharing were less influenced by the *mutation rate* in term of maintaining diversity in the population. This may be due to the reason that the population diversity is already maintained by fitness sharing. It was found that a moderate amount of diversity helps to achieve high quality solutions. In addition, the HOF method can also improve quality, but not as reliably as FS. The diversity maintenance methods that were tested in this study do not combine well with the HOF as CEACFH received the lowest quality (in terms of the GPs and theoretical optimum).

6 Testing CEAs for Multimodal Domains

Chapter 5 described the investigation of a basic CEA and three variants, in dealing with the intransitive number problem. This chapter aims to evaluate the capability of the four algorithms in terms of their generalisation and peak finding ability in multimodal domains. The optimization of multimodal problems is relatively challenging as the process needs to maintain parallel convergence into multiple solutions.

Multimodal domains have been studied by many researchers in the context of evolutionary algorithms, including Deb and Goldberg (1989), to test the effectiveness of the GA in finding multiple local optima. These authors identified that the GA converges not only to a single optimum but also to multiple peaks. Multimodal problems were also studied by Miller and Shaw (1995), Hansen and Kern (2004) and Ronkkonen, Li, Kyrki and Lampinen (2008). Some researchers, including Parsopuulos and Vrahatis (2004) and Yu and Suganthan (2010), also have tested multimodal functions using differential EAs. Researchers identified that multimodality often appears in real world problems such as machine learning problems (Mahfoud, 1995) and the inversion of teleseismic body waves (Koper, Wysession, & Wiens, 1999). A pilot study, a preliminary component of this study, also found that RT applications can demonstrate multimodality, which means that there may be more than one „good" strategy to address the opponent"s tactics.

In order to evaluate the capacity of CEAs in detecting multiple optima, a test problem is introduced and described in section 6.1. Section 6.2 outlines the selected parameters and operators used in the experimentation, followed by section 6.3 which shows the results for each algorithm using various performance measures.

6.1 Problem Domain: A Circular Multimodal Problem

The literature (including Deb & Goldberg, 1989; Yu & Suganthan, 2010) has shown that existing multimodal problems were designed for single population GAs or MOEA; however, none was found to be suitable for evaluating competitive CEAs. Therefore, in this study, a new circular multimodal problem known as the *n-peak* problem is

proposed. The advantage of using *n-peak* is that there are *n* equally good strategies for each side in which their locations (peaks) are known. Since there is no external objective function to optimize, the challenge for CEAs is to identify the defined *n* peaks. The *n-peak* problem is symmetric, which means both sides are evaluated using the same method. The domain for each side is the interval [0, 1]. Equations (6.1) to (6.6) show the method of calculating a score in which two individuals *x* and *y* compete and receive a score of either 0 or *L* or *H*. The symbol *n* represents the number of peaks to be identified and also the number of intervals within the domain. *H* and *L* are two payoff values with *H* > L and *L* > 0.

$$i_x = \lfloor (x \times n) \rfloor \tag{6.1}$$

$$i_y = \lfloor (y \times n) \rfloor \tag{6.2}$$

$$v_x = |0.5 - (x \times n) + i_x| \tag{6.3}$$

$$v_y = \left|0.5 - (y \times n) + i_y\right| \tag{6.4}$$

$$gap = mod(i_x - i_y,\ n) \tag{6.5}$$

$$score(x,y) = \begin{cases} H, & if\ gap\%2 = 0\ and\ v_x > v_y \\ L, & if\ gap\%2 = 1\ and\ v_x < v_y \\ 0, & otherwise \end{cases} \tag{6.6}$$

In a game between two players, one from each team, getting a payoff (winning) depends on two factors. The first factor is the intervals to which two players belong. Symbols i_x and i_y in Equations (6.1) and (6.2) represent the intervals in which individuals *x* and *y* fall. The second factor is the distance between the values and the centres of their intervals, which is represented by symbols v_x and v_y in Equations (6.3) and (6.4) for *x* and *y* individuals respectively. The value of *gap* in Equation (6.5) calculates a distance from *y* interval number to *x* interval number. For e.g. in a 5 peaks problem, if *x* is in the fourth interval and *y* is in second interval, the gap value will be 2 (i.e. mod ((4-2), 5)). On the basis of whether the *gap* value is odd or even number, as per Equations (6.6), *x* and *y* gets 0 or *L* or *H* score.

In order to visualize the winning teams in a colour code as per Equations (6.1) to (6.6), Figure 6.1 illustrates the situation when *H* = *L* = 1 and *n* = 5. If the intersection point of

x and y falls in the blue space, the x team wins and vice versa. For example, the red team wins in the positions marked as „A" (x = 0.1 & y = 0.05) and „B" (x = 0.3 & y = 0.95) but the blue team wins in the position marked as „C" (x = 0.75 & y = 0.7). The colour combinations of the blue and red were arranged in such a way that it follows the pattern of Equations (1) to (6).

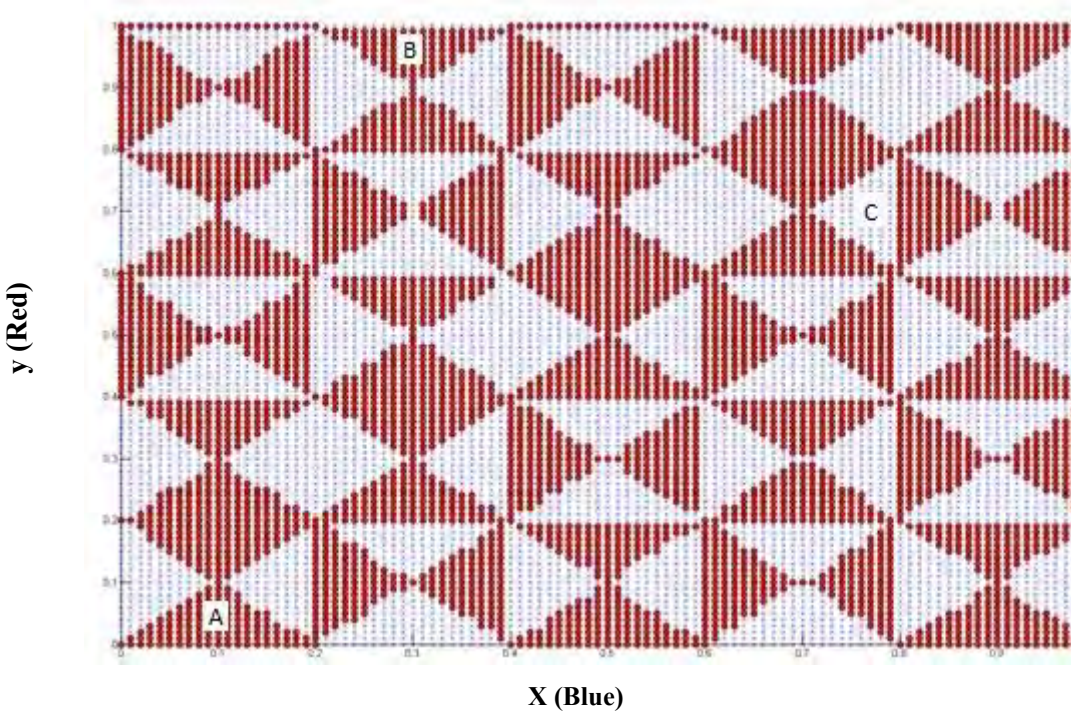

Figure 6.1: Winning and losing conditions in the multimodal function. Blue wins in the blue space and red wins in the red space.

Figure 6.2 also illustrates the situation when $H = L = 1$ and n=5. Those individuals, close to the boundaries of intervals, get a payoff of 1 against approximately 60% of randomly selected opponents, for an average payoff of 0.6. Those nearer the middle of their interval only get a payoff of 1 against about 40% of opponents, an average payoff 0.4. Although $n = 5$ has been used in Figure 6.2, the picture is similar for other odd values of n. By setting different values for H and L, the difference between peak and trough values can be manipulated. It is also straightforward to extend the idea to higher dimensions, by subdividing a hypercube into cells, and using Manhattan distance (as explained in Gillbert (1965)), between cells instead of using the gap value in Equation (6.5).

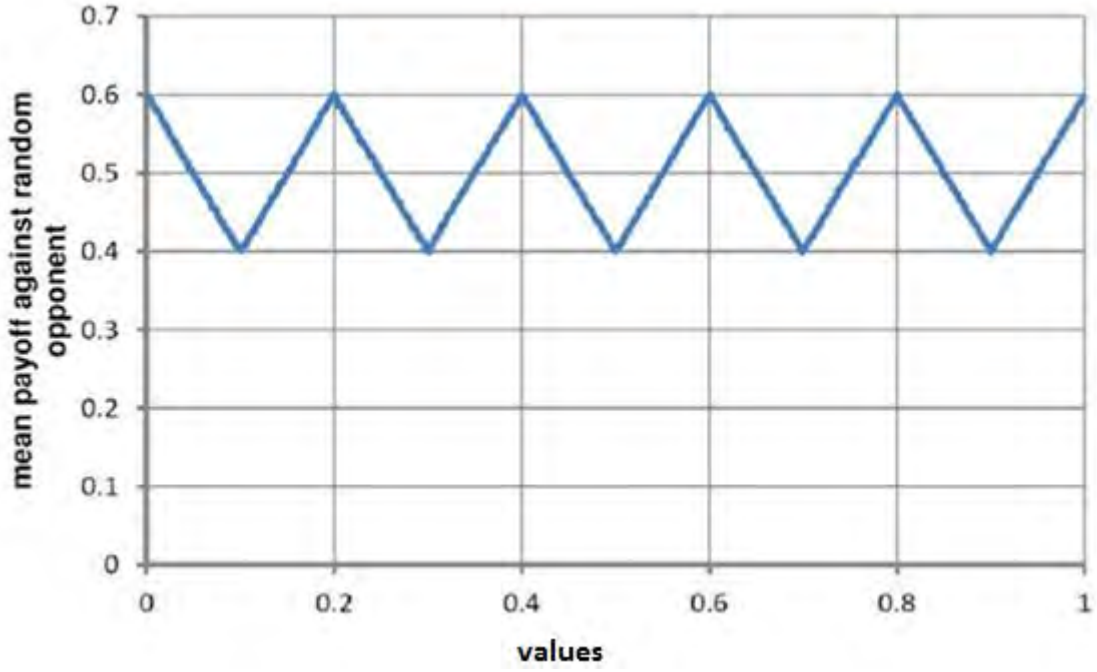

Figure 6.2: Mean payoffs against random opponents for solutions to the n=5 with H=L=1

6.2 Experimental Setup

Similar to the experiments in chapter 5, experiments are set up to compare the performance of a basic competitive CEA and variants as introduced in chapter 4 (CEAN, CEAFS, CEAHOF and CEACFH) in handling a multimodal problem using an appropriate set of performance measures. The performance of the algorithms was evaluated by measuring the quality of the population via the GP and the diversity via *genotypic* and *phenotypic diversity*. Unlike domains with a single optimum, the outcomes for a multimodal domain cannot be evaluated only on the basis of convergence to an optimum. Therefore, in order to test the effectiveness of algorithms in finding a number of local optima, performance measures such as the circular earth mover‟s distance (CEMD) and peak ratio and success ratio (previously described in sections 4.5.3 and 4.5.4 respectively) were also used.

The parameters used for the optimization process are listed in Table 6.1. To investigate the effects of diversity maintenance via FS and/or mutation, and of an archive in the form of a HOF, on the 5-peaks problem, each algorithm was executed 60 times to account for statistical variation, with *mutation rate* varying from 2.5% to 100% in

stepwise increments of 2.5%. In each run, results from the last 60 generations were averaged. The mean of these average values over 60 runs was used for the further analysis. For each execution, in each generation, diversity, generalisation ability, and peak finding ability were calculated. Generalisation ability was measured by using GPs in three ways: the *average, best and ensemble* (see section 4.5.1). Similarly to the intransitive number problem in chapter 5, ensemble did not provide meaningful information in this study despite its use by Chong et al. (2008, 2009). Ensemble measures displayed similar results for each of the four algorithms, making it hard to distinguish the algorithms‟ performance. Thus, the analysis of this measure was omitted from this study. Chong et al (2009) argued that diversity of the population highly influences performance/quality. In order to see the influence of diversity in the population‟s performance, diversity was measured in two ways: *genotypic and phenotypic*. Peak finding ability was measured using the three techniques: CEMD, peak ratio (PR) and success ratio (SR). Similar to chapter 5, the data for GPs and two diversity measures were examined in four ways: (1) An interaction (profile) plot of each measure against *mutation rates,* (2) ANOVA test, (3) correlation analysis for measures associated with each of the four algorithms and (4) scatter plots are presented to visualise the relationship between quality and diversity.

Table 6.1: CEA parameters used

Properties	**Algorithms/Values**
Population size	50 in each population
Gene value	0 to 1
Crossover	Single point
Crossover rate	60%
Mutation	Gaussian
Mutation rates	2.5% to 100% stepwise increments of 2.5%
Selection	Stochastic universal sampling
Generations	300
Number of runs	60
Niche radius	0.2 (best value suggested by experimentation)
HOF sample size	50 (equal to population size)

Initial gene values were randomly generated values between 0 and 1. As in the intransitive number problem, the interaction between two individuals produces a score for each individual involved. Since the multimodal problem is symmetrical, two

opposing individuals were evaluated using the same method using Equations (6.1) to (6.6). For calculating fitness as shown in Equation (2.1), when the individual is evaluated against all the members from the competing population scores received by the individual are averaged. In terms of FS, an empirical study was carried out to investigate suitable values for niche radius. The value of 0.2 was found to be the best niche radius for this domain.

6.3 Results and Analysis

As mentioned in section 6.2, four algorithms, the CEAN, CEAFS, CEAHOF and CEACFH were employed in optimizing this multimodal problem. The four algorithms‟ quality in terms of their generalisation and peak finding ability are presented in this section. Generalisation ability is measured using GP (see section 4.5.1) and the peak finding ability is measured using the techniques such as CEMD, PR and SR (see sections 4.5.3 and 4.5.4). Additionally, diversity of the populations is also measured to evaluate the influence of diversity on the performance of populations. In this section, the relationship between diversity and quality is also presented.

6.3.1 Evaluation of Algorithms via GPs

This section presents an analysis of the two competing populations‟ quality via the *estimated best GPs* for all four algorithms. Two populations were labelled blue and red respectively. The GP measured the performance of evolving populations by evaluating against the fixed test population (see Table 4.4). Each of the four algorithms was run 60 times and the average value for the *best GP* of those 60 runs in each generation is presented as convergence plots which are depicted in Figure 6.3. The x-axis represents the number of generations and the y-axis represents the *estimated best GPs*.

Figure 6.3 (a) and (b) show the *best GPs* of the blue and red teams respectively when 2.5% *mutation rate* was applied. The solution associated with the CEACFH scored the highest *estimated best GP* in both the blue and red team. Solutions associated with the CEAFS achieved the second highest *estimated best GP* and those associated with the CEAHOF and CEAN received relatively lower *estimated best GP*. Subsequently, when

a 50% *mutation rate* was applied, convergence plots of the blue and red teams are depicted in Figure 6.3 (c) and (d) respectively. Solutions associated with the CEAHOF have better *estimated best GP* values in comparison to those associated with the 2.5% *mutation rate*. The performance of all four algorithms is quite noisy, fluctuating up and down within a narrow range of 0.52 to 0.6. Similar results can be seen in both teams when a 100% *mutation rate* was applied (Figure 6.3 (e) and (f)). The convergence plots with 2.5%, 50% and 100% *mutation rate* show that the blue and red teams" performances were similar in achieving the *estimated best GP*. This was expected, as the proposed multimodal problem was designed to be symmetric, i.e. both sides are evaluated using the same function. These similarities confirm that it is sufficient to use data from only one team for further analyses. The same procedure as described in section 5.4.1 was used here.

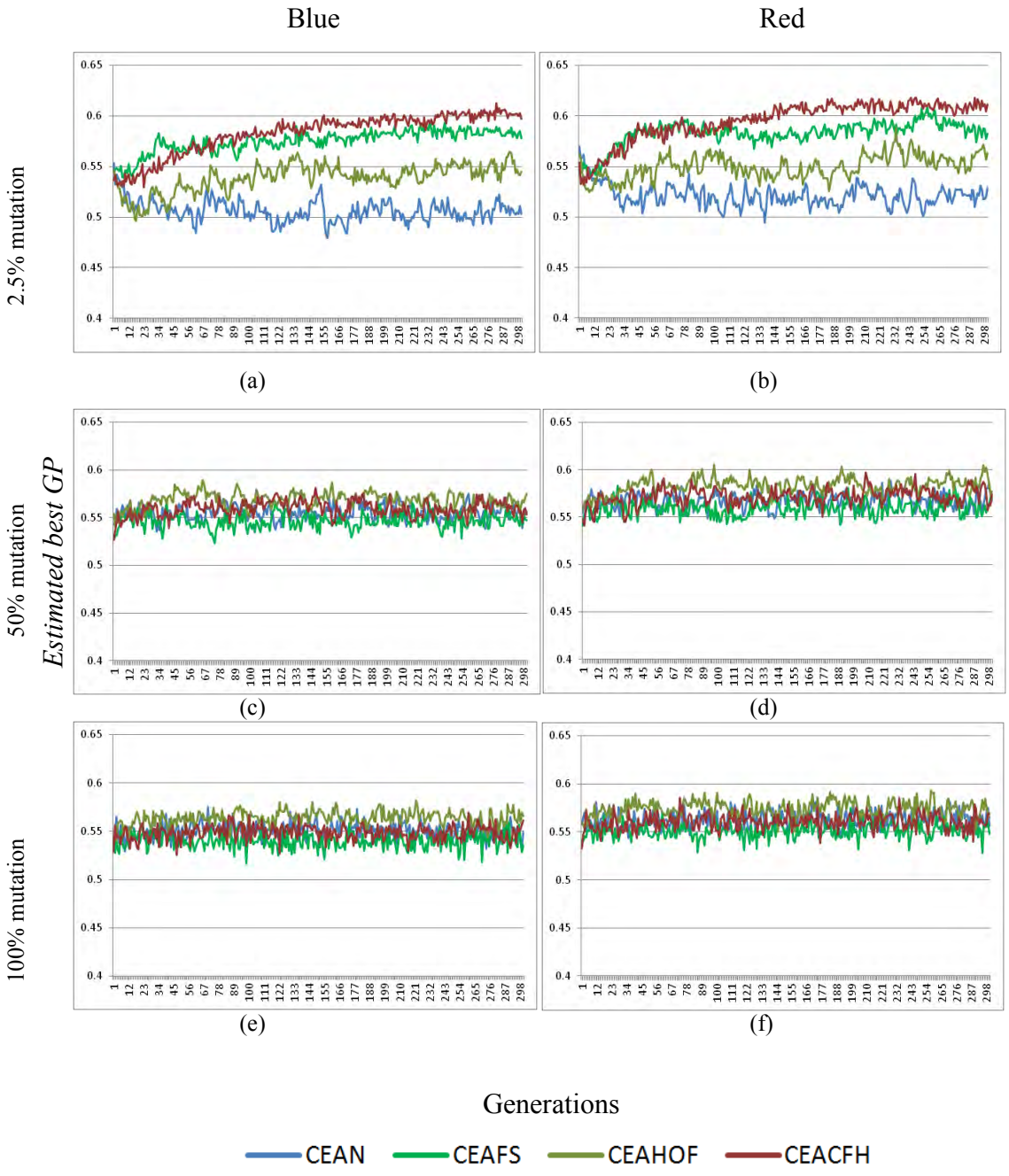

Figure 6.3: Convergence plots showing the estimated best GPs of the four algorithms for the blue and red team in mutation rates of (a) 2.5% for blue (b) 2.5% for red (c) 50% for blue (d) 50% for red (e) 100% for blue and (f) 100% for red teams

6.3.1.1 Analysing Estimated Best GP

Figure 6.4 is an interaction plot that depicts the *estimated best GP* of the populations for each of the four algorithms, with *mutation rate* varied from 2.5% to 100% in stepwise increments of 2.5%. For reference, assuming that both populations are reasonably diverse, GP should theoretically be in the range 0.4 to 0.6. Figure 6.4 shows that the

estimated best GP of CEAN and CEAHOF increases for *mutation rate* of up to about 20%. Subsequently when *mutation rate* of above 20% were applied, the *estimated best GP* was level out in CEAN and slowly declined in CEAHOF. In CEAFS and CEACFH, an increase in *mutation rate* up to above 10%, increases the *estimated best GP* which slowly deteriorates with increasing the *mutation rate* above 20%. From the figure, it also appears that there is a specific *mutation rate* where the *estimated best GP* is optimum for CEAFS and CEACFH, supporting the argument that an appropriate *mutation rate* is required to enhance the quality of the population. For these two algorithms *mutation rate* of about 10% seems the best. However, for CEAN, *mutation rate* of 20% or above seems good and for CEAHOF *mutation rate* between 20% and 40% seems good.

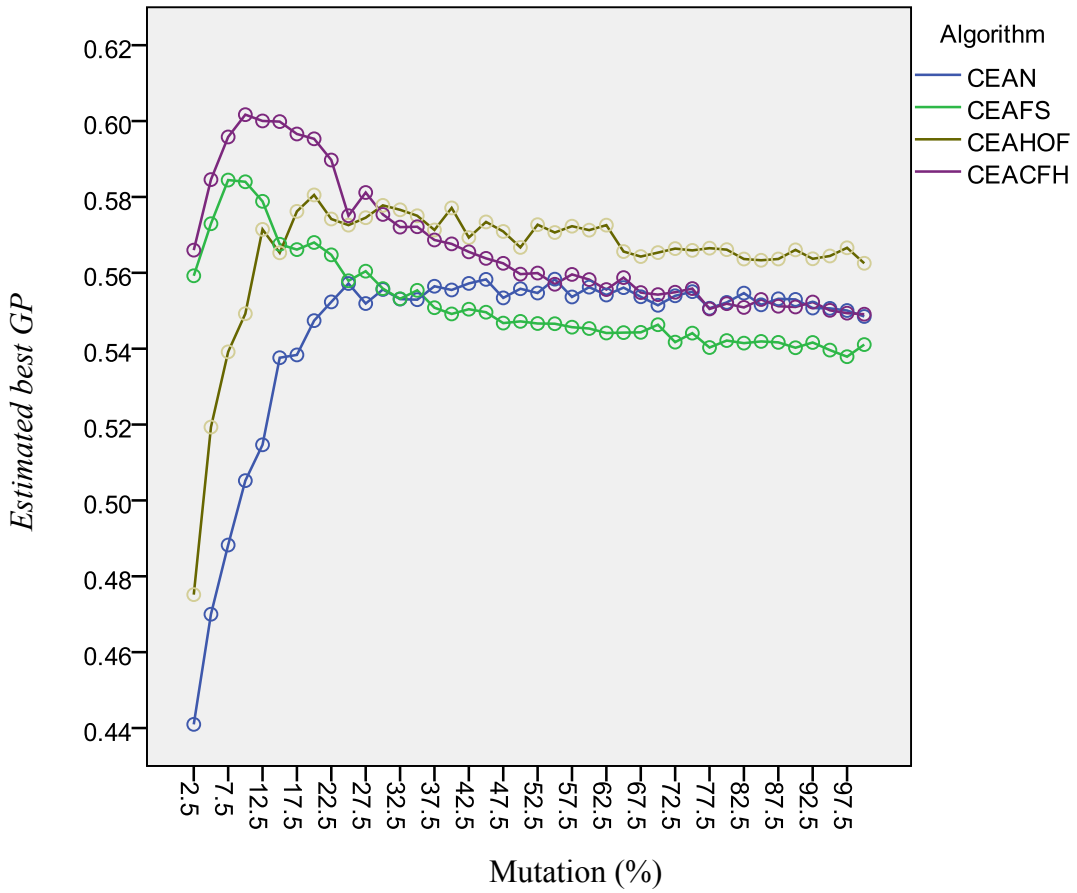

Figure 6.4: Interaction plots of *estimated best GP* (mean over the final 60 generations out of 300 generations in 60 runs) versus *mutation rate* for each of the 4 algorithm variants

The *estimated best GP* was analysed by means of a two-way between-subjects ANOVA test with 40 levels of *mutation rate* and four types of *algorithm used*. All effects were statistically significant. The interaction effect was, ($F(117.9440) = 59.879$, $p < 0.05$, Partial Eta Squared = 0.426). The associated ANOVA table is depicted in Appendix B.1.

6.3.1.2 Analysing Estimated Average GP

Figure 6.5 is an interaction plot that shows the *estimated average GP* associated with each of the four algorithms with *mutation rate* varies from 2.5% to 100% in 2.5% stepwise increments. Similar to the *estimated best GP*, the *estimated average GP* for CEAFS and CEACFH increases up to *mutation rates* about 10%, then rapidly decay for higher *mutation rate*. For CEAN and CEAHOF, increase of *mutation rate* up to above 15% increases the *estimated average GP*. Subsequently it level out up to about *mutation rate* of 20% which rapidly decay above *mutation rate* of 20%. The slope of each of the four algorithms was approximately similar for *mutation rate* greater than 40% and increasing *mutation rate* above 50% seems to have little impact on the GP of each of the algorithms.

The *estimated average GP* was analysed by means of a two-way between-subjects ANOVA test with 40 levels of *mutation rate* and four types of *algorithm used.* All effects were statistically significant. The interaction effect was, ($F(117, 9440) = 58.028$, $p < 0.05$, Partial Eta Squared = 0.418). The associated ANOVA table is depicted in Appendix B.2.

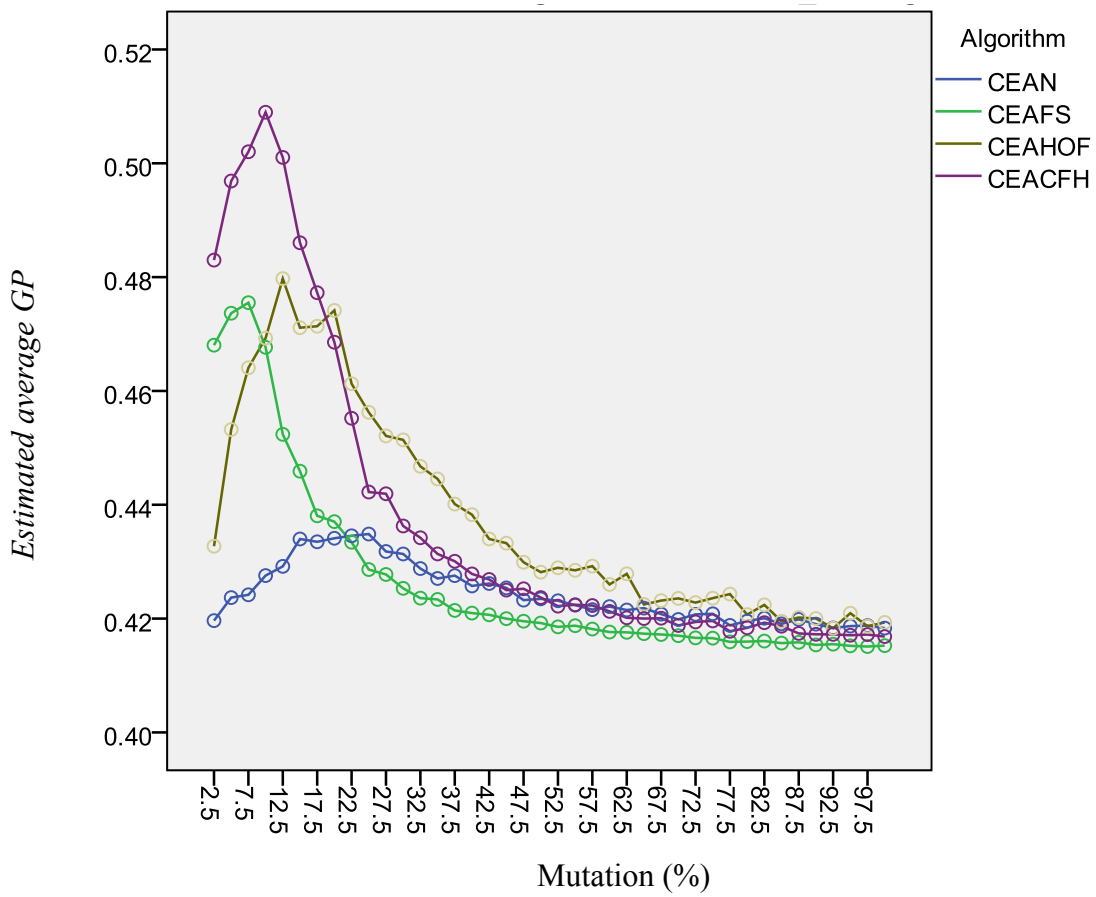

Figure 6.5: Interaction plots of estimated average GP (mean over the final 60 generations out of 300 generations in 60 runs) versus mutation rate for each of the 4 algorithm variants

From the analysis of *estimated average GP* in this section and the previous section for *estimated best GP*, the two-way between-subjects ANOVA test with 40 levels of *mutation rate* and four types of *algorithm used* showed consistently that the interaction effect was significant for both *GP* measures. In both cases, the population associated with CEACFH consistently has higher *GP* value at low mutation rates in comparison with the other three algorithms. Interestingly, as *mutation rate* increases towards 100%, the four algorithms were spread within a band value of 0.54 to 0.57 for *estimated average GP* (towards upper limit of the theoretical range of 0.4 to 0.6) while the *estimated average GP* of the four algorithms converges towards the lower theoretical limit; around 0.42.

The following section presents the analysis of a naïve CEA with three variants in terms of their ability of finding multiple peaks.

6.3.2 Analysis of Algorithms for Peak Detection Ability

This section explores the performance of the four algorithms in finding multiple peaks via the techniques of CEMD, PR and SR. Again each algorithm was evaluated with *mutation rate* varying from 2.5% to 100% in 2.5% increments.

6.3.2.1 Analysis via CEMD

Figure 6.6 shows an interaction plot in which CEMD associated with all four algorithms was investigated by varying *mutation rate* from 2.5% to 100% with a stepwise increment of 2.5%. The x-axis represents the *mutation rate* and y-axis represents the CEMD. Note that lower CEMD values imply that the distribution of the evolving population is similar to the ideal distribution. Thus, in this analysis, the algorithm which scores low CEMD values is better as it shows that the evolving population can better detect more of the specified number of peaks.

The CEMD results associated with both CEACFH and CEAFS appeared to be quite similar and have the lower CEMD values in comparison to CEAN and CEAHOF. The performance of both these algorithms was better at low levels of *mutation rate*. As *mutation rate* increased, the performance of CEACFH and CEAFS deteriorates. The CEMD plots associated with the CEAN and CEAHOF show that these algorithms improved their performance as the *mutation rate* increases but never attain performance equal to CEACFH or CEAFS. Eventually, the performance of all four algorithms became similar within a band of 2 to 3 at *mutation rates* closer to 100%; *mutation rates* beyond 50% appears to have little impact on CEMD (indicated by slope of the associated points in this region).

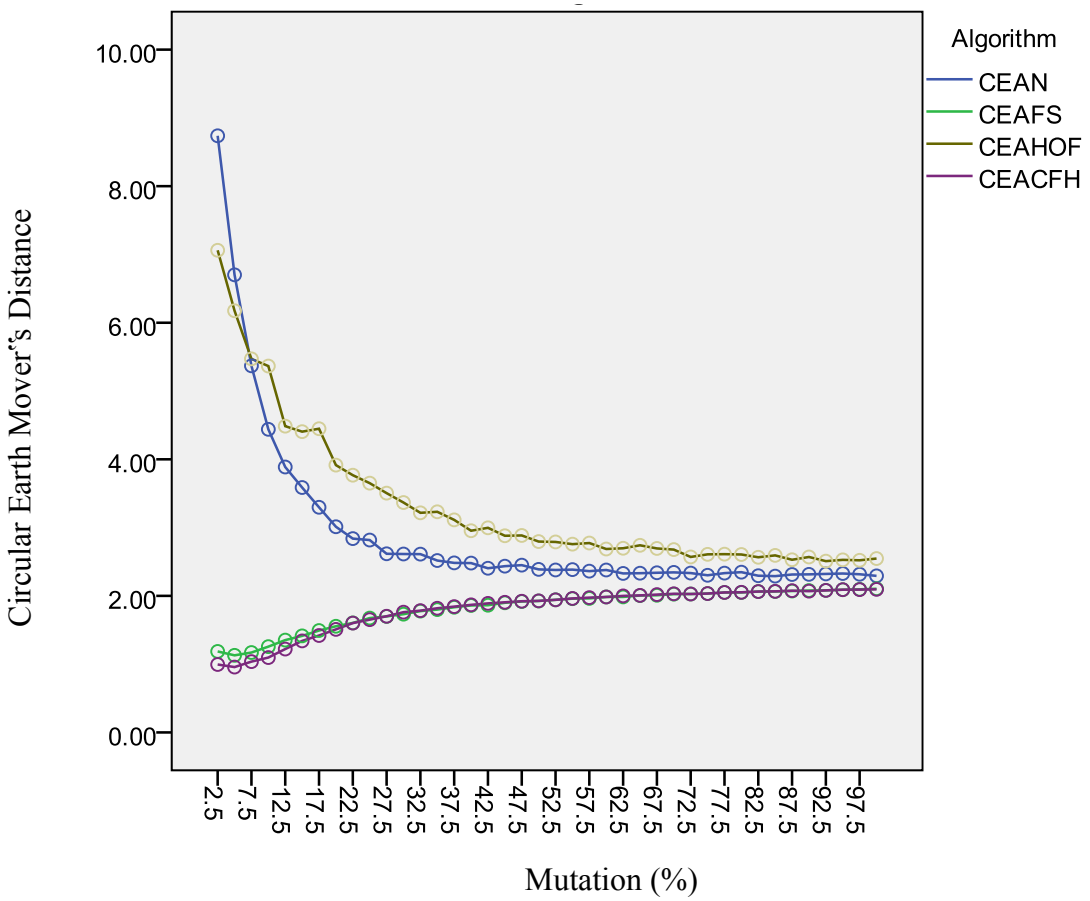

Figure 6.6: Interaction plots of circular earth mover's distance (mean over the final 60 generations out of 300 generations in 60 runs) versus mutation rate for each of the 4 algorithm variants

Consequently, the CEAFS and CEACFH, both of which incorporated fitness sharing, were relatively good algorithms that detect a number of peaks in this 5-peaks multimodal domain. The performance of the CEAN and CEAHOF may also be improved by using higher *mutation rates*.

The CEMD was analysed by means of a two-way between-subject ANOVA test with 40 levels of *mutation rate* and four types of *algorithm used*. All effects were statistically significant. The interaction effect was, ($F(117, 9440) = 151.577$, $p < 0.05$, Partial Eta Squared = 0.843). The table associated with ANOVA is depicted in Appendix B.3.

6.3.2.2 Analysis via PR and SR

In order to support the result obtained for CEMD, two more tests were performed, the PR and SR (details in section 4.5.4). The PR was calculated on the basis of what percentage of optima (peaks) were identified. In this experiment, there were 60 runs each with 300 generations. Unlike the GPs and CEMD, the average of only the last generation‟s population from each run was calculated for this analysis.

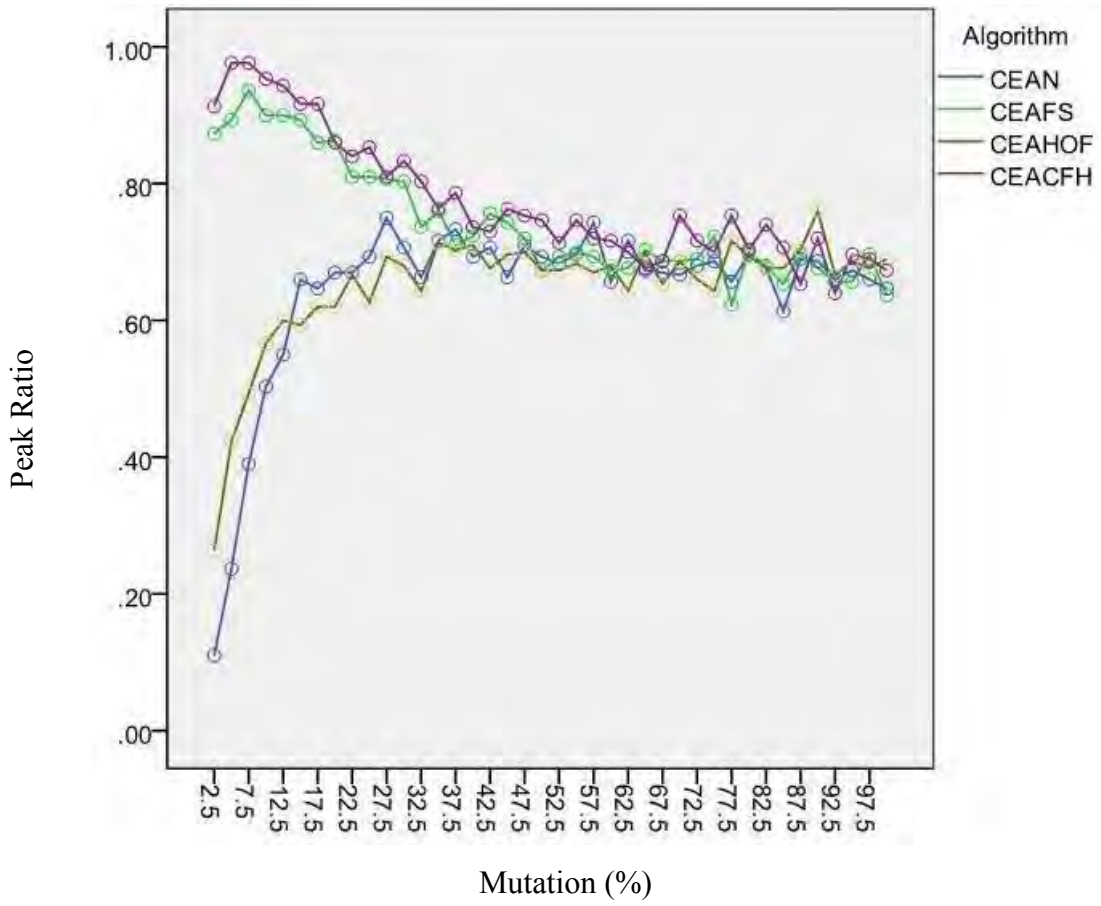

Figure 6.7: Interaction plots of peak ratio versus mutation rate for each of the 4 algorithm variants. Each point is an average of how many global optima the algorithm detects in the final generation over 60 runs

Figure 6.7 shows the PR for *mutation rate* ranging from 2.5% to 100% at 2.5% increments were used. This analysis also supported the finding made using CEMD, that the CEACFH appear to be slightly better in terms of detecting the specified peaks in comparison to the other three algorithms. The performance of the CEAFS is very similar to CEACFH. Both of these algorithms incorporated fitness sharing. In these two algorithms, as *mutation rate* increases (between 7.5% to approximately 40%), PR

values decrease and appear to flatten out between 40% to 100%, indicating that increasing *mutation rate* in this interval has little impact on the performance of these two algorithms. The CEAN and CEAHOF were not effective in detecting multiple optima with lower *mutation rate*. Their performance gradually improved as *mutation rate* increases (between 2.5% to approximately 40%) and then flattened out when *mutation rate* is in the range between 40% and 100%. The performance of all four algorithms was similar for higher *mutation rate* (as indicated by the slopes of their respective points), showing that increasing *mutation rates* in this interval has little impact on the performance of these algorithms.

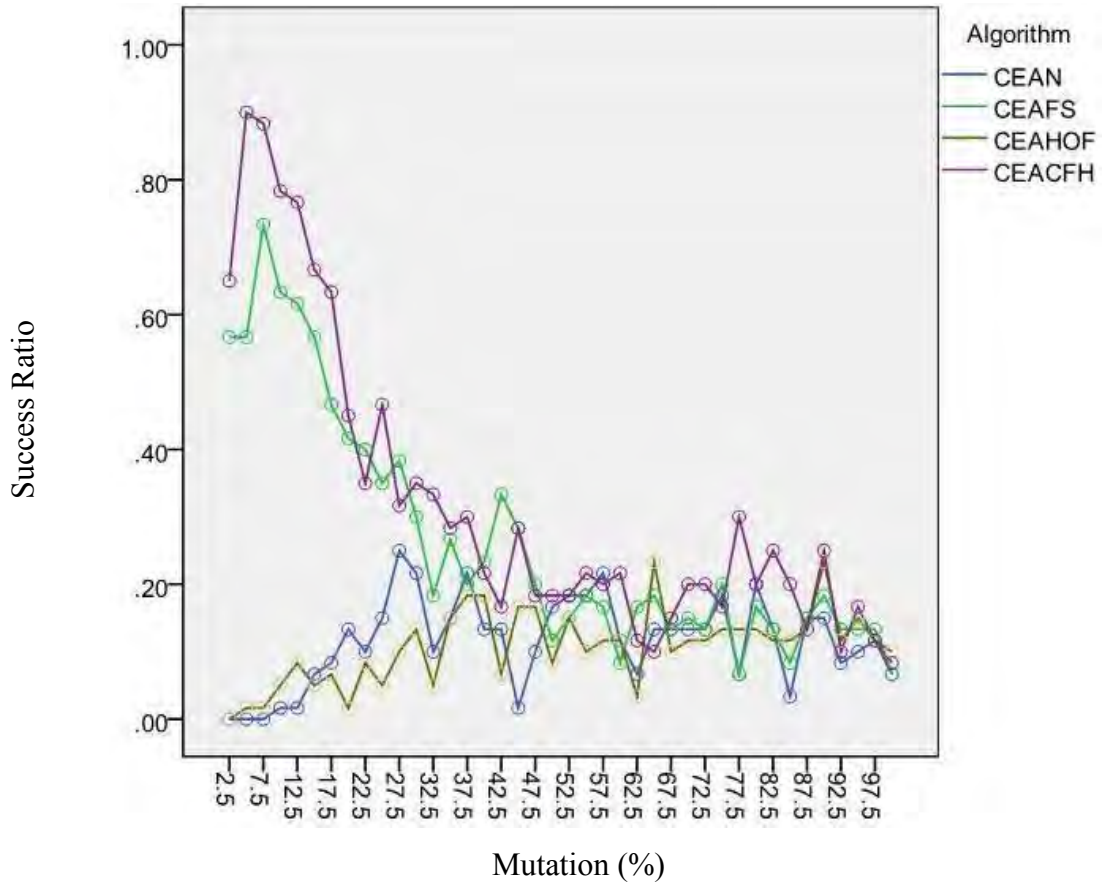

Figure 6.8: Interaction plots of success ratio versus mutation rate for each of the 4 algorithm variants. Each point is an average of how many times the algorithms recognized all five global optima in the last generation over 60 runs

The SR was also calculated according to the method described in section 4.5.4. Each point is an average of 60 runs in which each run it receives either 1 or 0 on the basis of whether it recognizes all local optima within 10^{-2} tolerance. Figure 6.8 is an interaction

plot that shows the interaction of the *mutation rate* and SR. Results are similar to those obtained with PR. CEACFH is relatively good in detecting all global optima. The performance of the CEAFS was similar to the CEACFH. Both these algorithms‟ performance decreased when *mutation rate* increased (between 7.5% to approximately 40%). The performance of the CEAN and CEAHOF in detecting all global optima was not effective; however, their performance improved slightly when higher *mutation rate* was applied, between 2.5% to approximately 40%. Similar to PR, the performance of all four algorithms was similar when *mutation rate* was between 40% and 100%, fluctuating up and down within a narrow band between 0.02 and 0.3.

The above analysis showed performance of each of the four algorithms‟ ability in detecting multiple optima. Similar to the result from the CEMD, the PR and SR also show that CEACFH is a better algorithm for optimizing this multimodal problem. Specific to this domain, the result indicated that the performance of the naïve CEA can be enhanced by integrating a combined approach of the HOF and FS. The PR and SR indicated that the problem can be best optimized if CEACFH is used with a *mutation rate* of 5%. The analysis indicated that *mutation rate* can influence the performance of all four algorithms.

6.3.3 Analysis of Diversity

In order to investigate diversity of the population associated with each of the four algorithms employed in this study, diversity was measured in two ways: genotypic and phenotypic. This section presents the analysis of each of these diversity measures in sections 6.3.3.1 and 6.3.3.2 respectively.

6.3.3.1 Analysis of Genotypic Diversity

This section examines *genotypic diversity* associated with the evolved populations for each of the four algorithms. Figure 6.9 is an interaction plot which shows *genotypic diversity* versus *mutation rate* which varied from 2.5% to 100% in 2.5% increments. The x-axis is *mutation rates* and y-axis is *genotypic diversity.* In the CEAFS and CEACFH, the diversity was high even for low *mutation rate,* i.e. there was no effect of *mutation rate* on *genotypic diversity* in these algorithms. These are the algorithms that

incorporated FS. This indicated that FS is an effective means of increasing diversity in populations. In CEAN and CEAHOF there was rapid increase in genotypic diversity from mutation rates of 2.5% to 20%. Both these algorithms‟ plots plateau out with some fluctuations for *mutation rate* greater than 30%. Genotypic diversity associated with all four algorithms was within a band value of 30 to 32 for the *mutation rate* above 30%. The slope of the algorithms was similar in CEAFS and CEACFH and also in CEAN and CEAHOF respectively; the slope was almost negligible above the *mutation rate* of 30%. As mentioned in section 5.4.1.1, the CEAHOF produce less diverse population than the CEAN. This may be due to the additional interaction with HOF members.

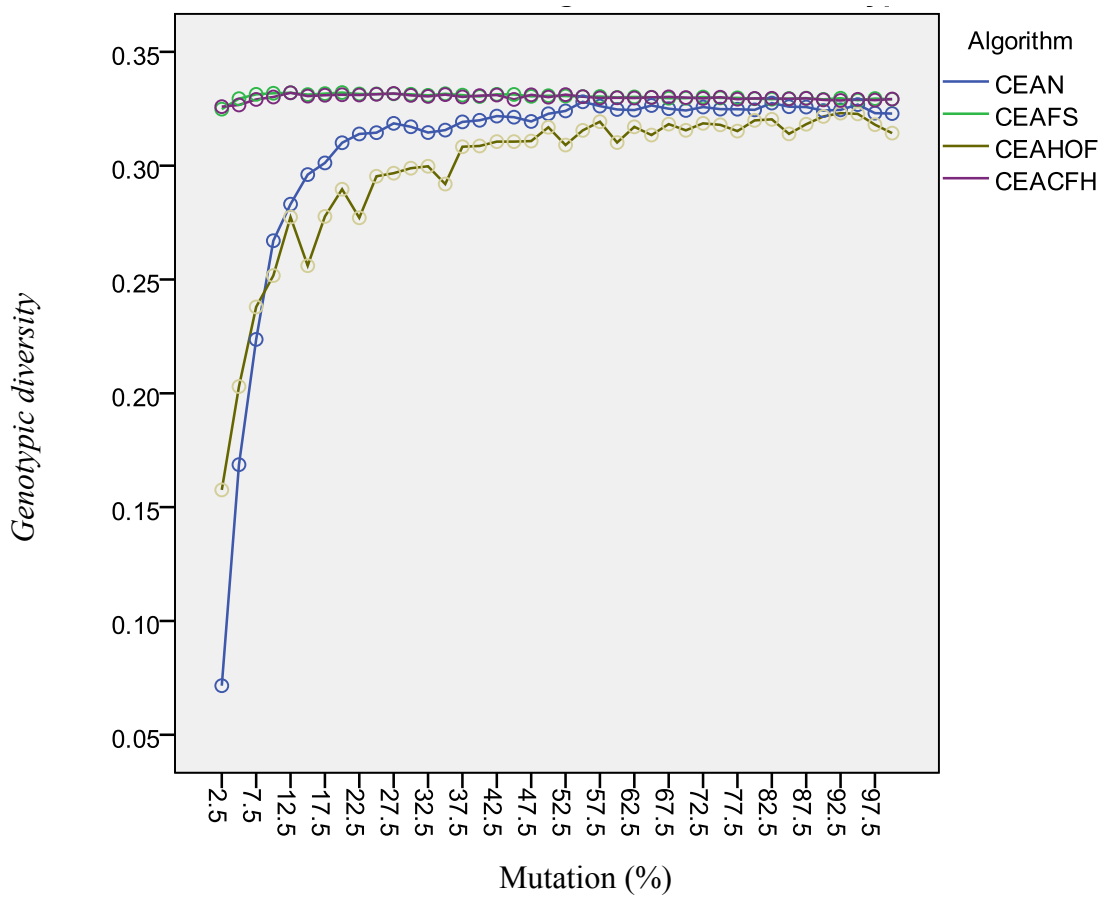

Figure 6.9: Interaction plots of genotypic diversity (mean over the final 60 generations out of 300 generations in 60 runs) versus mutation rate for each of the 4 algorithm variants

The *genotypic diversity* was analysed by means of a two-way between-subject ANOVA test with 40 levels of *mutation rate* and four types of *algorithm used.* All effects were statistically significant. The interaction effect was, ($F(117, 9440) = 56.284$, $p < 0.05$, Partial Eta Squared = 0.411). The associated ANOVA table is depicted in Appendix B.4.

As discussed in chapter 5, although mutation influences the diversity of the population associated with CEAN and CEAHOF, the effect was very low in the populations associated with CEAFS and CEACFH. As mentioned earlier, this could be due to the use of FS which already diversify the population. However, increasing *mutation rate*s between 2.5% to 30% increases diversity in the population evolved by CEAN and CEAHOF and from the plots associated with peak finding (CEMD, PR and SR), it can be seen that their performance in peak detection also improved.

6.3.3.2 Analysis of Phenotypic Diversity

This section presents the effect of *mutation rate* on *phenotypic diversity* in all four algorithms. Figure 6.10 is an interaction plot which shows *phenotypic diversity* versus *mutation rates* which were varied from 2.5% to 100% in 2.5% increments. In the CEAN and CEAHOF, there was a rapid increase in *phenotypic diversity* from the *mutation rate* of 2.5% to 20% which plateau out when the *mutation rate* is greater than 20%. In the CEAFS and CEACFH, an increase in *mutation rate,* from 2.5% to 12.5%, increases *phenotypic diversity* but it decreased from *mutation rate* of 17.5% to 27.5% which plateau out when the *mutation rate* is greater than 27.5%.

The phenotypic diversity of the four algorithms was almost the same for *mutation rate* greater than of 27.5%, with the slope of the line associated with the data points for each algorithm being almost zero indicating that *mutation rate* has no impact on phenotypic diversity once the *mutation rate* is greater than 27.5%.

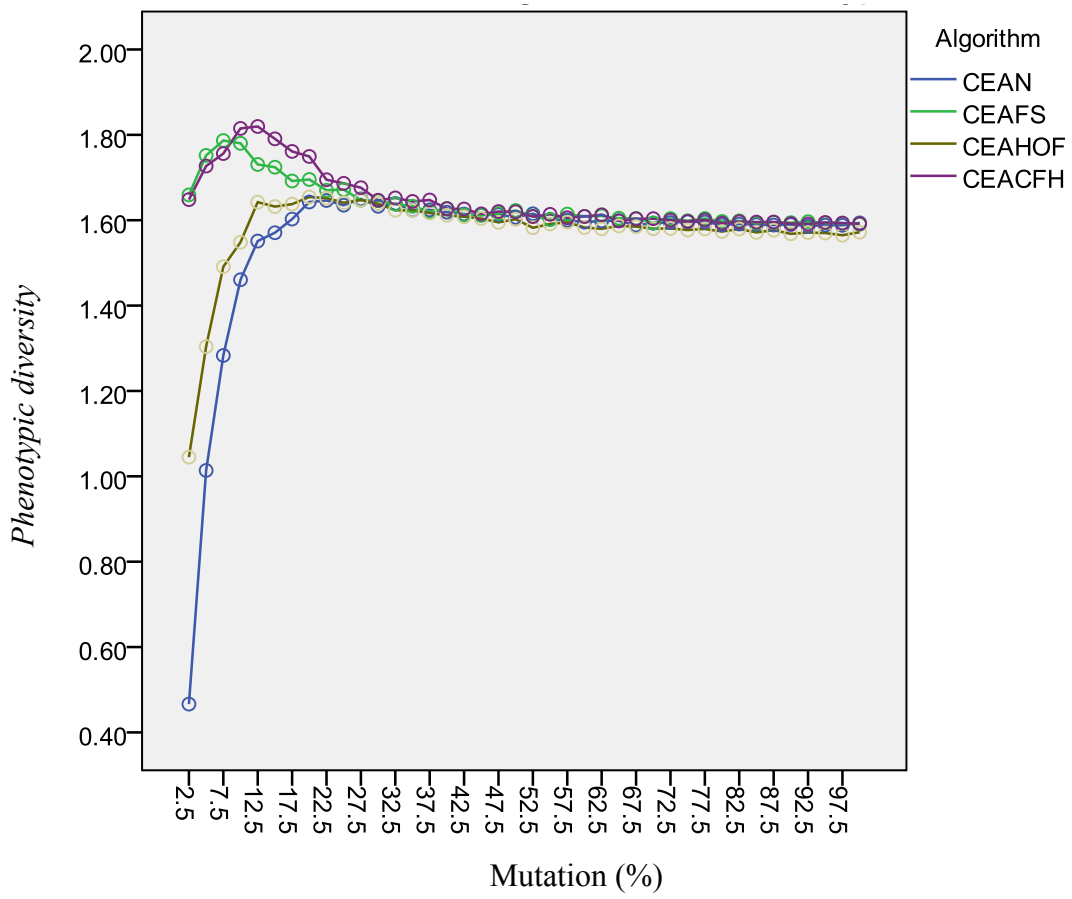

Figure 6.10: Interaction plots of phenotypic diversity (mean over the final 60 generations out of 300 generations in 60 runs) versus mutation rate for each of the 4 algorithm variants

The *phenotypic diversity* was analysed by means of two-way between-subjects ANOVA test with 40 levels of *mutation rate* and four types of *algorithm used.* All effects were statistically significant. The interaction effect was ($F(117, 9440) = 213.414$, $p < 0.05$, Partial Eta Squared = 0.726). The table associated with ANOVA is depicted in Appendix B.5.

The trend for *genotypic* and *phenotypic diversity,* of the populations associated with CEAN and CEAHOF, is similar. In these two algorithms, *genotypic diversity* increases up to about *mutation rate* of 50% whereas *phenotypic diversity* increases up to about *mutation rate* of 20%. The trend of these two diversity measures for CEAFS and CEACFH is also similar. At *mutation rates* between 2.5% and 12.5%, the diversity slightly increased, it decreased at *mutation rates* between 12.5% and 22.5% and flattens out for *mutation rate* above 22.5%. This indicated that there is a positive relationship between two diversity measures. Based on the interaction plots, the two algorithms with

FS may have weak relationship between *genotypic* and *phenotypic diversity*. A correlation analysis is presented to evaluate their relationship in the following section.

6.3.4 Relationship between Diversity and Quality

Correlation analysis was conducted to evaluate the relationship between different variables such as GPs, CEMD, *genotypic diversity* and *phenotypic diversity*. The relationships between these variables in all four algorithms, CEAN, CEAFS, CEAHOF and CEACFH are presented in Table 6.2, Table 6.3, Table 6.4 and Table 6.5 respectively. In CEAN, there was a strong correlation between *genotypic* and *phenotypic diversity* as discussed in the evaluating of the interaction plots for these two diversity measures. The positive correlation between each of the diversity measures with the *estimated best GP* was also strong, indicating that an increase in diversity of the population leads to an increase in the *estimated best GP*. Conversely, there was a strong negative correlation between CEMD and each type of diversity measures and also with *estimated best GP*. This also showed that, as expected, a population with the high diversity has a small CEMD and low *best GP*-value. Lastly, the *estimated average GP* is negatively correlated with the *genotypic diversity* and positively correlated with the *phenotypic diversity*.

Table 6.2: Correlation between various factors in the CEAN

	Genotypic	Phenotypic	Avg_GP	Best_GP	CEMD	Mutation
Genotypic	1					
Phenotypic	**.854****	1				
Avg_GP	-.020	**.135****	1			
Best_GP	**.758****	**.755****	**.309****	1		
CEMD	**-.899****	**-.904****	**.065****	**-.820****	1	
Mutation	**.513****	**.382****	**-.462****	**.475****	**-.621****	1
**. Correlation is significant at the 0.01 level (2-tailed).						
*. Correlation is significant at the 0.05 level (2-tailed).						
N= 2400						

In CEAFS, there was a weak positive correlation (0.111) between *genotypic* and *phenotypic diversity.* The correlation between *genotypic diversity* and *estimated average*

GP was not significant which indicated that in this particular algorithm, the change in the *genotypic diversity* does not impact the *estimated average GP*. The CEMD was negatively correlated with the *estimated best GP, estimated average GP* and both types of diversity measures. As discussed in section 6.3.2.1, this is expected as a low value in CEMD shows that the evolving solutions are similar to the distribution of peaks in the specified *n-peak* problem. In addition, the correlation between the CEMD and *mutation rate* is strongly positive, that is in this particular algorithm an increase in *mutation rate* increases CEMD (as shown in Figure 6.6), implying that the algorithm performance in detecting peaks decreases. In addition, the *estimated best GP* positively correlated with *genotypic* and *phenotypic diversity*; however, the correlations are moderate at 0.125 and 0.336 respectively.

Table 6.3: Correlation between various factors in the CEAFS

	Genotypic	Phenotypic	Avg_GP	Best_GP	CEMD	Mutation
Genotypic	1					
Phenotypic	**.111****	1				
Avg_GP	.036	**.636****	1			
Best_GP	**.125****	**.336****	**.746****	1		
CEMD	**-.066****	**-.674****	**-.856****	**-.599****	1	
Mutation	**-.139****	**-.584****	**-.680****	**. 523****	**.888****	1
**. Correlation is significant at the 0.01 level (2-tailed).						
*. Correlation is significant at the 0.05 level (2-tailed).						
N= 2400						

As indicated in Table 6.4 and similarly to CEAN and CEAFS, there was negative correlation between the CEMD with each of the two diversity measures, *genotypic and phenotypic* in CEAHOF. As described in the previous section, an increase in diversity improves the performance of the population associated with CEAHOF in evolving solutions that are similar to the ideal solutions. Both diversity measures were positively correlated with the *estimated best GP*; their relationship was significant but moderate, indicating that improving diversity will also improve the quality in terms of GP. Similar to CEAN, there is a negative correlation between the CEMD and *mutation rate* which indicated that an increase in *mutation rate* enhance this algorithm‟s performance to

evolve solutions that are similar to the distribution of peaks in the specified *n-peak* problem.

Table 6.4: Correlation between various factors in the CEAHOF

	Genotypic	Phenotypic	Avg_GP	Best_GP	CEMD	Mutation
Genotypic	1					
Phenotypic	**.388****	1				
Avg_GP	**-.288****	**.133****	1			
Best_GP	**.403****	**.487****	**.362****	1		
CEMD	**-.657****	**-.542****	**.493****	**-.407****	1	
Mutation	**.458****	**.166****	**-.650****	**.172****	**-.727****	1
**. Correlation is significant at the 0.01 level (2-tailed).						
*. Correlation is significant at the 0.05 level (2-tailed).						
N= 2400						

As shown in Table 6.5, the correlation between *genotypic and phenotypic diversity* was positive but weak in terms of CEACFH. Interestingly, the correlation between *genotypic diversity* and *estimated best GP* is weak. CEMD is positively and strongly correlated with the *mutation rate* which indicated that as in CEAFS, an increase in *mutation rate* increases CEMD (as shown in Figure 6.6), implying that the algorithm performance in detecting peaks decreases. CEACFH is very similar in terms of its relationship with the other measures except that for this algorithm, it has a very weak negative correlation between *genotypic diversity* and *estimated average GP*.

Table 6.5: Correlation between various factors in the CEACFH

	Genotypic	Phenotypic	Avg_GP	Best_GP	CEMD	Mutation
Genotypic	1					
Phenotypic	**.045***	1				
Avg_GP	-.005	**.669****	1			
Best_GP	**.130****	**.426****	**.749****	1		
CEMD	**.055****	**-.656****	**-.894****	**-.606****	1	
Mutation	**-.056****	**-.612****	**-.750****	**-.613****	**.865****	1
**. Correlation is significant at the 0.01 level (2-tailed).						
*. Correlation is significant at the 0.05 level (2-tailed).						
N= 2400						

6.3.4.1 Relationship between Genotypic Diversity and Estimated Best GP

In order to visualize the relationship between the *genotypic diversity* and *estimated best GP*, Figure 6.11 contains four scatter plots representing each algorithm. The CEAN plot showed that *genotypic diversity* and the *estimated best GP* has a positive correlation, showing the relationship that was discussed in the previous section. The number of points at the top part is relatively more than at the bottom, which indicates that there were more diverse populations with the high GPs and a small number of low diverse populations with low *best GPs*. The points are much more scattered in the CEAHOF which confirmed a lower correlation relationship (as indicated in Figure 6.5). There are still quite a high number of points associated with populations with high diversity and high GP but it appears that there is a concentration of points in the middle region (average diversity with average GP). In the case of CEAFS and CEACFH, the points were almost in a horizontal straight line which is why there was a very weak correlation between *genotypic diversity* and the *estimated best GP*. The populations associated with these 2 algorithms all have high *genotypic diversity,* probably from the incorporation of the fitness sharing algorithm and their GP values are concentrated within a range of approximately 0.5 to 0.65.

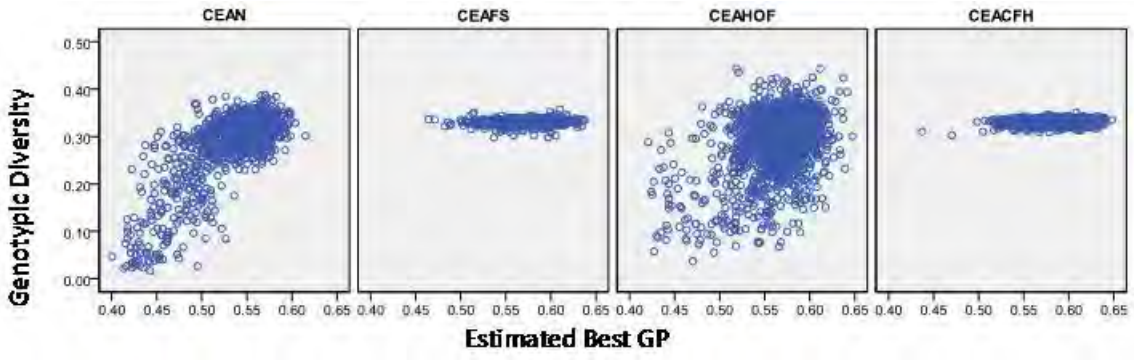

Figure 6.11: Scatter plot of estimated best GP versus genotypic diversity for CEAN, CEAFS, CEAHOF and CEACFH algorithm. Each point is a mean value of particular mutation rate and there were 40 mutation variations, there were 60 runs for each mutation rate.

6.3.4.2 Relationship between Genotypic Diversity and Estimated Average GP

Figure 6.12 consists of four scatter plots representing the relationship between the *genotypic diversity* and *estimated average GP* in each of the four algorithms. The CEAN chart shows that the points are vertically lined up which implies that there is not

much change in the *estimated average GP* as *genotypic diversity* changes from values near 0 to 0.4. The *estimated average GP* appears to concentrate around a band ranging from approximately 0.4 to 0.45 with more points towards the higher end for *genotypic diversity*. In the CEAFS and CEACFH, the distribution of points is similar to the relationship between *genotypic diversity* and the *estimated best GP* except for the range being between 0.4 and 0.55. In the CEAHOF, the points are much more scattered, with concentration of points in the middle region. There also appears to be a negative correlation between the two measures.

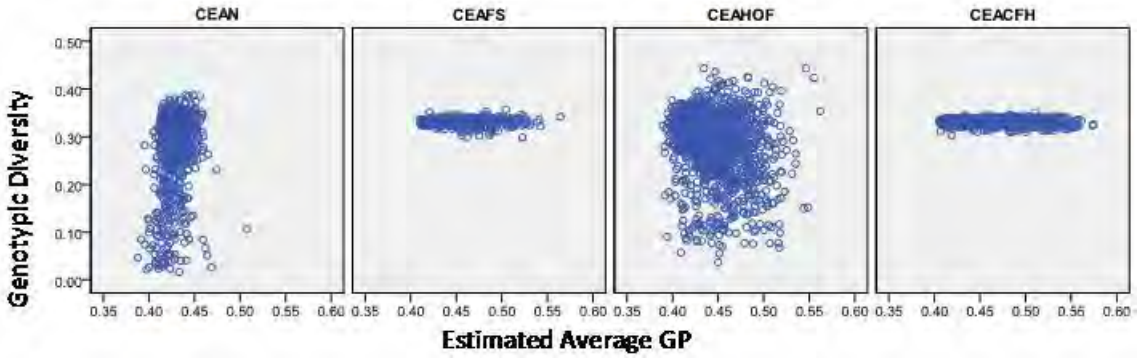

Figure 6.12: Scatter plot of estimated average GP versus genotypic diversity for the CEAN, CEAFS, CEAHOF and CEACFH algorithm. Each point is a mean value of particular mutation rate and there were 40 mutation variations, there were 60 runs for each mutation rate.

6.3.4.3 Relationship between Genotypic Diversity and CEMD

For the purpose of visualizing the relationship between the *genotypic diversity* and CEMD, Figure 6.13 is presented. In the CEAN and CEAHOF, there is a negative correlation; higher *genotypic diversity* is associated with lower CEMD. This strong negative correlation between these two factors is clearly visible in the scatter plot. In the CEAFS and CEACFH, both variables, the *genotypic diversity* and CEMD, are concentrated in a very small region, in the left-hand upper corner. Bearing in mind that the points were collected when the *mutation rates* were varied, this indicated that change in the *mutation rate* neither change *genotypic diversity* nor the *best GP*.

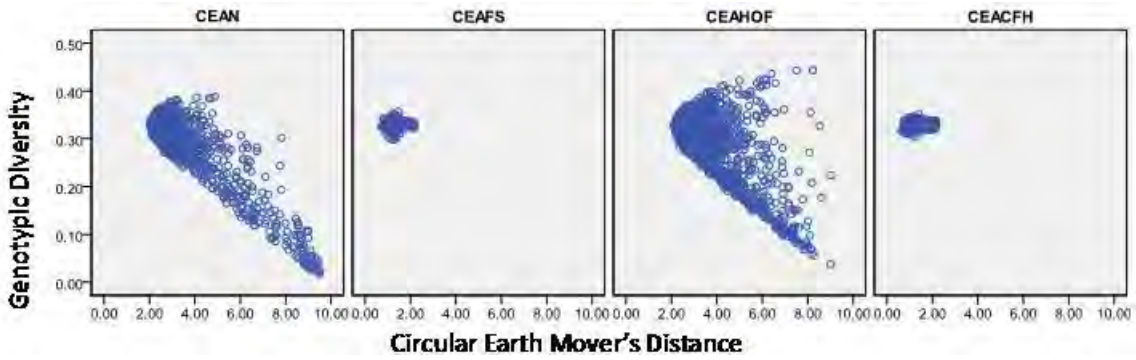

Figure 6.13: Scatter plot of CEMD versus genotypic diversity for the CEAN, CEAFS, CEAHOF and CEACFH algorithm. Each point is a mean value of particular mutation rate and there were 40 mutation variations, there were 60 runs for each mutation rate.

6.3.4.4 Relationship between Estimated Best GP and CEMD

Figure 6.14 consists of four scatter charts representing each algorithm, demonstrating the relationship between the *estimated best GP* and CEMD. In the CEAN, when the *estimated best GP* decreases, the CEMD increases, clearly showing a negative correlation. Concentration of points can be found in the region bounded by *estimated best GP* having values between 0.5 to 0.6 and CEMD having values between 2 to 4.

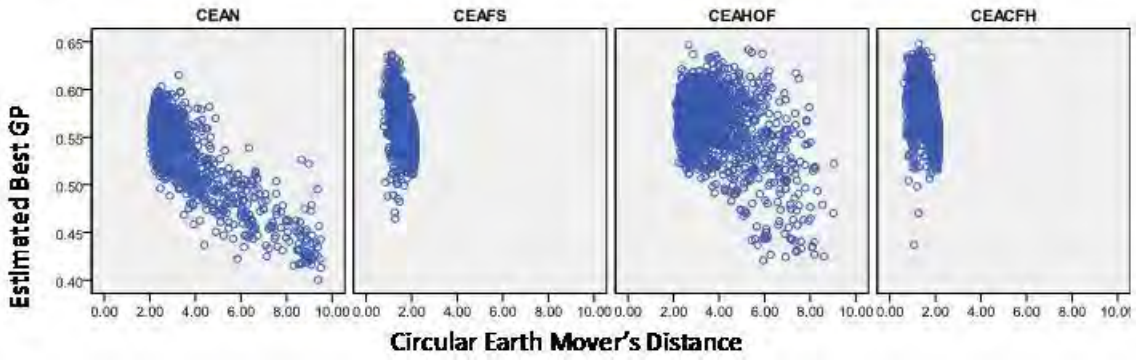

Figure 6.14: Scatter plot of estimated best GP versus CEMD for the CEAN, CEAFS, CEAHOF and CEACFH algorithm. Each point is a mean value of particular mutation rate and there were 40 mutation variations, there were 60 runs for each mutation rate.

In the chart associated with CEAHOF, negative correlation is not so clearly shown, with concentration of points can be found in the region bounded by *estimated best GP* having values between 0.5 to 0.65 and CEMD having values between 2 to 5. The points in the

two other charts for the CEAFS and CEACFH are similar, showing negative correlations. The distribution is concentrated in the upper left-hand quadrant with CEMD having a narrow range around 2 and estimated best GP having values between 0.5 and 0.63.

6.3.4.5 Relationship between Estimated Average GP and CEMD

The visualization of the relationship between the *estimated average GP* and CEMD are depicted in Figure 6.15. In the CEAN algorithm, there is a very weak correlation (approximately a horizontal line can be fitted to the points) between CEMD and the *estimated average GP*. In the CEAHOF, points are much more scattered and appears to be positively correlated. In the CEAFS and CEACFH, the points are almost vertically lined up, showing a strong negative correlation (low CEMD and high estimated average GP versus high CEMD and low estimated average GP). The CEMD values are within a very narrow range around 1.

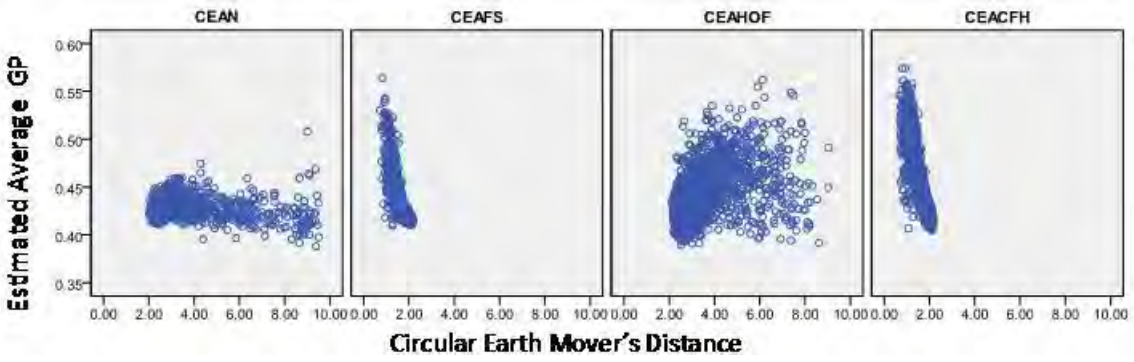

Figure 6.15: Scatter plot of estimated average GP versus CEMD for the CEAN, CEAFS, CEAHOF and CEACFH algorithm. Each point is a mean value of particular mutation rate and there were 40 mutation variations, there were 60 runs for each mutation rate.

6.4 Conclusion

A test problem proposed in this chapter demonstrated a multimodal problem for a competitive CEA. The *n-peak* problem was defined and CEAs were used to optimize the problem by using a naïve CEA and by integrating three variants on it. The challenges for CEAs in the *n-peak* problem were to detect the known number of peaks that exists in the search space. A performance measure, GP, was used to evaluate the

algorithms‟ generalizing ability. Other performance measures such as CEMD, PR and SR were used to evaluate the algorithms‟ peaks detection ability.

In this 5-peak multimodal domain, the CEACFH appeared to be the best algorithm that produces populations with the best generalizing ability and with individuals located at more of the peaks than the other three algorithms. It was found that the performance of CEACFH was high when a moderately low level of *mutation rate* (near 5%) was applied.

A strong correlation between *genotypic* and *phenotypic diversity* shows that they shared common properties. Additionally, the FS technique was effective in maintaining high diversity in populations and experiments in increasing *mutation rates* in CEAFS and CEACFH shows that they have little or no impact on diversity and quality in the evolved population.

Importantly, the result demonstrates that even the naïve algorithm, CEAN, when utilized with sufficient diversity in the population, increased its ability in detecting multiple optima. Although the CEAN and CEAHOF‟s performance improved with higher *mutation rates*, they were unable to perform as well as the algorithms with FS where diversity is biased towards fitter solutions. Algorithms that incorporated fitness sharing perform well in terms of identifying multiple peaks in this domain.

7 CEAs for Red Teaming

Chapters 5 and 6 detailed the capabilities of CEAs in addressing the intransitive number and multimodal problems respectively. Those chapters demonstrated how a naïve CEA and three variants performed in dealing with intransitivity and multimodality, which are also characteristics of RT. This chapter aims to evaluate these four algorithms‟ performance in terms of their generalising ability in RT applications. Two RT scenarios that were created using MANA (section 2.1.2.2.5) are evaluated. In addition, experiments were conducted to test the ability of each of the four algorithms in locating more local optima. To conduct this test, *a peak detection technique* (section 4.1.2.1) was used. Additionally, the strategies generated from each of the four algorithms were evaluated.

This chapter begins with a description of the two MANA scenarios: anchorage protection and coastline protection. Section 0 presents the experimental setup. Sections 7.3 and 7.4 show the results and strategies evolved for anchorage protection and coastline protection scenarios respectively.

7.1 MANA Scenarios

Two maritime security scenarios, anchorage protection (Han, et al., 2007) and coastline KIN protection (Chua, et al., 2008) as discussed in the pilot study of chapter 3, were chosen to test the performance of CEAs in RT.

These scenarios are of interest as in many Southeast Asian countries maritime security threats such as piracy, terrorism against maritime plant installation including oil and gas platforms increase the demand for security initiatives (Liss, 2007). Some countries, including New Zealand, Singapore, Australia and the United States, have been introducing new techniques to secure their maritime systems to prevent unauthorized activities (Ilachinski, 2000; Lauren, 2002; Yang, et al., 2006). One of the techniques that they use is simulation software, which is utilized to develop plans and to test them to ensure maritime security. Details for the two scenarios chosen for this study are presented in the sections 7.1.1 and 7.1.2 respectively.

7.1.1 Anchorage Protection

The first scenario involves protection of a safe anchorage area for commercial vessels. Sea pirates, terrorism and unlawful interference with maritime transport systems have led countries to introduce an organized anchorage security plans to ensure the safety of maritime transport systems (Liss, 2007). Anchorage protection includes anchorage security risk assessment and the implementation of security plans to address identified risks. Additionally, it also covers escorting important vessels and consideration of all security threats (Han, et al., 2007).

The anchorage protection scenario was initially investigated by Han et al. (2007). Three categories of boats are considered: attacker, defender and neutral boats. Figure 7.1 depicts the initial maritime security scenario which was designed by considering a 100 x 50 nautical mile (NM) area of operation (AO). This area was represented using a (400 × 200) grid area for which the origin (0, 0) was at the left top corner of the scenario. The anchorage is designed to be a 30 x 10 NM area, which is protected by the blue security patrol vessels (PVs). The green boats are neutral commercial boats anchored in the protected area. The blue boats patrol against the threat to the anchorage and the red boats are attackers which try to capture or destroy the green boats by penetrating the security provided by the blue.

Five red boats, three at the top and two at the bottom, can move in any direction with low range weapons and a short detection range. The objective of the red boats (attackers) is to penetrate the blue security system to reach the neutral commercial boats that are guarded by the blue PVs. The blue boats are patrolling to protect the neutral commercial boats under threat from the red boats. The blue boats are heavily equipped and carry sophisticated weapons. Each blue boat has two triangular patrol routes within which they perform surveillance. In the scenario the neutral commercial boats are not armed and do not move.

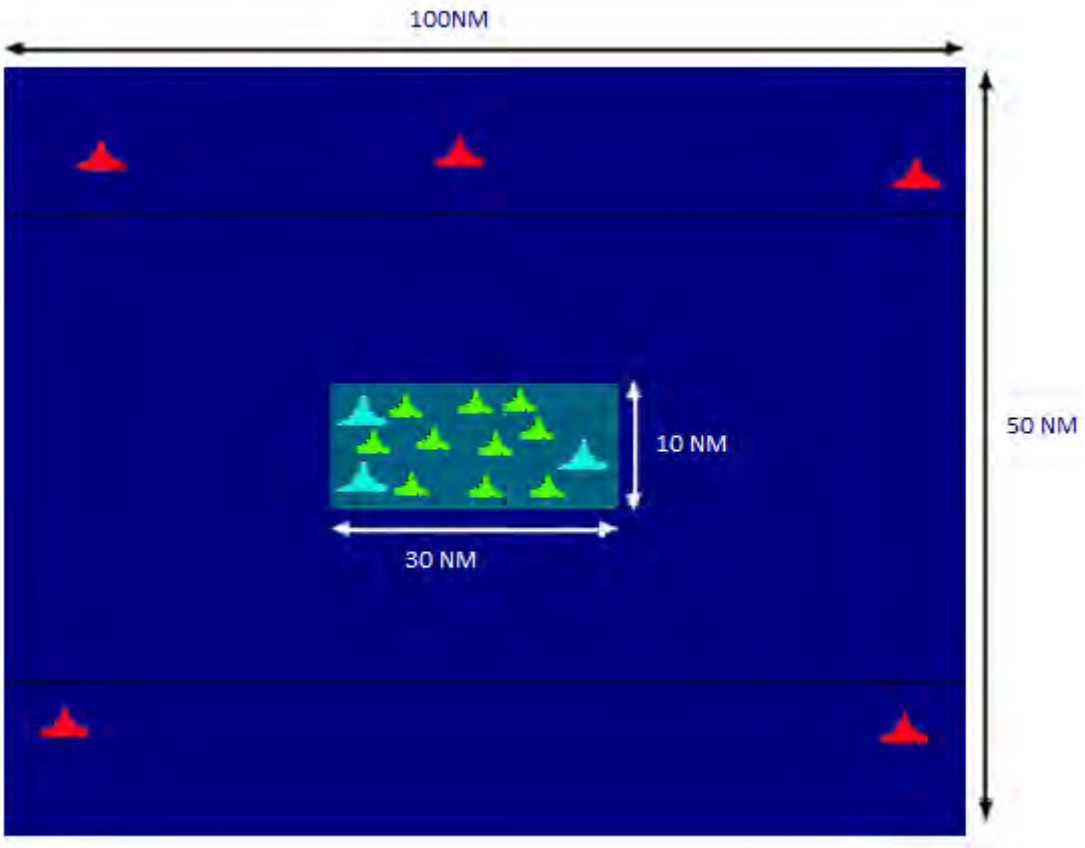

Figure 7.1: Anchorage protection scenario (source: Choo et al., 2009)

When optimizing this scenario, the blue and red fitness was calculated differently to account for the different priorities of each team. Features of the scenario and the parameters considered for the optimization process are depicted in Table 7.1. The parameters were the blue and red boats‟ characteristics were evolved; resulting in new strategies. When a scenario was run multiple times, the output contained information of how many times the blue and red team won and also the blue and red attrition. For the blue team, MOEs are defined as depicted in Equation (7.1).

Table 7.1: Anchorage scenario related information

Scenario Features			
Features	**Red boats**	**Blue boats**	**Commercial boats**
Number of boats	5 (3 top, 2 bottom)	3	10
Movement route	Any direction	Triangular	Immobile
Movement speed	16 knots	16 knots	0
Detection range	2 NM	6 NM	Nil
Parameters Studied			
Red Parameters		**Min Value**	**Max Value**
Home position[b] (x,y)[a]			
-For 3 Red Craft		(0,0)	(399,39)
-For 2 Red Craft		(0,160)	(399,199)
Intermediate Waypoint[c]		(0,40)	(399,159)
Final Waypoint[d] (x,y)		(139,79)	(259,119)
Blue Parameters			
Home position (x,y)		(70,40)	(329,159)
Waypoints 1 and 2 (x,y)			

[a](x,y): Grid locations (400 × 200)

[b]Home Positions: the positions at which agents appear when the scenario begins

[c]Intermediate Waypoints: the positions that define the routes to reach the destinations

[d]Final Waypoints: the positions of the final destinations (goals).

The emphasis was set by assigning a weighting value in maximizing red casualties. The red team‟s MOEs are depicted in Equation (7.2). The aim was to maximize the green and blue casualties and reduce the red casualties. In MANA, the simulation termination condition was set to 1000 simulation steps, or all red or blue agents destroyed. Since the weighting values (4 in equation (7.1) and 1.5 and 3 in equation (7.2)) are arbitrary, the analyst will determine appropriate values according to the context. A weighting value of 4 for the blue team (Equations (7.1)) and weighting values of 1.5 and 3 for the red team (Equation (7.2)), were found to be suitable in this scenario context.

$$blue\ score = 4(rc + 1) - \frac{gbc}{4} \quad \textbf{(7.1)}$$

$$red\ score = 1.5(gbc + 1) - \frac{rc}{3} \quad \textbf{(7.2)}$$

where,

rc = mean red casualties

gbc = green and blue casualties

7.1.2 Coastline Protection

The second scenario concerns protection of a key installation from attack by sea. The KIN coastline protection scenario (Chua et al., 2008) includes protection of KINs along with stopping anyone who tries to enter the coastline unlawfully. The scenario demonstrates the threat to KIN protection from non-military boats which try to penetrate regular surveillance carried out by three blue boats. The fairly low speed blue boats patrol a specific area of the coastline with low level weapons. Conversely, the red boats are without weapons and try to penetrate the blue patrol to get to the land using different escape tactics and routes (Chua, et al., 2008). In the scenario, there are three KINs and three blue patrol boats. Each blue boat has its patrol route which it constantly follows.

This scenario has been studied in the pilot study described in chapter 3. However, the scenario developed by Chua, et al. (2008) is slightly modified here in order to make it suitable for use in CEA environments. The modified blue surveillance route and KINs, along with the initial positions of the red boats are depicted in Figure 7.2. There are three red boats whose objectives are to reach the coastline while avoiding the blue patrol boats. This scenario was designed in a grid area (200 × 400) for which the origin (0, 0) was at the left top corner of the scenario.

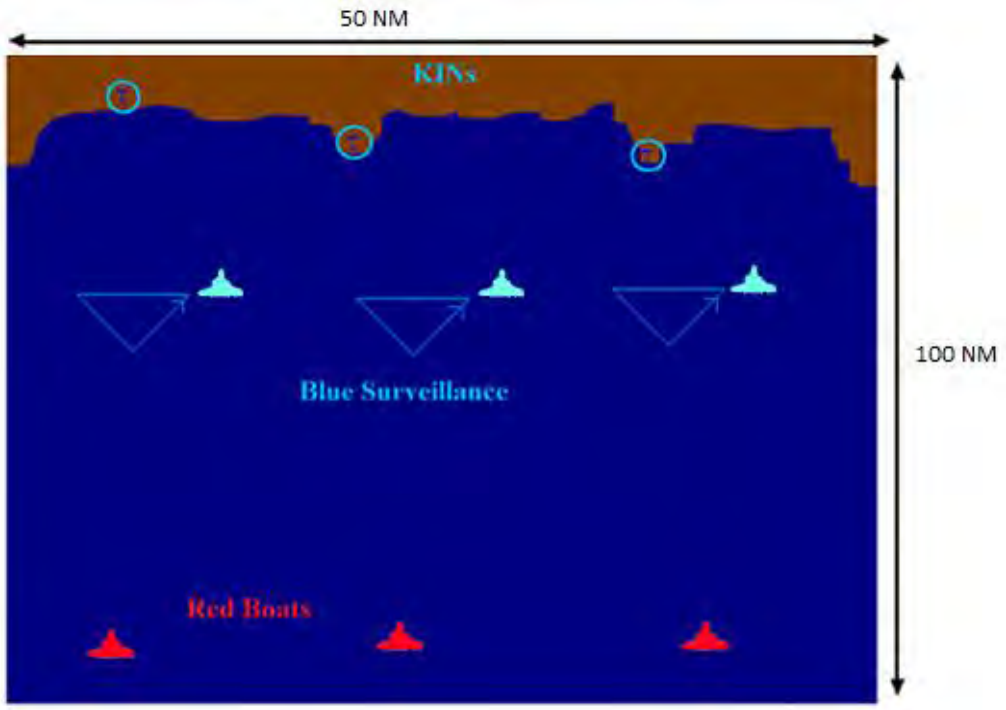

Figure 7.2: Coastline scenario (Source: Chua et al., 2008)

The blue and red fitness was calculated differently by prioritizing factors relevant to each team. The features of the scenario and the associated parameters that were optimized are outlined in Table 7.2. Similar to the anchorage protection scenario, these parameters are related to the blue and red boats‟ characteristics, which are evolved during the optimization process to produce new strategies. When a scenario is run multiple times, an output file contains information of how many times the blue and red team wins and also the ratio of the blue and red attrition. MOEs for the blue team were defined using Equation (7.3) to calculate the interaction score. The emphasis was set by using weighting values for stopping the red team reaching their destination. The red team‟s MOEs are depicted in Equation (7.4). One of its aims was to maximize the achievement of its goal (that is, breaking the blue boat patrolling tactics by getting at least one red boat to the land) and another was minimizing red casualties. As mentioned in the anchorage protection scenario, the weighting values are not pre-defined. Thus, the analyst will determine an appropriate value according to the context. Suitable weighting values found for this scenario were values of 0.5 and 10 for the blue team (Equations (7.3)) and 10 and 1.5 for the red team (Equation (7.4)). In MANA, the simulation termination condition was set to 1000 simulation steps, or all red agents destroyed, or any red agent achieving the goal of reaching the land.

Table 7.2: Coastline scenario related information

Scenario's Features			
Type of boats	**Red boats**	**Blue boats**	**KINs**
Number	3	3	3
Movement route	Any direction	Triangular	Immobile
Movement speed	25 knots	15 knots (patrol) 25 knots (chase)	0
Detection range	1NM	3 NM	5 NM
Parameters Studied			
Parameters for each red boat		**Min Value**	**Max Value**
Home position (x,y) (y non-evolving)		(0,350)	(199,350)
Final waypoint(x,y) (y non-evolving)		(0,40)	(199,40)
Parameters for each blue boat			
Home Position(x,y) (y non-evolving)		(0,140)	(199,140)
Waypoints 1 (x,y) (y non-evolving)		(0,140)	(199,140)
Waypoints 2 (x,y) (y non-evolving)		(0,190)	(199,190)
Common Parameters			
-Alive Enemy[a]		-100	100
-Alive Friends1[b]		-100	100
-En Threat 1 (High)[c]		-100	100
-Next Waypoint[d]		-100	100
-Alternate Waypoint[e]		-100	100
-Movement Speed[f]		0	100

[a]Alive Enemy: Measure of attraction or repulsion to the agent with enemy allegiance

[b]Alive Friends1: Measure of attraction or repulsion to the agent with same allegiance

[c]En Threat 1 (High): Measure of attraction or repulsion to the agent with enemy allegiance with low threat level

[d]Next Waypoint: The value determines the dedication of agents in achieving the goal

[e]Alternate Waypoint: The value determines how determine agents" are in moving in their pre-determined waypoint

[f]Movement Speed: The value determines the number of grid agents move in a given time step.

$$blue\ score = 0.5\ \times bw + \ 0.5\ \frac{rc}{nra} + 10.0\ (1 - rw) \tag{7.3}$$

$$red\ score = 10\ \times rw + 1.5\ \frac{nra - rc}{nra} + 1 - bw \tag{7.4}$$

where,

bw = proportion of blue wins

rw = proportion of red wins

rc = mean red casualties

nra= number of red agent

7.2 Experimental Setup

Experiments were set up to compare the performance of a naïve CEA and variants (CEAN, CEAFS, CEAHOF and CEACFH), in optimizing the two RT scenarios using an appropriate set of performance measures. As in the intransitive number problem and the multimodal problem, one aspect of performance is generalisation performance, which measures how well solutions found for one side in a contest, learned via a CEA, generalise to compete well against arbitrary strategies for the other side. Specific to RT optimization, another relevant aspect of performance is to identify how many locally optimal solutions are found in the final population from each algorithm (i.e. how many local optima are located). Subsequently, for both the blue and red teams, the solutions (i.e. parameters associated with the scenario) generated by each of the four algorithms were used to instantiate the scenario and their strategies were examined by running a simulation associated with the scenario.

To investigate the effect of diversity maintenance via fitness sharing and/or mutation, and the effect of using an archive in the form of a HOF, all four algorithms were tested with *mutation rates* of 10%, 20%, 40% and 60%, on each of the two scenarios chosen for this study. In chapters 5 and 6, 40 different *mutation rates* were applied; however, due to the large computation time required to simulate the red teaming scenarios, only four *mutation rates* were evaluated in RT optimizations. For a small and large *mutation rate,* a value of 10% and 60% were chosen respectively. Additional evaluation was

required to see the effect of *mutation rates*; thus, two intermediate *mutation rate*s 20% and 40% were also evaluated.

The parameters used in these experiments are listed in Table 7.3. Each of the four algorithms was executed 15 times to account for statistical variation. For calculating the GPs and diversity of the population, the last 10 generations‟ values were averaged from each execution of the algorithm and these average values from 15 runs were again averaged.

Similar to the investigation in chapters 5 and 6, in every execution, in each generation, generalisation ability was measured by using GPs (average and best) (see section 4.5.1) and diversity of the population was measured on the basis of genotypic (see section 4.5.5.1) and phenotypic (see section 4.5.5.2) diversity. The data for GPs and two diversity measures were examined in four ways: (1) A profile (interaction) plot of each measure against *mutation rates,* (2) ANOVA test, (3) Correlation analysis for measures associated with each of the four algorithms and (4) Scatter plots to visualise the relationship between quality and diversity.

Table 7.3: CEA parameters used in the algorithms studied

Properties	**Algorithms/Values**
Population size	15 in each population
Gene value	0 to 100
Crossover	Single point
Crossover rate	60%
Mutation	Polynomial
Mutation rates	10%, 20%, 40% and 60%
Selection	Stochastic universal sampling
Generations	50
Number of runs	15
Niche radius	80 (best value suggested by experimentation)
HOF sample size	15 (equal to population size)
MANA simulation run	15 per evaluation

Initial gene values were randomly generated values between 0 and 100. As mentioned in section 3.1.5, the scenario needs to be executed many times via MANA which increases the computational time. Thus, MOEs were calculated by executing 15 simulation runs in each evaluation of individuals. On the basis of MOEs, the score is calculated for the

interaction between two individuals. Since the RT problem is asymmetric the two teams are evaluated using different formulas. For the anchorage protection scenario, a score is calculated using Equations (7.1) and (7.2) for the blue and red team respectively. Likewise, a score for the blue and red team, in the coastline protection scenario, is calculated using Equations (7.3) and (7.4) respectively. Each score received by an individual, when it is evaluated against all the members from the competing population, are averaged for calculating fitness as shown in Equation (2.1). In terms of FS, an empirical study was carried out to investigate suitable values for niche radius. The niche value of 80 was found to be the best one for this domain. The experiment for each algorithm with a specific *mutation rate* was run 15 times to account for statistical variation.

7.3 Results and Analysis for the Anchorage Protection Scenario

In this section, the performance of the four algorithms in optimizing the anchorage protection scenario is presented in terms of the GP, diversity and testing for the existence of multiple optima in the populations produced by these algorithms. Similar to chapters 5 and 6, the generational plots for the two competing sides are presented section 7.3.1.

7.3.1 Analysis of GPs in the Blue and Red Teams

This section presents the analyses of the blue and red teams‟ *estimated best GP* via convergence plots in Figure 7.3. The *estimated best GP* of the populations from CEAN, CEAFS, CEAHOF and CEACFH were individually calculated for each of the two teams. In the execution of each algorithm, there were 50 generations. To account for statistical variation, each algorithm was executed 15 times and the average of these 15 runs was calculated in each generation for each *mutation rate* used in this study. In Figure 7.3, the x-axis is the number of generations and the y-axis represents the *estimated best GPs*.

Figure 7.3 (a) and (b) show the *estimated best GP*s of the blue and red teams respectively when 10% *mutation rate* was applied. In the blue team, the *estimated best*

GP value of the solutions associated with CEACFH was slightly lower than the other three algorithms. The *estimated best GP* associated with CEAFS, CEAHOF and CEAN were similar. In the red team, the *estimated best GP* associated with all four algorithms was similar. Due to the fact that the computation time was high only 15 simulations per evaluation and 50 generations were run. This causes the graphs to appear less smooth than for the scenarios in previous chapters and at low mutation rates, some algorithms appear not to have converged. In realistic scenario, decisions must be made in a reasonable time, with limited computational time. So an algorithm that finds the optima faster has an advantage.

Likewise, Figure 7.3 (c) and (d) show blue and red‟s *estimated best GP*s at a *mutation rate* of 20%. Similar to Figure 7.3 (a) and (b), CEACFH did not perform as well as in comparison to the other three algorithms in the blue team whereas the performance of all four algorithms was similar in the red team. Figure 7.3 (e) and (f) demonstrate the two teams‟ estimated *best GP*s at a *mutation rate* of 40%. The performance of each of the four algorithms was similar to those at the *mutation rate* of 20% for both the teams. At a *mutation rate* of 60% (Figure 7.3 (g) and (h)) the four algorithms associated with the blue and red team also show the similar performance to those obtained at a *mutation rate* of 20%.

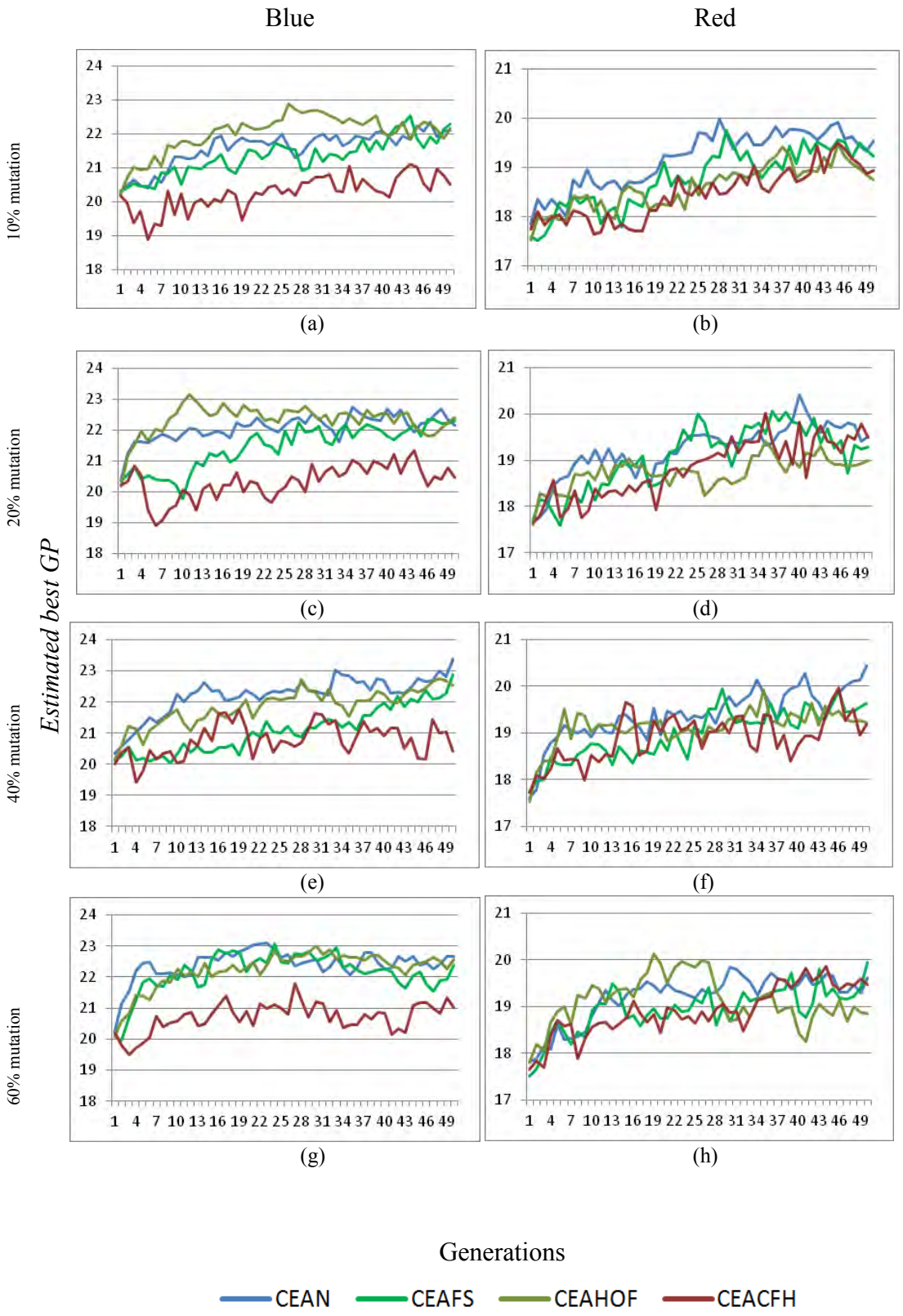

Figure 7.3: Convergence plots showing the estimated best GPs of the four algorithms for the blue (a, c, e, g) and red (b, d, f, h) team for mutation rates of: (a, b) 10%, (c, d) 20%, (e, f) 40%, (g, h) 60%

To further examine the result shown in the convergence plots, statistical test were conducted. The ANOVA procedures and interaction plots associated with the *estimated best GP* and the *estimated average GP* are presented in the following sub-sections.

7.3.1.1 Analysing Estimated Best GP

Figure 7.4 shows the interaction plots of the *estimated best GP* associated with CEAN, CEAFS, CEAHOF and CEACFH versus *mutation rates* for the blue team. The *mutation rates* of 10%, 20%, 40% and 60% were employed to investigate the *estimated best GP* in this study. The x-axis is *estimated best GP* and y-axis is *mutation rates.*

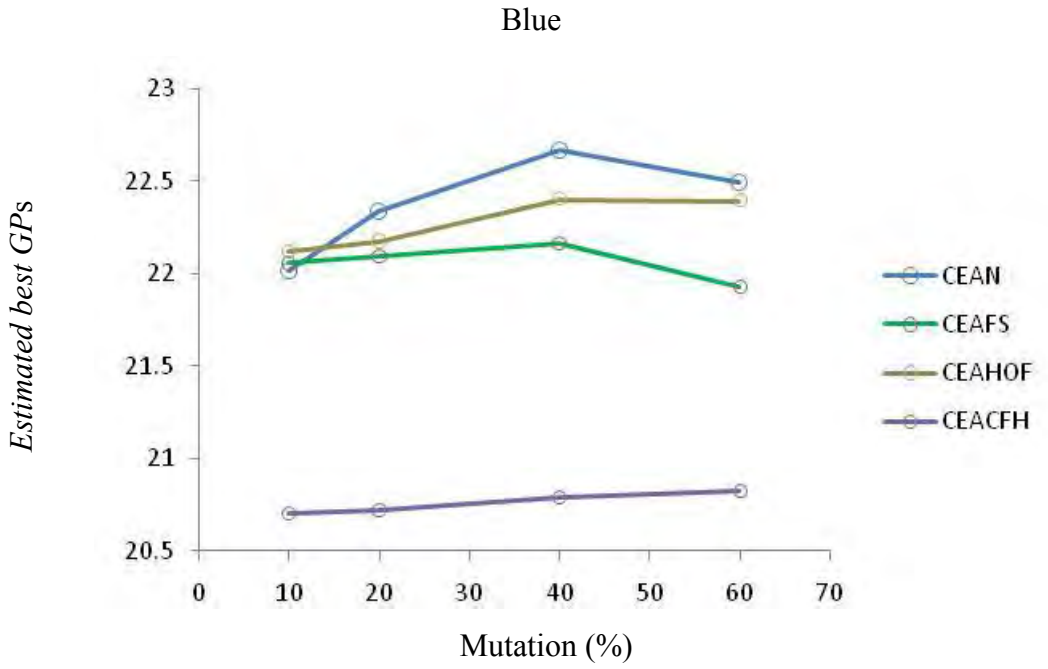

Figure 7.4: Interaction plots of estimated best GPs (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate, for each of the 4 algorithm variants in the blue team

Figure 7.4 also shows that the *estimated best GP* associated with CEACFH was relatively lower than the three other algorithms, which confirms the result depicted in Figure 7.3. The performance of CEAN, CEAHOF and CEAFS was similar, and with values within a band width of 22.0 to 22.5. The performance of CEAN at the *mutation rate* of 40% was slightly higher than the other three algorithms in terms of *estimated*

best GP value. From the interaction plot, it is hard to evaluate whether there is an effect of *mutation rate* on *estimated best GP* and it appears that there is no interaction.

Two-way ANOVA was conducted to analyse *estimated best GP* with four levels of *mutation rate* and four types of *algorithm used*. The table associated with ANOVA is depicted in Appendix C.1. ANOVA shows that the overall model is statistically significant, ($F(15, 224) = 4.835$, $p < 0.05$, Partial Eta Squared = 0.245). The effect of *algorithms used* was also statistically significant ($F(3, 224) = 23.03$, $p < 0.05$, Partial Eta Squared = 0.236). However, *mutation rate* $F(3, 224) = 0.592$, $p = 0.621$, Partial Eta Squared = 0.008 and the interaction *(algorithms used × mutation rate)* ($F(9, 224) = 0.184$, $p = 0.996$, Partial Eta Squared = 0.007) were not statistically significant.

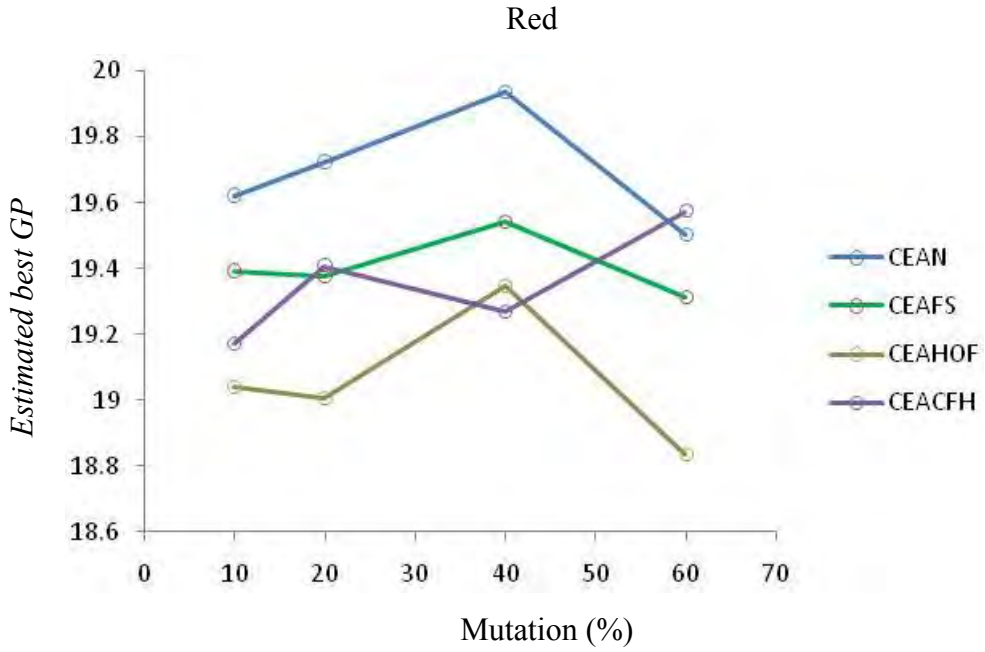

Figure 7.5: Interaction plots of estimated best GPs (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate, for each of the 4 algorithm variants in the red team

Similar to the blue team, the interaction plot of *estimated best GP* associated with each of the four algorithms versus *mutation rates* for the red team is depicted in Figure 7.5. The performance of each of the four algorithms in terms of *estimated best GP* value is similar, with points associated with each of the algorithms falling within a band width of 18.8 to 19.9. As in the blue team, there does not appear to be any interaction and it

was hard to evaluate whether the *mutation rate* has any influence on *estimated best GP* for the red team.

Two-way ANOVA was conducted to analyse *estimated best GP* with four levels of *mutation rate* and four types of *algorithms used*. A table associated with ANOVA procedure is depicted in Appendix C.2, which shows that the overall model is not statistically significant, (F (15, 224) = 0.973, p = 0.485, Partial Eta Squared = 0.061). The effect of the *algorithm used* was statistically significant (F (3, 224) = 3.394, p = 0.019, Partial Eta Squared = 0.043) but the effect size is small. However, *mutation rate* (F (3, 224) = 0.518, p = 0.670, Partial Eta Squared = 0.007) and the interaction *(algorithm used* × *mutation rate)* (F (9, 224) = 0.317, p = 0.969, Partial Eta Squared = 0.013) were not statistically significant.

Since the RT is an asymmetric problem, the fitness for the blue and red was measured using different fitness functions (see Equations (7.1) and (7.2)). Thus, the interaction plots for the *estimated best GP* of the blue and red team have different scales. While evaluating the two teams, it was found that the performance of CEACFH was non-competitive for the blue team whereas it was equally competitive for the red team. From this behaviour of CEACFH, it appears that the performance of an algorithm differs on the basis of the tasks employed in the scenario.

7.3.1.2 Analysing Estimated Average GP

This section demonstrates the effect of varying *mutation rates* on the *estimated average GP* for each of the four algorithms studied in this thesis. Figure 7.6 depicts an interaction plot of *estimated average GP*s versus *mutation rate* in all four algorithms for the blue team.

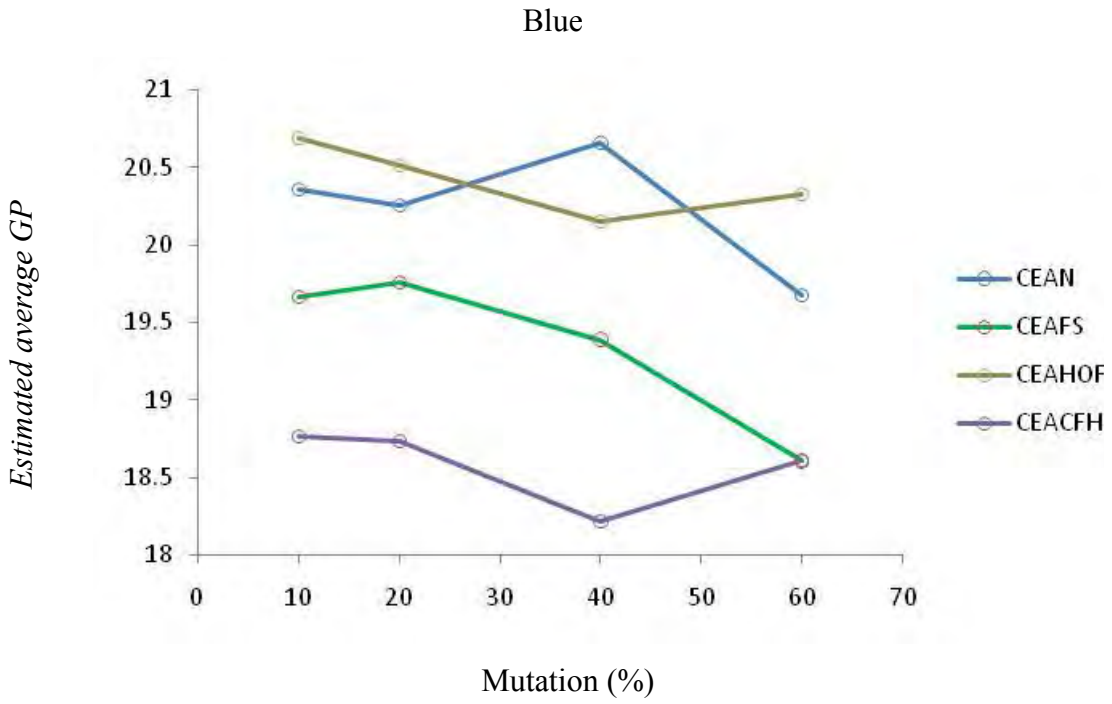

Figure 7.6: Interaction plots of estimated average GPs (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate, for each of the 4 algorithm variants in the blue team

Figure 7.6 shows that, as expected, *estimated average GP* is negatively influenced by *mutation rate* in all four algorithms. The relatively higher *estimated average GP* were found in the population of the CEAHOF and CEAN. The performance of the solutions associated with CEAFS was higher than that in the CEACFH except in the mutation rate of 60%, in which both these algorithms received similar *estimated average GP* value. However the performance of CEAFS is slightly lower in comparison to CEAN and CEAHOF.

As in section 7.4.1.1, *estimated average GP* was analysed by means of two-way between subjects ANOVA test with four levels of *mutation rate* and four types of *algorithm used*. The overall model was statistically significant, ($F(15, 224) = 6.483$, $p < 0.05$, Partial Eta Squared = 0.303). The interactions and *mutation rate* was not statistically significant, ($F(9, 224) = 0.923$, $p = 0.506$, Partial Eta Squared = 0.036) and ($F(3, 224) = 2.442$, $p = 0.065$, Partial Eta Squared = 0.032) respectively. However, the effect of *algorithm used* was statistically significant, ($F(3, 224) = 27.206$, $p < 0.05$,

Partial Eta Squared = 0.267). The table associated with ANOVA is depicted in Appendix C.3.

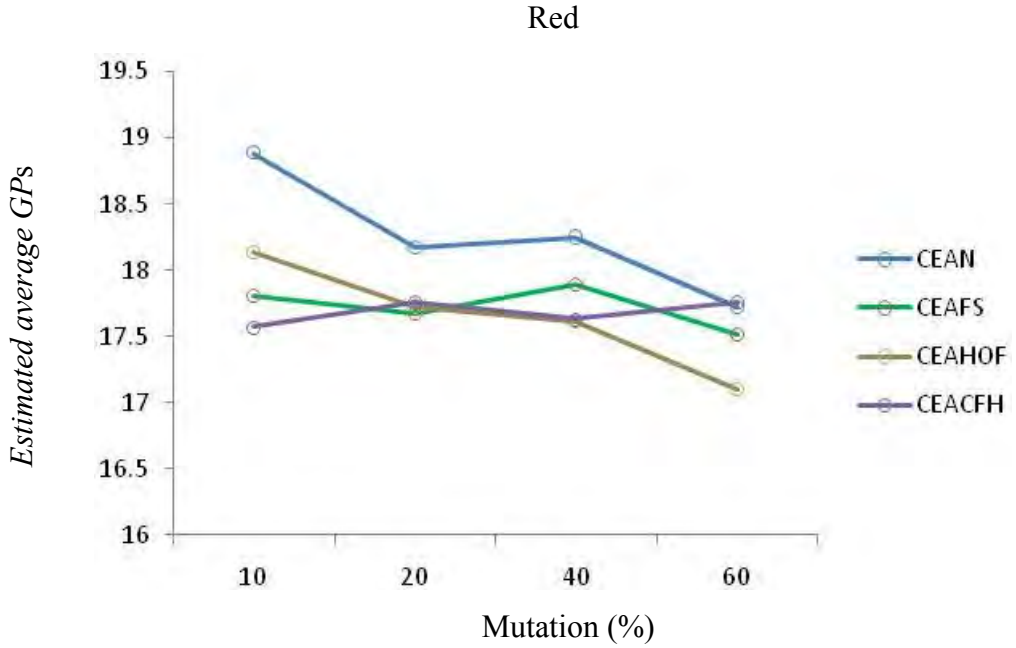

Figure 7.7: Interaction plots of estimated average GPs (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate for each of the 4 algorithm variants in the red team

An interaction plot from the red team is depicted in Figure 7.7. The *estimated average GP* associated with all four algorithms was similar when *mutation rate* of 10%, 20%, 40%, 60% was employed. The effect of the *mutation rate* is negative in CEAN and CEAHOF. However, the effect of the *mutation rate* is not clear in CEAFS and CEACFH.

Two-way ANOVA was conducted to analyse *estimated average GP* with four levels of *mutation rate* and four types of *algorithm used*. The ANOVA output shows that the overall model was not statistically significant for the red team, (F (15, 224) = 2.124, p = 0.10, Partial Eta Squared = 0.125). The effect of *algorithm used* and *mutation rate* was significant but the effect is small, (F (3, 224) = 4.552, p = 0.04, Partial Eta Squared = 0.057) and (F (3, 224) = 3.018, p = 0.031, Partial Eta Squared = 0.039) respectively. The interaction effect was also not statistically significant, (F (9, 224) = 1.017, p =

0.427, Partial Eta Squared = 0.039). The table associated with ANOVA is depicted in Appendix C.4.

In the anchorage protection scenario, the analysis of the GPs, on the basis of the ANOVA test, showed that there was no interaction effect on *estimated best* and *estimated average GP* in both the blue and red teams. As in the findings of chapters 5 and 6, there is a specific *mutation rate* where the *estimated best GP* and *estimated average GP* is optimum for each algorithm, supporting the argument that an appropriate *mutation rate* is required to enhance the quality of the population. It was expected that fitness sharing would enhance the performance of the coevolutionary algorithm in terms of generalisation ability. However, the performance associated with CEAFS was not better than the other three algorithms. The reason may be fitness sharing was based on the genomes rather than the behaviours. Maintaining diversity on the basis of behaviours is not practicable as there are no suitable methods to distinguish strategies except observing the simulation. A small change in genome may make large influence to a team‟s strategy, which may be the reason that the performance of CEAFS was not effective in this domain problem.

7.3.2 Local Optima Test for the Evolved RT Strategies

The pilot study showed that multimodality was one of the characteristics of RT problems. Thus, after measuring the algorithms‟ performance according to their GP, an additional test was conducted to detect the number of local optima found. Details of how the local optima technique works were presented in section 4.1.2.1.

In the experiment for the anchorage protection scenario, the population size was 15, which provided 15 evolved strategies for each team in each generation. For the local optima test, only the last generation evolved using a 40% *mutation rate* with each of the four algorithms was analysed. The evolved strategies from each of the four algorithms are depicted in Appendix E.5, Appendix E.6, Appendix E.7 and Appendix E.8 respectively. For each of these specific strategies, as in section 4.1.2.1, 10 random strategies were developed within a radius of 5 (each strategy (individual) in the population consists of genome of length of 18 for the blue and 30 for the red team. Each

gene is a real number ranging from 0 to 100.). The „small value" and epsilon value was set to 0.2 and 0.025 respectively. Table 7.4 shows the number of local optima detected in the final generation for all four algorithms.

Table 7.4: Local optima test result for the anchorage protection scenario

Algorithms	Local Optima
Blue	
CEAN	5
CEAFS	9
CEAHOF	8
CEACFH	4
Red	
CEAN	6
CEAFS	8
CEAHOF	8
CEACFH	5

Table 7.4 shows that the performance of the CEAFS and CEAHOF appeared to be the best out of four algorithms tested. These two algorithms were capable of detecting more local optima than the other two algorithms.

The above analysis showed that although all four algorithms (except CEACFH in the blue team) performed similarly in terms of generalising ability, CEAFS and CEAHOF locate more optima in comparison to CEAN and CEACFH. The peak finding performance of CEAFS was expected, as higher diversity encourages a more complete exploration of the search space, but the good performance of HOF was unexpected. An explanation for this result is a possible subject for future research.

As in the intransitive number problem and multimodal problem, experiments were conducted to investigate diversity of the populations produced by the algorithms when 10%, 20%, 40% and 60% *mutation rate* was applied. The following section presents details of this experiment.

7.3.3 Evaluating Diversity

In order to analyse diversity of the populations in all four algorithms employed in this study, interaction plots and ANOVA analysis associated with *genotypic* and *phenotypic diversity* are presented in the sections below.

7.3.3.1 Analysing Genotypic Diversity

This section explores *genotypic diversity* associated with the four algorithms. Interaction plots of the blue and red teams are depicted in Figure 7.8 and Figure 7.9 respectively. In both figures, the *genotypic diversity* associated with CEAFS and CEACFH was clearly higher than in two other algorithms. CEAN and CEAHOF, which do not use fitness sharing, received relatively low *genotypic diversity*. This indicated that, even in this domain, FS increased diversity in the populations. Based on the slope of the associated points in each of the four algorithms, an increase in *mutation rate* increases *genotypic diversity* of the population.

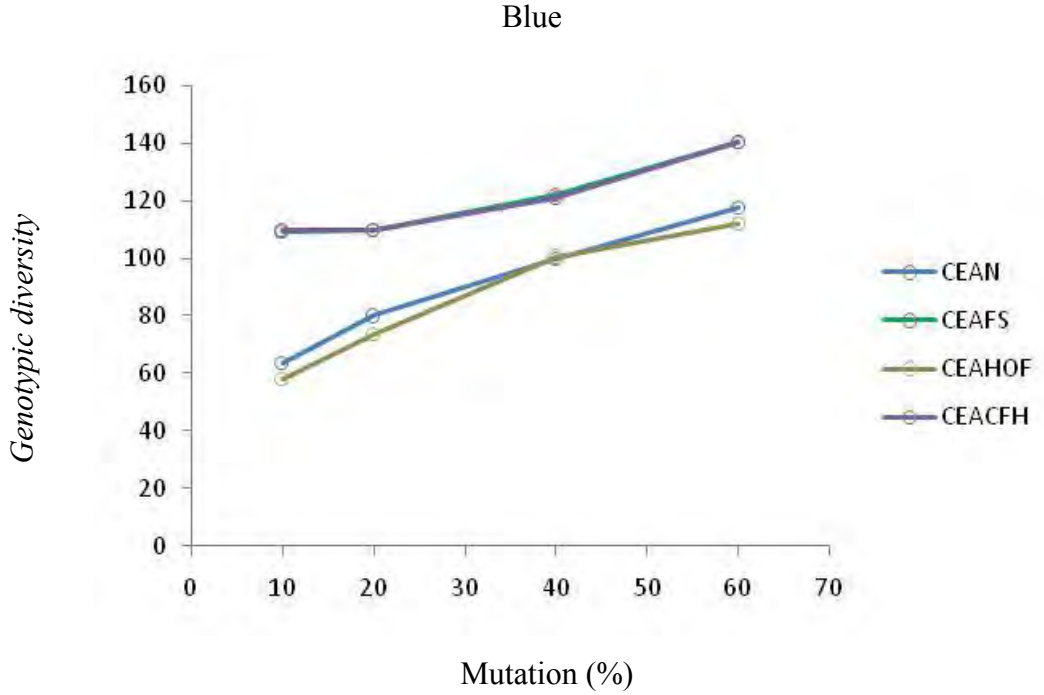

Figure 7.8: Interaction plots of genotypic diversity (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate for each of the 4 algorithm variants in the blue team

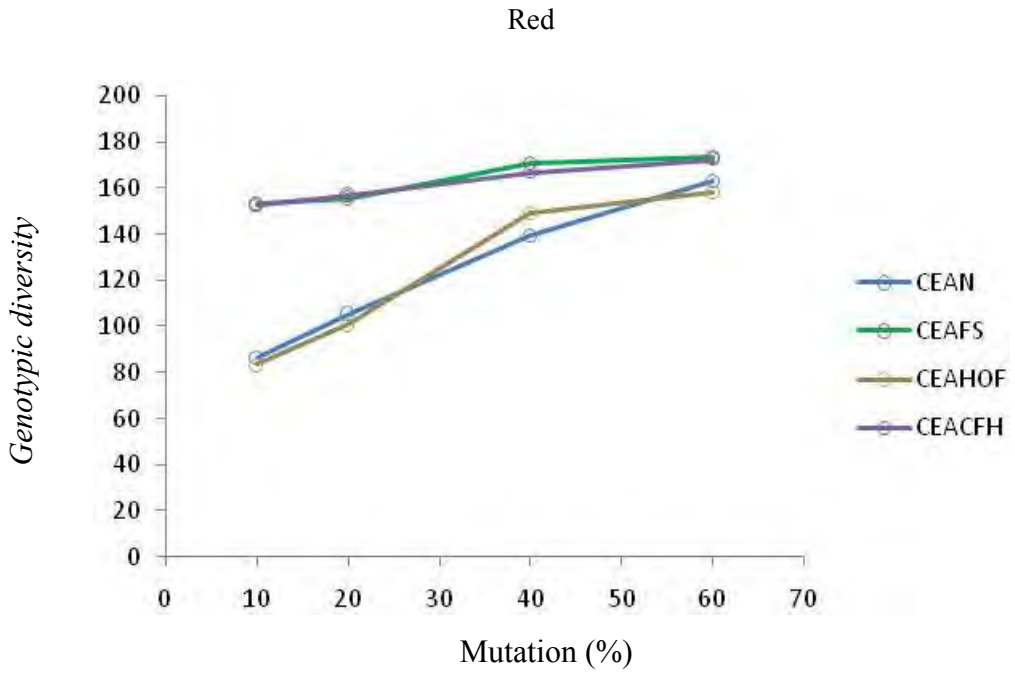

Figure 7.9: Interaction plots of genotypic diversity (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate for each of the 4 algorithm variants in the red team

For each team, the dependent variable, *genotypic diversity*, was analysed using a two-way between-subjects ANOVA with four levels of *mutation rate* and four types of *algorithm used*. The test for both the teams shows that the overall model and each of the independent variables involved, including the interaction effect, were statistically significant. The interaction effect for the blue and red team was (F (9, 224) = 8.455, p = 0.00, Partial Eta Squared = 0.254) and (F (9, 224) = 15.240, p = 0.00, Partial Eta Squared = 0.383) respectively. The tables associated with ANOVA are shown in Appendix C.5 and Appendix C.6.

7.3.3.2 Analysing Phenotypic Diversity

A similar analysis as in section 7.3.3.1 was conducted for *phenotypic diversity*. Figure 7.10 and Figure 7.11 show interaction plots of *mutation rates* versus *phenotypic diversity* for all four algorithms from the blue and red teams respectively. In the blue team, as *mutation rate* increases phenotypic diversity also increases. The phenotypic diversity of the blue team associated with each of four algorithms was similar at the

mutation rate of 40%. At the *mutation rate* of 60%, CEAN and CEAHOF received similar *phenotypic diversity*. In the red team, there is a smaller effect of changing *mutation rates* on phenotypic diversity as phenotypic diversity was spread within a band value of 1.918 to 1.962. However, CEAFS and CEACFH clearly received higher *phenotypic diversity* at the mutation rates of 40% and 60%.

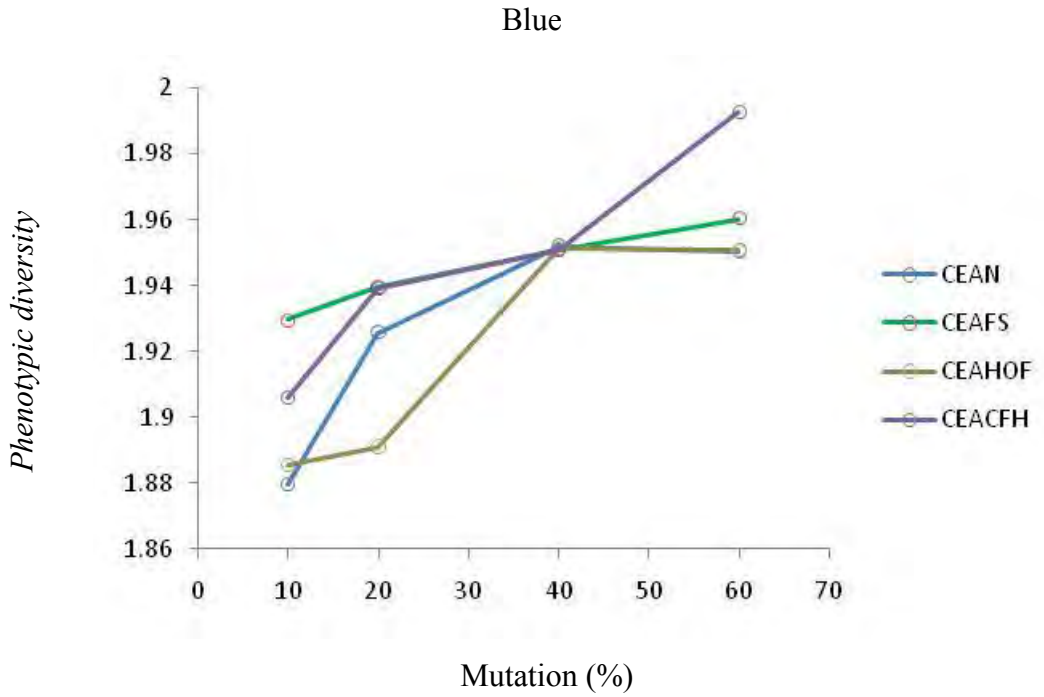

Figure 7.10: Interaction plots of phenotypic diversity (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate for each of the 4 algorithm variants in the blue team.

For each team, the dependent variable *phenotypic diversity* was analysed using two-way between-subjects ANOVA with four levels of *mutation rate* and four types of *algorithm used*. For the blue team, the overall model and the effect of *mutation rate* were significant but the effect of the interaction was not statistically significant. The interaction effect for the blue team was ($F(9, 224) = 0.977$, $p = 0.460$, Partial Eta Squared = 0.038). In the red team, neither the overall model nor the effect of any of the independent variables was statistically significant. The interaction effect was ($F(9, 224) = 0.302$, $p = 0.974$, Partial Eta Squared = 0.012). The tables associated with ANOVA are depicted in Appendix C.7 and Appendix C.8.

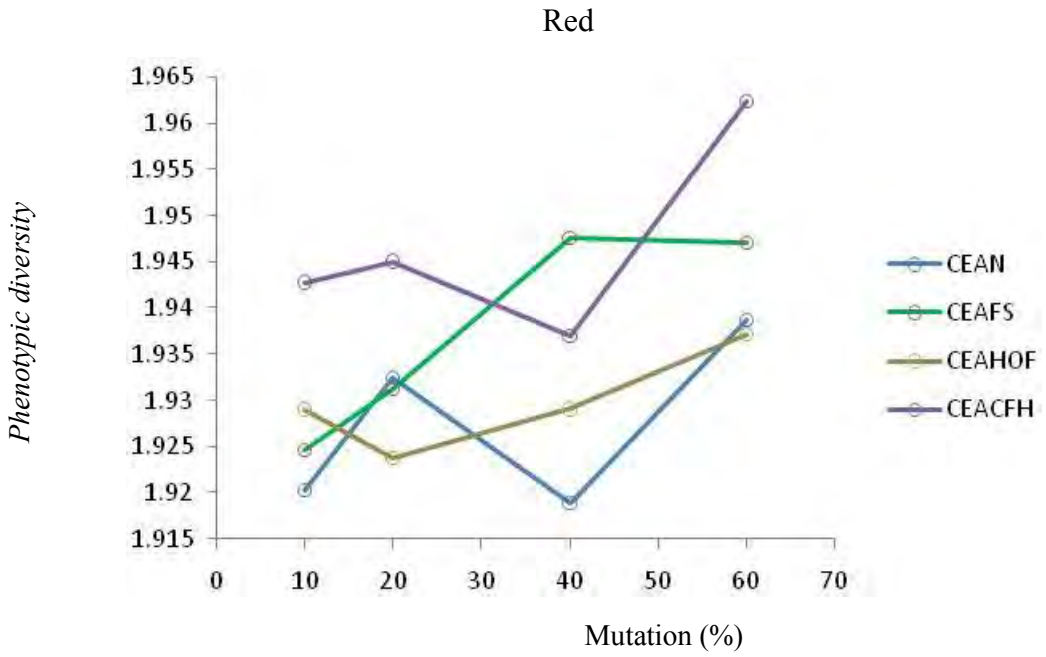

Figure 7.11: Interaction plots of phenotypic diversity (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate for each of the 4 algorithm variants in the red team.

The two diversity measures, *genotypic* and *phenotypic,* were inconsistent as *genotypic diversity* was highly influenced by the *mutation rate*; however, the effect of *mutation rate* on *phenotypic diversity* was low. While measuring *genotypic diversity,* CEAFS and CEACFH produce relatively more diverse population than CEAN and CEAHOF; however, while measuring *phenotypic diversity,* the performance of all four algorithms was similar.

Rather than examining diversity in isolation, its effect on GP is a useful performance indicator for the algorithms studied. Chong et al. (2008, 2009) analysed the relationship between the diversity of a population and GP. The authors found that diversity maintained by the implicit and explicit method in the population highly influenced GP. Therefore, the relationship between diversity and quality of the populations produced by all four algorithms are analysed in the following section to investigate whether the same findings hold true in this study.

7.3.4 Relationship between Diversity and GPs

This section examines the relationship between diversity and GP involved in this study. Correlation analysis was conducted for each of the two teams and each of the four algorithms individually. Thus, there are eight correlation tables (2 teams x 4 algorithms = 8) presented. Each table includes *genotypic diversity*, *phenotypic diversity*, *estimated average GP*, *estimated best GP* and *mutation rate*.

Table 7.5: Correlation between variables involved in the CEAN for the blue team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	**.331****	1			
Avg_GP	-.195	**-.270***	1		
Best_GP	.163	-.112	**.874****	1	
Mutation	**.904****	**.349****	-.174	.161	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

Table 7.6: Correlation between variables involved in the CEAN for the red team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	.179	1			
Avg_GP	**-.359****	**-.260***	1		
Best_GP	-.040	-.227	**.878****	1	
Mutation	**.897****	.088	**-.317***	-.026	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

Table 7.5 and Table 7.6 show correlations of the variables involved in the CEAN for the blue and red team respectively. The correlation between the *genotypic diversity* and *mutation rate* was positive and strong in both teams which indicated that increase in *mutation rate* is correlated with increased *genotypic diversity*. This finding supports the outcomes of the interaction plot and ANOVA output in section

7.3.3.1. The correlation between *genotypic* and *phenotypic diversity* was also positive but weak in both teams. This indicates that spread in genomes slightly affect the distribution of fitness of individuals in the population generated by the CEAN. The correlation between the *estimated best GP* and *average GP* was positive and strong which indicated that the increase in the overall performance of the population in this algorithm also improves the performance of the top individual (strategy) in both teams. The correlation between the *estimated best GP* and *genotypic diversity* was unexpectedly weak and not significant in both teams. This result indicated that increase in diversity is not necessarily associated with an increase in the performance of the naïve algorithm for this specific RT problem. The reason for this is not known but may be related to the small population size (15).

Table 7.7: Correlation between variables involved in the CEAFS for the blue team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	**.264***	1			
Avg_GP	**-.537****	-.062	1		
Best_GP	-.124	-.128	**.749****	1	
Mutation	**. 767****	.207	**-.422****	-.047	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

Table 7.8: Correlation between variables involved in the CEAFS for the red team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	.216	1			
Avg_GP	-.172	-.113	1		
Best_GP	-.017	.027	**.910****	1	
Mutation	**.504****	.180	-.085	-.012	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

The correlation analysis was also conducted to examine the relationship between the same variables in the CEAFS for the blue and red team, which are presented in Table 7.7 and Table 7.8 respectively. Similar to the CEAN, the correlation between the *genotypic diversity* and *mutation rate* was significant in the CEAFS for both teams. Their correlation was also positive and strong which indicated that an increase in *mutation rate* increases *genotypic diversity*. The correlation between the two diversity measures was positive but weak. This indicated that changes in genomes only slightly change the spread of individuals‟ fitness in the population evolved by the CEAFS. As in the CEAN, a positive and strong correlation between the *estimated best* and *average GP* is also shown for CEAFS, indicating that the increase in the performance of overall population was associated with an increase the performance of the top individual in the evolved population. However, the correlation between *genotypic diversity* and the *estimated best GP* here was weak and negative in both teams. This weak correlation indicates that an increase in diversity may not ensure an increase in the *best GP*.

Table 7.9: Correlation between variables involved in the CEAHOF for the blue team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	**.431****	1			
Avg_GP	-.184	.125	1		
Best_GP	-.007	**.318***	**.945****	1	
Mutation	**.905****	**.418****	-.091	.074	1
**. Correlation is significant at the 0.01 level (2-tailed). *. Correlation is significant at the 0.05 level (2-tailed). N= 60					

Table 7.10: Correlation between variables involved in the CEAHOF for the red team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	.057	1			
Avg_GP	**-.341****	.152	1		
Best_GP	-.082	-.129	**.934****	1	
Mutation	**.883****	.077	**-.266***	-.028	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

The correlations between variables involved in the analysis of CEAHOF are presented in Table 7.9 and Table 7.10 for the blue and red teams respectively. Similar to the CEAN and CEAFS, *genotypic diversity* was significantly related to the *mutation rate* in both teams and also the correlation between them was positive and strong. This implied that even in the CEAHOF, an increase in *mutation rate* is correlated with increases in *genotypic diversity*. The correlation between *genotypic* and *phenotypic diversity* was also positive; however, in the red team the correlation was weaker. There was a positive and strong correlation between the *estimated best GP* and estimated *average GP*. As in the previously mentioned two algorithms, the correlation between *genotypic diversity* and the *estimated best GP* was also weak and negative.

Table 7.11: Correlation between variables involved in the CEACFH for the blue team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	**.351****	1			
Avg_GP	**-.281***	.071	1		
Best_GP	-.171	.147	**.885****	1	
Mutation	**.806****	**.510****	-.094	.051	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

Table 7.12: Correlation between variables involved in the CEACFH for the red team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	.072	1			
Avg_GP	-.073	.001	1		
Best_GP	-.130	**.057***	**.887****	1	
Mutation	**.452****	.137	.058	.138	1
**. Correlation is significant at the 0.01 level (2-tailed). *. Correlation is significant at the 0.05 level (2-tailed). N= 60					

Table 7.11 and Table 7.12 show the relationship between variables involved in the CEACFH for the blue and red teams respectively. As in the other three algorithms, g*enotypic diversity* was strongly and positive correlated with *mutation rates* in both teams. The correlation between *genotypic* and *phenotypic diversity* was positive but weak. The *estimated best GP* is negatively correlated with *genotypic diversity* in both teams.

The correlation analysis in each of the four algorithm employed in this study shows that *genotypic diversity* was strongly correlated with *mutation rate*. However, *phenotypic diversity* and *mutation rate* are not strongly correlated in all four algorithms. The correlation between the *estimated average GP* and *estimated best GP* were also strongly positive.

The relationship between the variables involved is visualized via scatter plots in the following section.

7.3.4.1 Relationship between Genotypic Diversity and Estimated Best GP

In order to visualize the relationship between the *estimated best GP* and *genotypic diversity*, Figure 7.12 (a) and (b) presented scatter plots for the blue and red teams‟ population respectively.

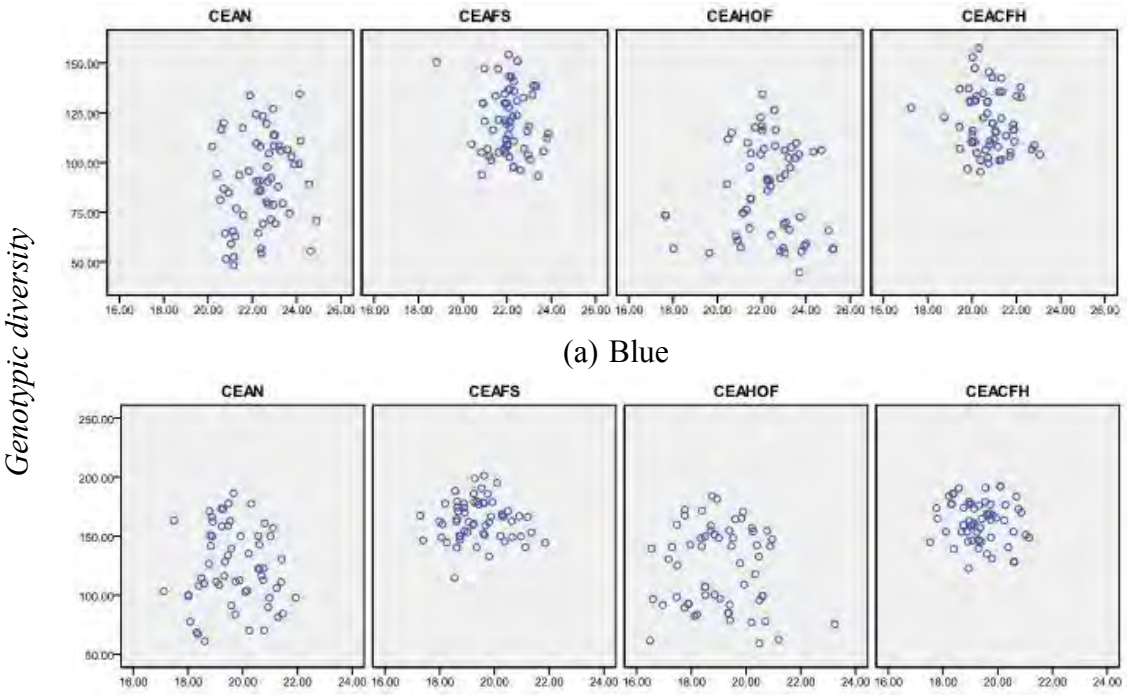

Figure 7.12: Scatter plots of genotypic diversity versus the estimated best GP in the CEAN, CEAFS, CEAHOF and CEACFH for (a) blue and (b) red team. Each point is a mean of the last 10 generations over 15 runs in 4 varieties of mutation rate which made 60 points in each plot.

Each plot consists of four parts representing four algorithms: the CEAN, CEAFS, CEAHOF and CEACFH. In both teams points associated with these two algorithms were less spread than in CEAN and CEACFH.

7.3.4.2 Relationship between Genotypic Diversity and Estimated Average GP

Figure 7.13 (a) and (b) shows the relationship between the *estimated average GP* and *genotypic diversity* for the blue and red team respectively. Although *genotypic diversity* was higher in the CEAFS and CEACFH, the average GP was lower as the points did not reach at the right side of plot. In case of the CEAN and CEAHOF, although the diversity was not very high, the average GP was higher. This result was also shown by the other statistical test, correlation, in section 7.3.4.

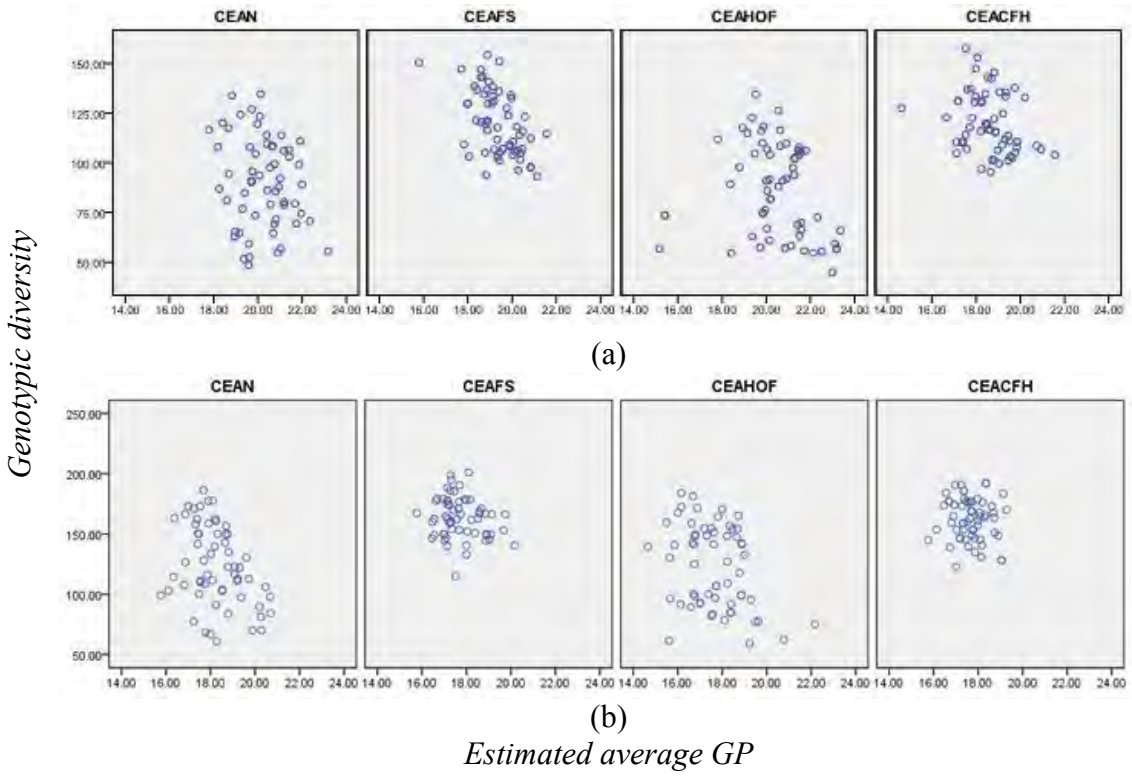

Figure 7.13: Scatter plots of genotypic diversity versus the estimated average GP in the CEAN, CEAFS, CEAHOF and CEACFH for (a) blue and (b) red team. Each point is a mean of the last 10 generations over 15 runs in 4 varieties of mutation rate which made 60 points in each plot.

The above section demonstrated the relationship between variables involved in this study. The plots show similar results to those shown in the intransitive number problem but the clusters are not so tight and clear in this study. The reason for this greater statistical variation may be due to the noise caused by the simulation run. Due to the computational time, only 15 simulation runs were executed to evaluate the individuals.

In order to visualize the evolved parameters as strategies, scenarios were created with the evolved parameters from each of the four algorithms. The following section provides the discussion of the evolved tactics.

7.3.5 Evolved Strategies for the Anchorage Protection Scenario

In optimizing the anchorage protection scenario, four CEA variants were used and their performance was evaluated. Each algorithm was tested at *mutation rate* of 10%, 20%, 40% and 60%. Each algorithm at a specific *mutation rate* produces two optimized

populations, one each for the blue and red. As the population size for this study was set to 15, each optimized population suggests 15 sets of parameters, each of which represents a strategy. Due to the high number of evolved strategies, it is not possible to discuss them all here. Therefore, a set of parameters (corresponding to a single strategy) with a higher fitness value was selected for examination in more detail for each team. The best evolved sets of parameters at a *mutation rate* of 40% from each of the four algorithms are discussed below:

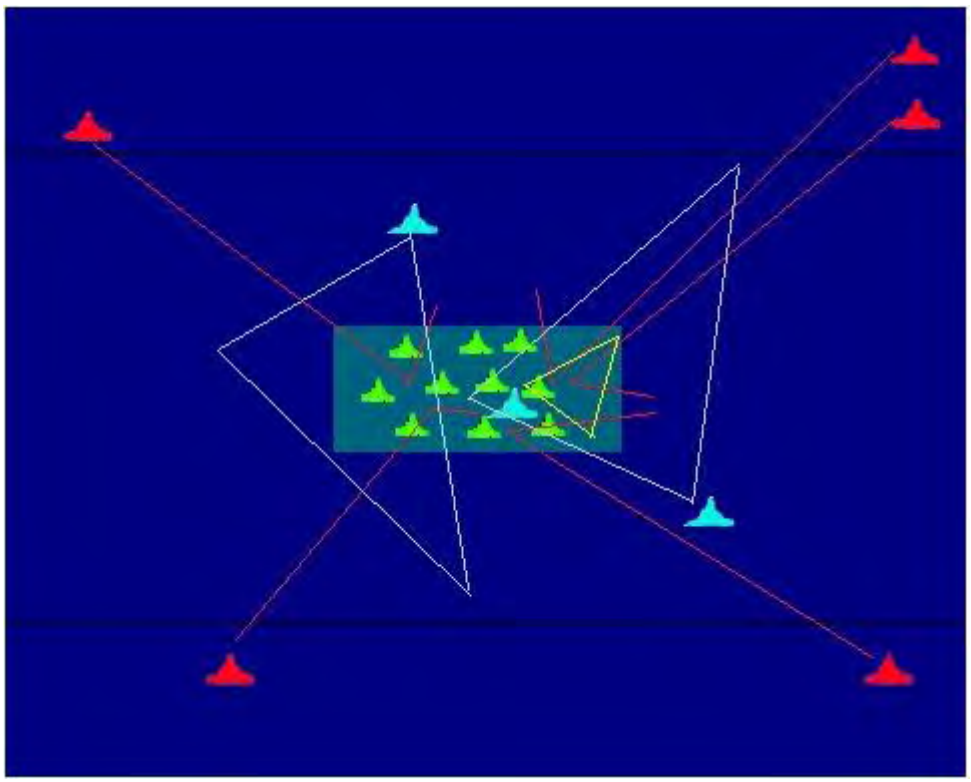

Figure 7.14: The red and blue evolved tactics when the scenario was optimized using the CEAN algorithms at 40% mutation rate

The best CEAN evolved strategies of the blue and red team for the anchorage protection scenario are depicted in Figure 7.14. The red boats used penetration tactics (direct attack) in which the team attacked the anchorage directly to maximize the opponent‟s casualties without regard to their own casualties. The tactics followed by the red team seemed appropriate as their purpose was to maximize the destruction of the neutral ships. The blue patrolling strategies were distributed into two types, outer (shown as blue triangles) and inner surveillance (shown as yellow triangle). Two blue boats tried

to stop the red boats from the outside and one blue boat tried to destroy the red that entered in the anchorage area.

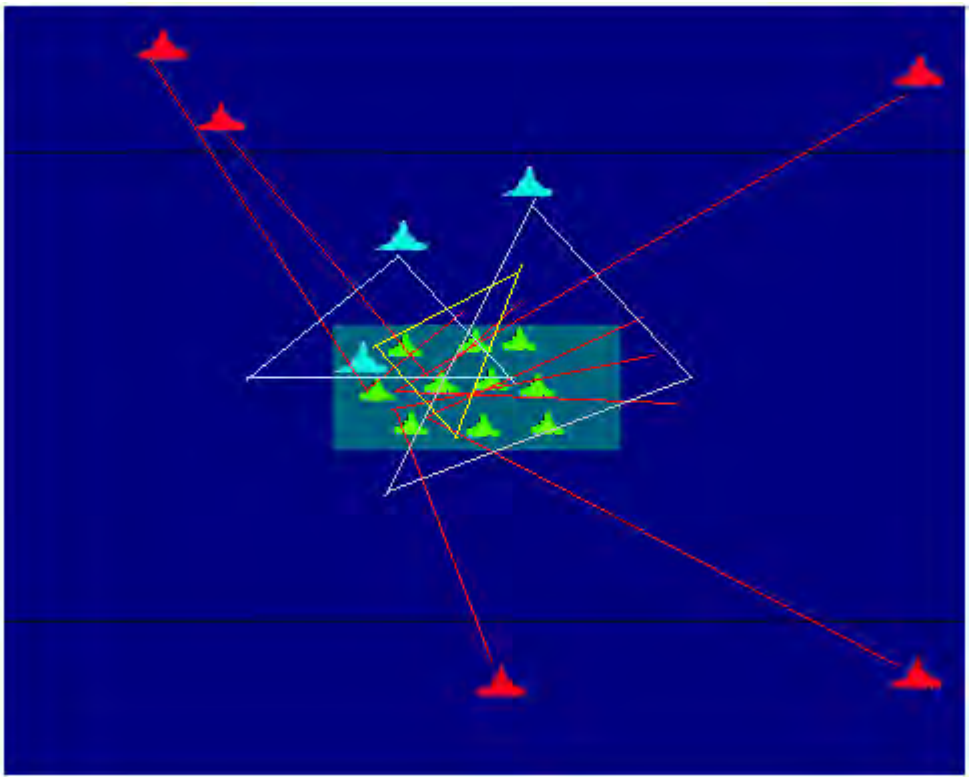

Figure 7.15: The red and blue tactics when the scenario was optimized using the CEAFS algorithm at 40% mutation rate

The best CEAFS evolved strategies for the blue and red teams are depicted in Figure 7.15. The red boats targeted the anchorage area from different locations to destroy the neutral green shipping and ran away from the place attacked to protect themselves from the blue boats. The blue boats focused surveillance more on the top side from where three red boats try to penetrate. The blue strategy may be effective as one blue boat can reach at the bottom part of the anchorage by the time when two red boats try to enter.

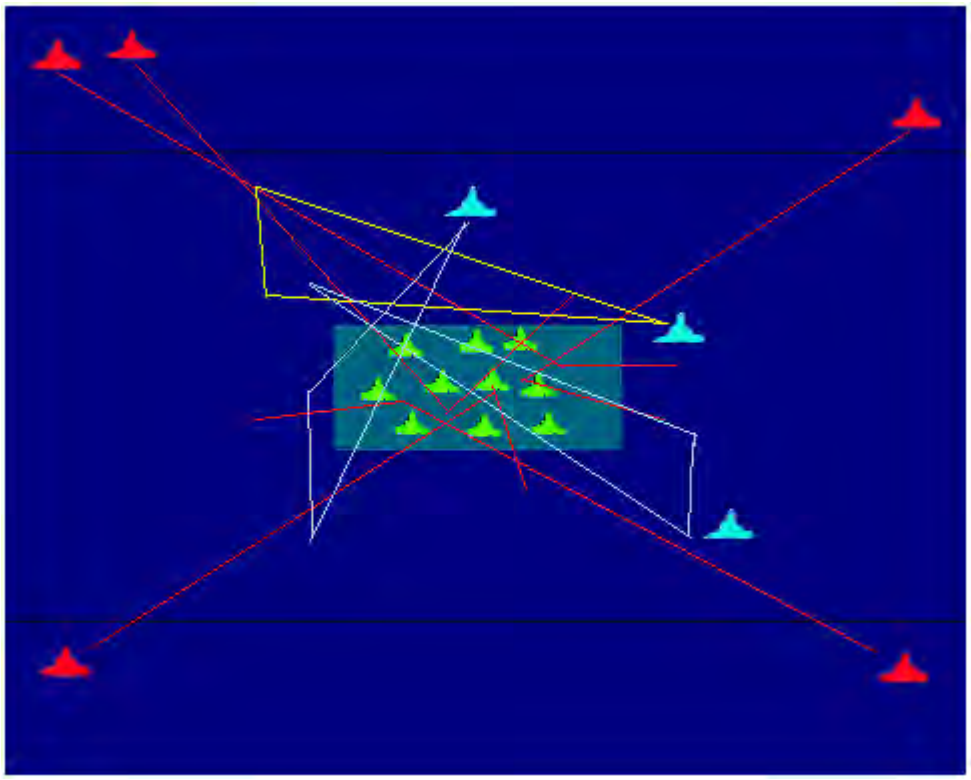

Figure 7.16: The red and blue emerged tactics when the scenario was optimized using the CEAHOF at 40% mutation rate

Figure 7.16 depicts the red team and blue teams‟ strategy that were evolved when the scenario was optimized using the CEAHOF algorithm, again at 40% *mutation rate*. Similar to the previous two scenarios, the red team incorporated a direct attack plan from all four corners. Since there were only three blue boats, the red boat strategy could maximize the blue attrition which was one of the aims of the red boats. The blue surveillance focussed on outer patrol and gave priority on the top left side of the anchorage from where two red boats can simultaneously attack. The red boats‟ strategy is especially challenging as they were distributed in different starting locations. However, the blue strategy was also effective as they did not focus on any specific area but widened their surveillance area, so that which they could detect penetration action in every corner of the anchorage.

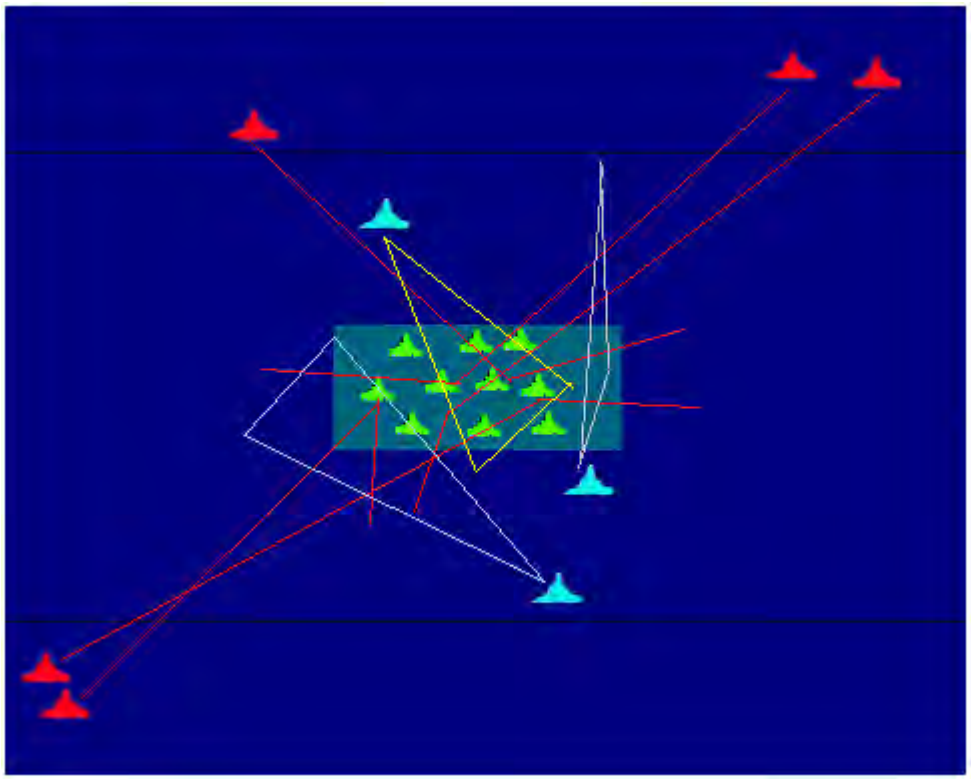

Figure 7.17: The red and blue emerged tactics when the scenario was optimized using the CEACFH at 40% mutation rate

The evolved blue and red strategies from CEACFH at 40% *mutation rate* are depicted in Figure 7.17. Similar to the tactics associated with the three previous scenarios, in this scenario the red team also followed a penetration strategy in which they directly attacked the anchorage from three corners to maximize the blue casualties and destroy the green boats. The blue team also focused their surveillance on each of the three corners from where there were possibilities of a red boat entering the anchorage. The blue strategy may be effective as they discovered the location from where the red boats could enter in the anchorage. The red strategy on the other hand could maximize the blue casualties as the green boats are within firing range when they encounter the blue boats.

From the scenario analysis, it was found that the red boats follow a direct attack strategy using different starting locations. The blue team focus more on outer surveillance. In each of the scenarios analysed above, out of three blue boats two of them focus on outer patrol whereas one blue boat gives priority to inner patrol.

7.4 Results and Analysis for the Coastline Protection Scenario

Similar to section 7.3, CEAN, CEAFS, CEAHOF and CEACFH were employed in optimizing another RT scenario, the coastline protection scenario. The performance of these algorithms in terms of their generalisation ability and also their capabilities for producing multiple optimal solutions in this specific scenario are presented in this section. For measuring generalisation ability, GP was used and the existence of the multiple optimal solutions was analysed by the *peak detection technique* detailed in 4.1.2.1. In order to see the influence of diversity on the performance of the algorithms, diversity was also measured.

7.4.1 Analysis of the GPs for the Blue and Red Team

This section presents an analysis of the *estimated best GP* associated with the four algorithms from each of the two teams, the blue and red team. Each of the four algorithms was executed 15 times using the parameters shown in Table 7.3 for each of the teams involved. The *estimated best GP* was measured at *mutation rate* of 10%, 20%, 40% and 60%. The average of these 15 runs for each of the corresponding 50 generations is presented using convergence plots, depicted in Figure 7.18. The x-axis and y-axis represent the number of generations and the *estimated best GP* respectively.

When a *mutation rate* of 10% (Figure 7.18 (a) and (b)) was employed, in the blue team, CEAN and CEAHOF performed slightly better than CEAFS and CEACFH in terms of *estimated best GP*. In the case of the red team, the performance of CEACFH appeared to be less competitive whereas the performance of other three algorithms was similar. Despite using the same parameters in terms of simulation runs and generations, the performance of the algorithms were more stable in final generations in this scenario in comparison to the anchorage protection scenario.

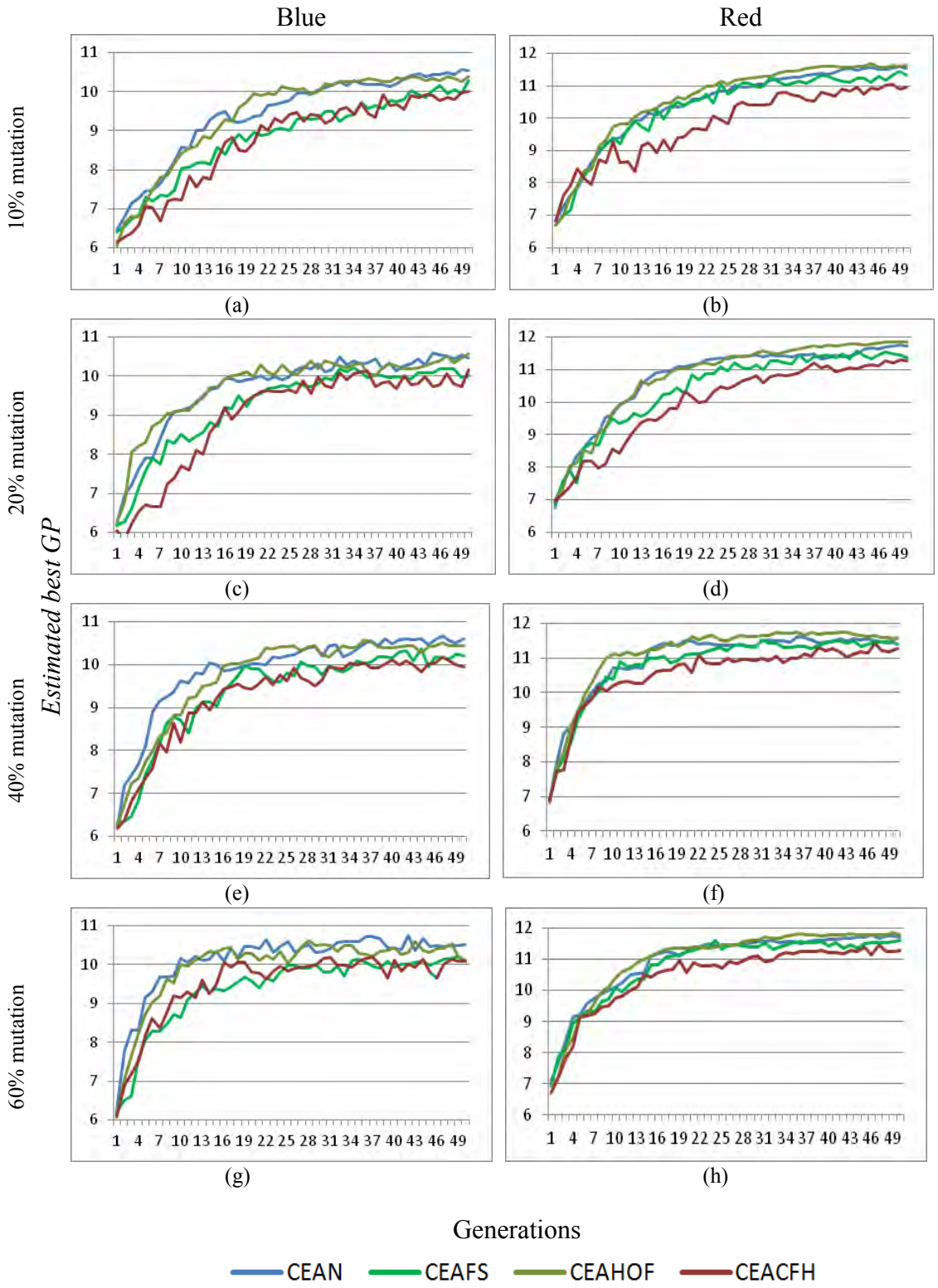

Figure 7.18: Convergence plots showing the estimated best GPs of the four algorithms for the blue (a, c, e, g) and red (b, d, f, h) team for mutation rates of: (a, b) 10%, (c, d) 20%, (e, f) 40%, (g, h) 60%

When 20% *mutation rate* was applied (Figure 7.18 (c) and (d)), similar result as in the 10% *mutation rate* was shown; however, it is noticeable that all four algorithms become stable by approximately the 40th generation. When a 40% *mutation rate* was applied (Figure 7.18 (e) and (f)), *estimated best GP* associated with all four algorithms were similar and fluctuating within a band width of 9 to 10.5 from approximately the 30th generation. When a *mutation rate* of 60% was applied, (Figure 7.18 (f) and (h)), all four algorithms were stable with some degree of fluctuations from approximately the 20th generation.

The coastline protection scenario is an asymmetric problem as two teams were evaluated using different formulas (Equations (7.3) and (7.4) for the blue and red team respectively). Therefore, it is expected to have different outcomes for each of the two teams. Figure 7.18 also showed that the trend of each algorithm in the blue and red teams was dissimilar. In order to further analyse the performance of the four algorithms, the *estimated best GP* and the *estimated average GP* are presented via interaction plots and ANOVA output.

7.4.1.1 Analysis of Estimated Best GP

Figure 7.19 shows the interaction plots of the *estimated best GP* associated with four algorithms versus *mutation rate.* The *estimated best GP* was evaluated by employing the *mutation rate* of 10%, 20%, 40% and 60%. As in section 7.3.1.1, each point is a mean value of last 10 generations over 15 runs. The x-axis is the *estimated best GP* value and the y-axis is *mutation rate*.

Figure 7.19 shows that *estimated best GP* associated with CEAN and CEAHOF was higher in comparison to CEAFS and CEACFH. In each of the four algorithms, the *estimated best GP* value was lowest at a *mutation rate* of 20% and highest at a *mutation rate* of 40%.

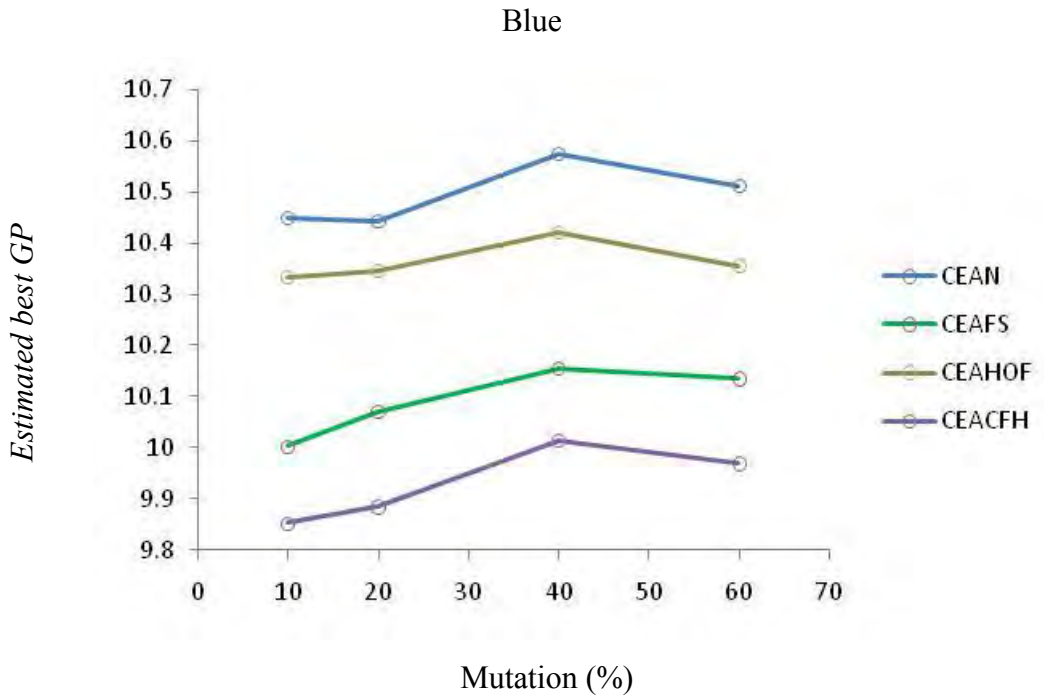

Figure 7.19: Interaction plots of the estimated best GP (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate for each of the 4 algorithm variants in the blue team

The *estimated best GP* was analysed by means of a two-way between-subjects ANOVA test with four levels for *mutation rate* and four levels for *algorithm used*. The overall model is statistically significant, ($F(15, 224) = 3.404$, $p < 0.05$, Partial Eta Squared = 0.186). The effect of the *algorithm used* was also statistically significant ($F(3, 224) = 16.064$, $p = <0.05$, Partial Eta Squared = 0.177). However, *mutation rate* ($F(3, 224) = 0.851$, $p = 0.467$, Partial Eta Squared = 0.011) and the interaction *(algorithms used* × *mutation rate)* were not statistically significant with ($F(9, 224) = 0.035$, $p = 1.00$, Partial Eta Squared = 0.001). The tables associated with ANOVA outputs are depicted in Appendix D.1

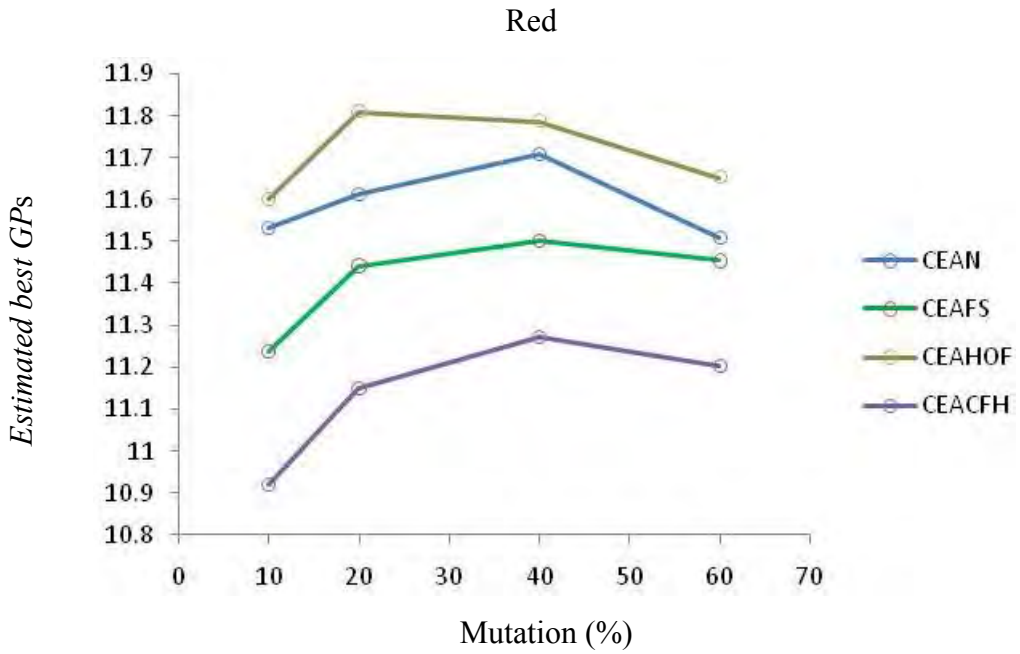

Figure 7.20: Interaction plots of the estimated best GP (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate for each of the 4 algorithm variants in the red team

Figure 7.20 shows an interaction plot of *estimated best GP* versus *mutation rate* from the red team. The *estimated best GP* associated with CEAN, CEAHOF and CEAFS were similar whereas CEACH was low. As in the blue team, *estimated best GP* of all four algorithms except CEAHOF was highest at a *mutation rate* of 40%. In CEAHOF, *estimated best GP* was slightly higher at a *mutation rate* of 20%. It appears that there is no interaction between *mutation rate* and *algorithm used* in this analysis.

ANOVA was conducted which shows that the overall model is statistically significant, ($F(15, 224) = 9.289$, $p < 0.05$, Partial Eta Squared = 0.383). The effect of the *algorithm used* ($F(3, 224) = 38.132$, $p < 0.05$, Partial Eta Squared = 0.338) and *mutation rate* ($F(3, 224) = 6.448$, $p < 0.05$, Partial Eta Squared = 0.079) was also statistically significant. However, the interaction *(algorithm used × mutation rate)* was not statistically significant with ($F(9, 224) = 0.622$, $p = 0.778$, Partial Eta Squared = 0.024). The table associated with ANOVA is depicted in Appendix D.2

As with the anchorage protection scenario studied in the previous section, the coastline protection scenario is also an asymmetric problem. The fitness for the blue and red team

was measured differently using Equations (7.3) and (7.4) respectively. Thus, the interaction plots for the *estimated best GP* show different scales for the two teams‟ data.

7.4.1.2 Analysis of Estimated Average GP

This section presents the *estimated average GP* associated with all four algorithms from the blue and red team. An interaction plot was created to visualize the interaction of *estimated average GP* and *mutation rate* from the blue team which are depicted in Figure 7.21. The *estimated average GP* associated with CEAN and CEAHOF shows that there was a negative impact of *mutation rates* on *estimated average GP*. However, *estimated average GP* associated with CEAFS and CEACFH was not highly influenced by varying *mutation rates.*

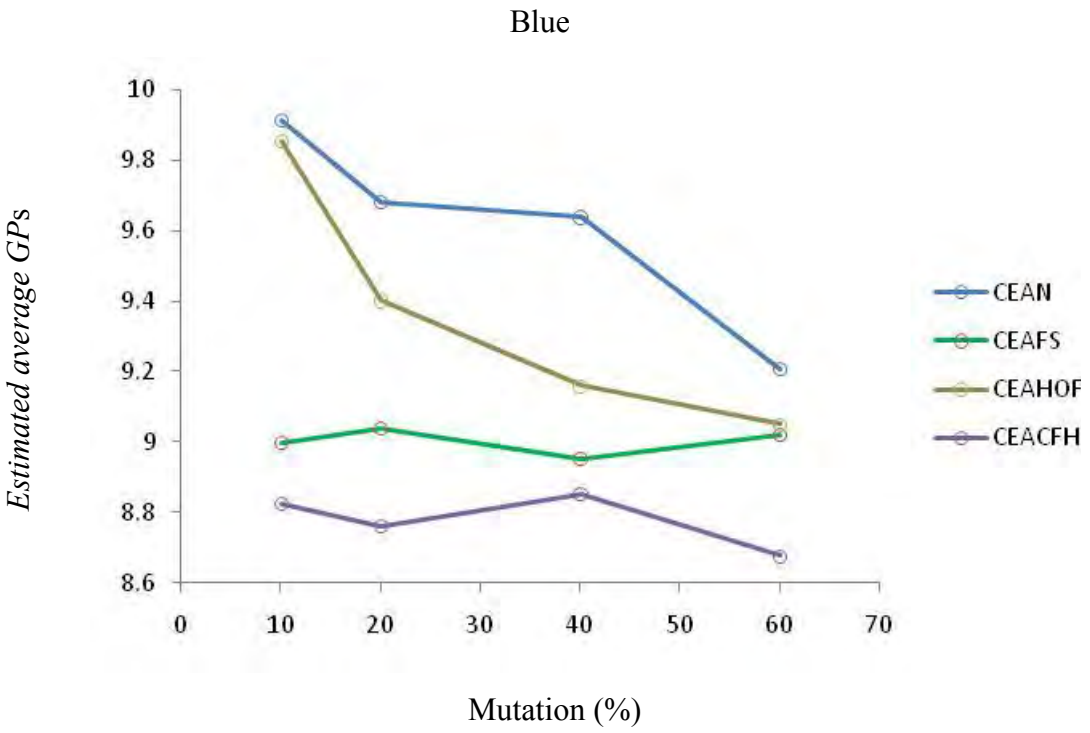

Figure 7.21: Interaction plots of the estimated average GP (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate, for each of the 4 algorithm variants in the blue team.

ANOVA was again conducted to further investigate *estimated average GP* and the effect of *mutation rate*, *algorithm used* and their interaction. The tables associated with ANOVA outputs are depicted in Appendix D.3. The test shows that the overall model is

statistically significant, (F (15, 224) = 5.489, p < 0.05, Partial Eta Squared = 0.269). The effect of the independent variables, *algorithms used and mutation rate* was also statistically significant. (F (3, 224) = 19.591, p < 0.05, Partial Eta Squared = 0.208) and (F (3, 224) = 4.074, p < 0.05, Partial Eta Squared = 0.052) respectively. However, their interaction (F (9, 224) = 1.261, p = 0.259, Partial Eta Squared = 0.048) was not statistically significant.

Figure 7.22 shows that, in the red team, *estimated average GP* associated with CEAN and CEAHOF were relatively higher than in CEAFS and CEACFH. As in the blue team, the performance associated with CEACFH was not competitive. Additionally, it appears that there is no interaction between the *estimated average GP* and *mutation rate*.

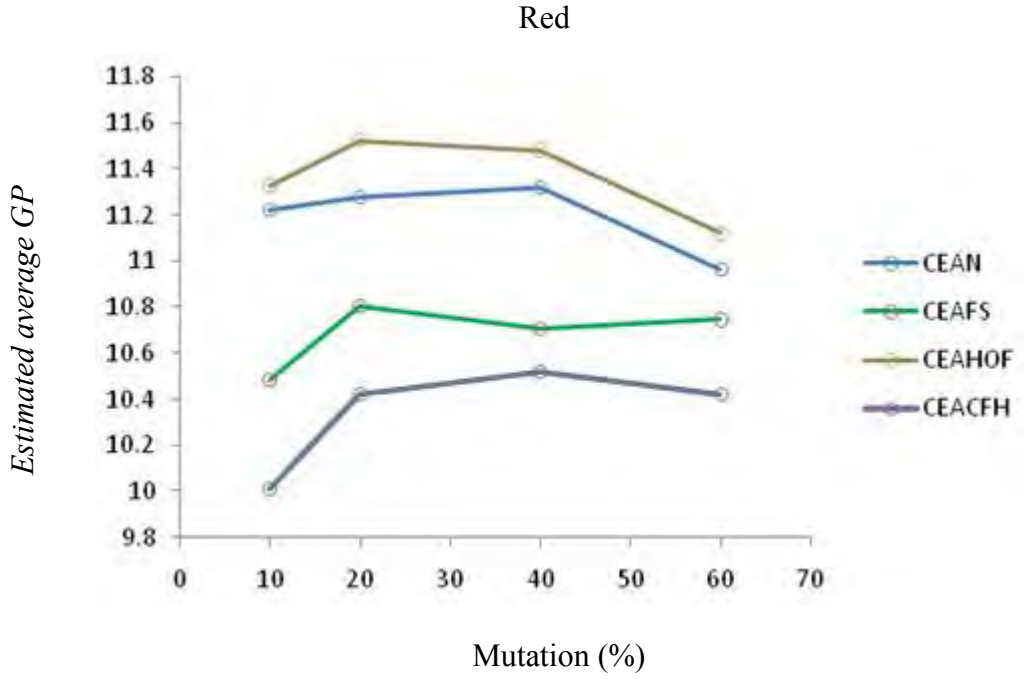

Figure 7.22: Interaction plots of the estimated average GP (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate, for each of the 4 algorithm variants in the red team.

The *estimated average GP* was analysed by means of ANOVA which shows that the overall model (F (15, 224) = 15.617, p < 0.05, Partial Eta Squared = 0.511), the effect of the *algorithm used* (F (3, 224) = 68.138, p = 0.00 (Partial Eta Squared = 0.477) and *mutation rate* (F (3, 224) = 5.093, p = 0.002, Partial Eta Squared = 0.064) were

statistically significant. However, the interaction *(algorithm used × mutation rate)* was not statistically significant with (F (9, 224) = 1.618, p = 0.111, Partial Eta Squared = 0.061). The table associated with ANOVA is depicted in Appendix D.4

The ANOVA analysis of *estimated best GP* and *estimated average GP,* in both the blue and red team, shows that there was no interaction effect on each of the GP analysed. The *algorithm used* was significant in all four models whereas *mutation rate* was also significant except in the *estimated best GP* of the blue team. As discussed in the anchorage protection scenario analysis, the expectation was that fitness sharing would provide better performance in terms of GP. However, the performance of CEAFS was not very different from the other three algorithms. The reason may be due to the fact that the fitness sharing was based on the genomes rather than behaviours.

As in the anchorage protection scenario, a *peak detection technique* was used to analyse whether each of the four algorithms locates multiple local optima. The results are described in the following sections.

7.4.2 Local Optima Test of the Evolved RT Strategies

Similar to the anchorage protection scenario as in section 7.3.2, an additional test was conducted that examines whether each of the algorithms are capable of locating multiple local optima. Since the population size was 15 for this empirical study, there were 15 evolved strategies for each team. For the local optima test, only the last generation population at a 40% *mutation rate* for each algorithm were analysed as in the anchorage protection scenario analysis. The evolved strategies from CEAN, CEAFS, CEAHOF and CEACFH for both the blue and red team are depicted in Appendix E.1, Appendix E.2, Appendix E.3 and Appendix E.4 respectively.

As in the anchorage protection scenario, 10 strategies were randomly generated within a radius of 5 (in this scenario, each gene is a real number ranging from 0 to 100). The small value was set to 0.2 and a value of 0.025 was used as an epsilon value. Table 7.13 shows the number of local optima detected in the final generation in all four algorithms.

Table 7.13: Local optima test result for the anchorage scenario

Algorithms	Local Optima
Blue	
CEAN	5
CEAFS	8
CEAHOF	7
CEACFH	5
Red	
CEAN	6
CEAFS	9
CEAHOF	8
CEACFH	5

Similar to the anchorage protection scenario result, Table 7.13 showed that CEAFS and CEAHOF were capable of locating more local optima than CEAN and CEACFH in both the blue and red teams.

The above analysis showed that although all four algorithms (except CEACFH in the blue team) performed similarly in terms of generalising ability, CEAFS and CEAHOF locate more multiple optima in comparison to CEAN and CEACFH. The peak finding performance of CEAFS was expected, as higher diversity encourages a more complete exploration of the search space, but the good performance of HOF was again unexpected. An explanation for this result is a possible subject for future research.

As in the anchorage protection scenario, experiments were conducted to investigate diversity of populations produced by the algorithms when 10%, 20%, 40% and 60% *mutation rate* was applied. The following section presents details of this experiment.

7.4.3 Analysing Diversity

This section presents the analysis of the four algorithms‟ capability in producing diverse population. For this, diversity of the population was measured in two ways: *genotypic* and *phenotypic diversity*. The *mutation rate*s of 10%, 20, 40% and 60% were used to examine the diversity of the population. Interaction plots and ANOVA output are presented to evaluate diversity of the population associated with these four algorithms. The results are explained on the basis of the effect on a *genotypic* and *phenotypic diversity* measures.

When analysing *genotypic diversity* of the blue and red team, similar results to those described in chapter 5 (an intransitive problem) and chapter 6 (a multimodal problem) were displayed in this scenario. CEAFS and CEACFH evolved populations with higher diversity in comparison to the populations evolved by CEAN and CEAHOF. Additionally, an increase in *mutation rate* increases the diversity of populations associated with all four algorithms. However, the effect of fitness sharing and *mutation rate* was small when *phenotypic diversity* was measured. All four algorithms were equally diverse in terms of *phenotypic diversity* in both blue and red team but the effect of *mutation rate* is higher in the blue team. Interaction plots of *genotypic diversity* from the blue and red team are depicted in Figure 7.23 and Figure 7.24 respectively. Likewise, interaction plots of *phenotypic diversity* were depicted in Figure 7.25 and Figure 7.26 for the blue and red team respectively. The tables associated with ANOVA for genotypic diversity is depicted in Appendix D.5 and Appendix D.6 for the blue and red team respectively. Additionally, the ANOVA outputs for the phenotypic diversity of the blue and red team were depicted in Appendix D.7 and Appendix D.8 respectively.

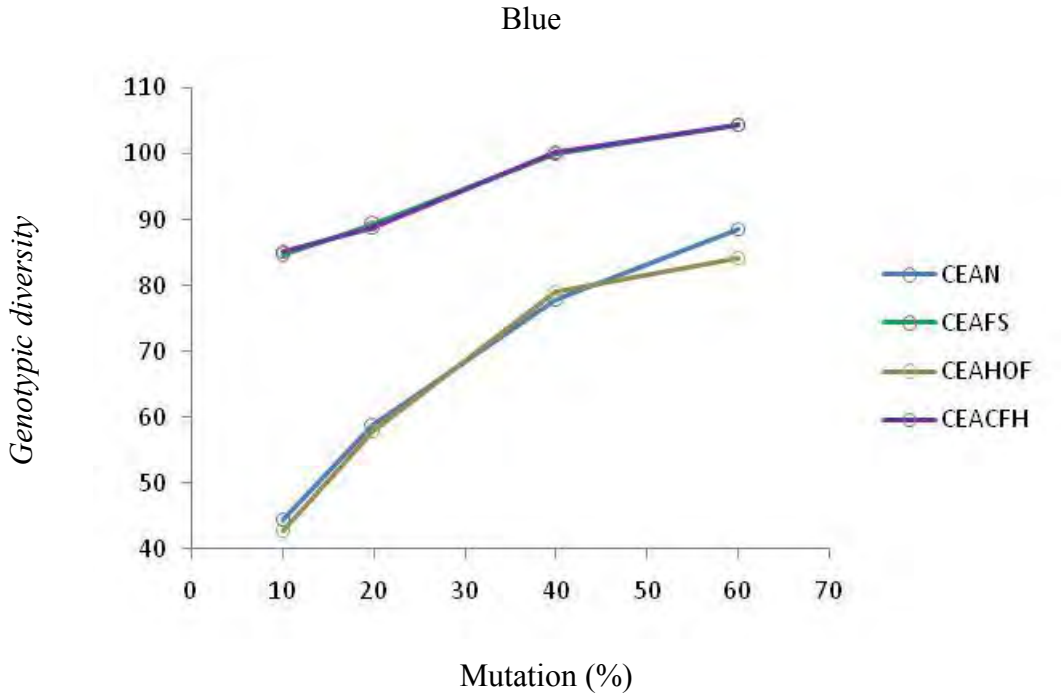

Figure 7.23: Interaction plots of genotypic diversity (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate, for each of the 4 algorithm variants in the blue team.

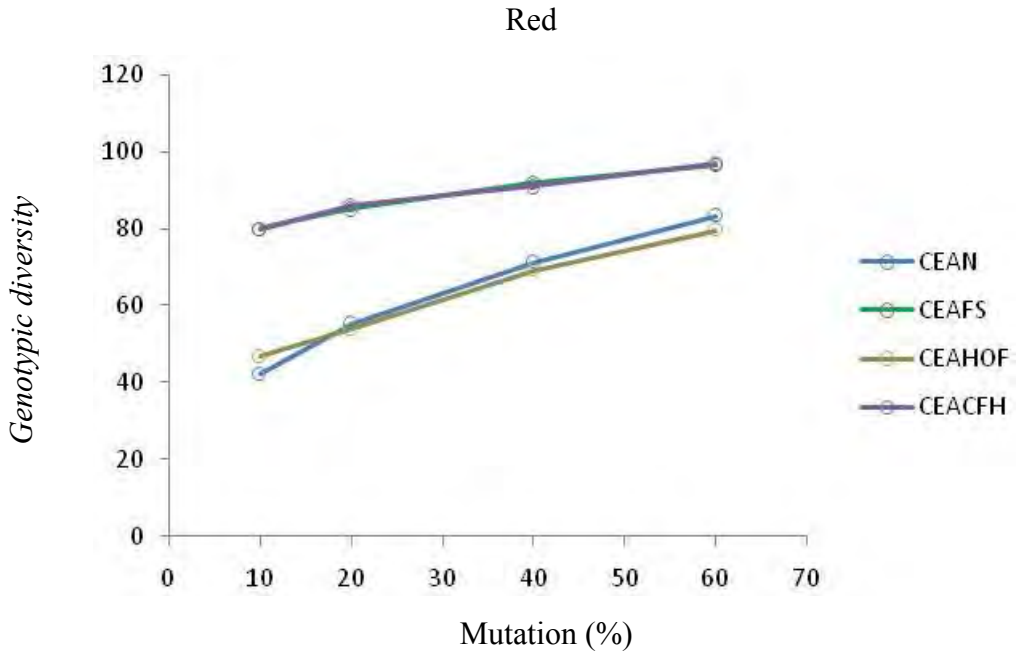

Figure 7.24: Interaction plots of genotypic diversity (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate, for each of the 4 algorithm variants in the red team.

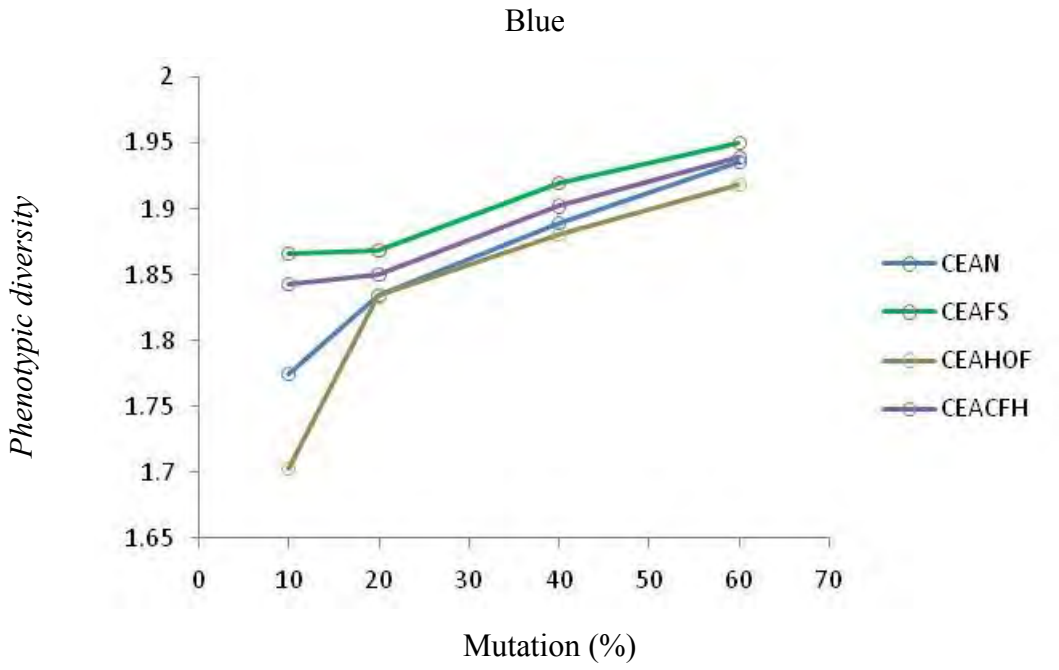

Figure 7.25: Interaction plots of phenotypic diversity (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate, for each of the 4 algorithm variants in the blue team.

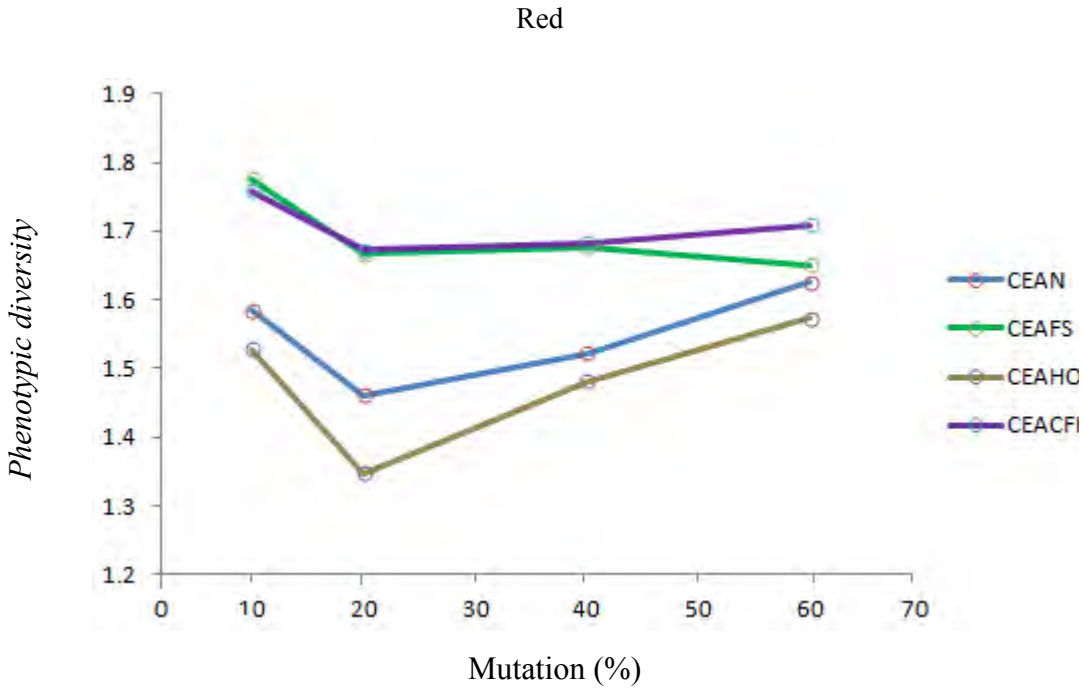

Figure 7.26: Interaction plots of phenotypic diversity (mean over the final 10 generations out of 50 generations in 15 runs) versus mutation rate, for each of the 4 algorithm variants in the red team.

As in the anchorage protection scenario, the two diversity measures were inconsistent as *genotypic diversity* was highly influenced by the *mutation rate.* The effect of *mutation rate* on *phenotypic diversity* was different for each of the two teams. In the blue team the effect was less whereas in the red team, *phenotypic diversity* was not highly influence by the *mutation rate* in CEAFS and CEACFH.

7.4.4 Relationship between Diversity and GPs

In order to examine the relationship between diversity and GP, correlation analysis was conducted. The relationships between variables involved in all four algorithms; CEAN, CEAFS, CEAHOF and CEACFH are presented individually.

Table 7.14: Correlation between variables in the CEAN algorithm of the blue team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	**.505****	1			
Avg_GP	**-.423****	**-.518****	1		
Best_GP	-.012	-.174	**.829****	1	
Mutation	**.885****	**.484****	**-.355****	.066	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

Table 7.15: Correlation between variables in the CEAN algorithm of the red team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	.196	1			
Avg_GP	-.100	**-.552****	1		
Best_GP	.047	**-.514****	**.963****	1	
Mutation	**.911****	.139	-.153	-.009	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

The variables in CEAN are shown in Table 7.14 and Table 7.15 for the blue and red team respectively. There was a significant positive correlation between the *mutation rate* and *genotypic diversity* which also support the result depicted in the interaction plots (Figure 7.23 and Figure 7.24) and ANOVA outcomes. Likewise, the *estimated best GP* and *estimated average GP* were positively correlated which was expected as an increase in the performance of overall population also increases the performance of the top individual. As expected, there was a positive correlation between *genotypic* and *phenotypic diversity* in both teams. Interestingly, the correlation between *phenotypic diversity* and *estimated average GP* was negative and strong.

Table 7.16: Correlation between variables in the CEAFS algorithm of the blue team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	.154	1			
Avg_GP	.080	**-.340****	1		
Best_GP	.130	-.127	**.871****	1	
Mutation	.**621****	**.461****	-.006	.120	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

Table 7.17: Correlation between variables in the CEAFS algorithm of the red team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	.017	1			
Avg_GP	-.110	**-.508****	1		
Best_GP	.044	**-.424****	**.873****	1	
Mutation	**.602****	**-.344****	.173	**.291***	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

The relationship between the variables involved in the CEAFS is shown in Table 7.16 and Table 7.17 which represents the blue and red tem respectively. As shown in the CEAN, the correlation between *genotypic diversity* and *mutation rate* was strong and positive in both teams. Likewise, the correlation between *genotypic* and *phenotypic diversity* was not significant in both the blue and red team. The correlation between *mutation rate* and *estimated best GP* was not statistically significant in the blue teams but the correlation was weak in the red team which support the result depicted in the ANOVA outcome.

Table 7.18: Correlation between variables in the CEAHOF algorithm of the blue team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	**.581****	1			
Avg_GP	**-.335****	**-.520****	1		
Best_GP	.042	-.199	**.761****	1	
Mutation	**.867****	**.637****	**-.339****	.031	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

Table 7.19: Correlation between variables in the CEAHOF algorithm of the red team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	**-.336****	1			
Avg_GP	**-.406****	**-.653****	1		
Best_GP	-.169	**-.561****	**.923****	1	
Mutation	**.858****	-.223	-.219	.010	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

The correlation between variables in the CEAHOF is shown in Table 7.18 and Table 7.19 for the blue and red team respectively. As in the analysis of CEAN and CEAFS, the correlation between *genotypic diversity* and *mutation rate* was strongly positive in both teams. The *phenotypic diversity* was significant with the *mutation rate* in the blue team but their correlation was not significant in the red team. Interestingly, *genotypic diversity* was not significant with the *estimated best GP* in both the blue and red team. This variation between two teams also reflects the nature of the asymmetric behaviour of RT application.

Table 7.20: Correlation between variables in the CEACFH algorithm of the blue team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	**.402****	1			
Avg_GP	-.140	**-.472****	1		
Best_GP	-.102	**-.261***	**.866****	1	
Mutation	**.571****	**.489****	-.070	.107	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

Table 7.21: Correlation between variables in the CEACFH algorithm of the red team

	Genotypic	Phenotypic	Avg_GP	Best_GP	Mutation
Genotypic	1				
Phenotypic	-.052	1			
Avg_GP	.231	-**.514****	1		
Best_GP	**.413****	**-.314***	**.852****	1	
Mutation	**.581****	-.091	**.328***	**.339****	1
**. Correlation is significant at the 0.01 level (2-tailed).					
*. Correlation is significant at the 0.05 level (2-tailed).					
N= 60					

Table 7.20 and Table 7.21 depict correlation tables from the blue and red team respectively for CEACFH. As in the other three algorithms, the correlation between *genotypic diversity* and *mutation rate* was positive in both teams. Interestingly in the blue team, the correlation between the *estimated best GP* and both diversity measures was not significant; however, in the red team *genotypic diversity* was positively correlated but *phenotypic diversity* was negatively correlated with the *estimated best GP*.

The above tables presented the correlation between variables involved in both the blue and red teams. In all four algorithms, as expected, *genotypic diversity* was strongly correlated with *mutation rate*. Likewise, *estimated average GP* and *estimated best GP* were also positively and strongly correlated. In order to visualize their relationship, the following section presents the scatter plots.

7.4.4.1 Relationship between Genotypic Diversity and Estimated Best GP

In order to visualize the relationships between variables involved in this study, the scatter plots are presented in Figure 7.27 (a) and (b), showing the relationship between *genotypic diversity* and *estimated best GP* in the blue and red team respectively. The plots show that in both teams, the CEAFS and CEACFH received higher diversity but low *estimated best GP*. Contrary, the CEAN and CEAHOF achieve low diversity but higher *estimated best GP*.

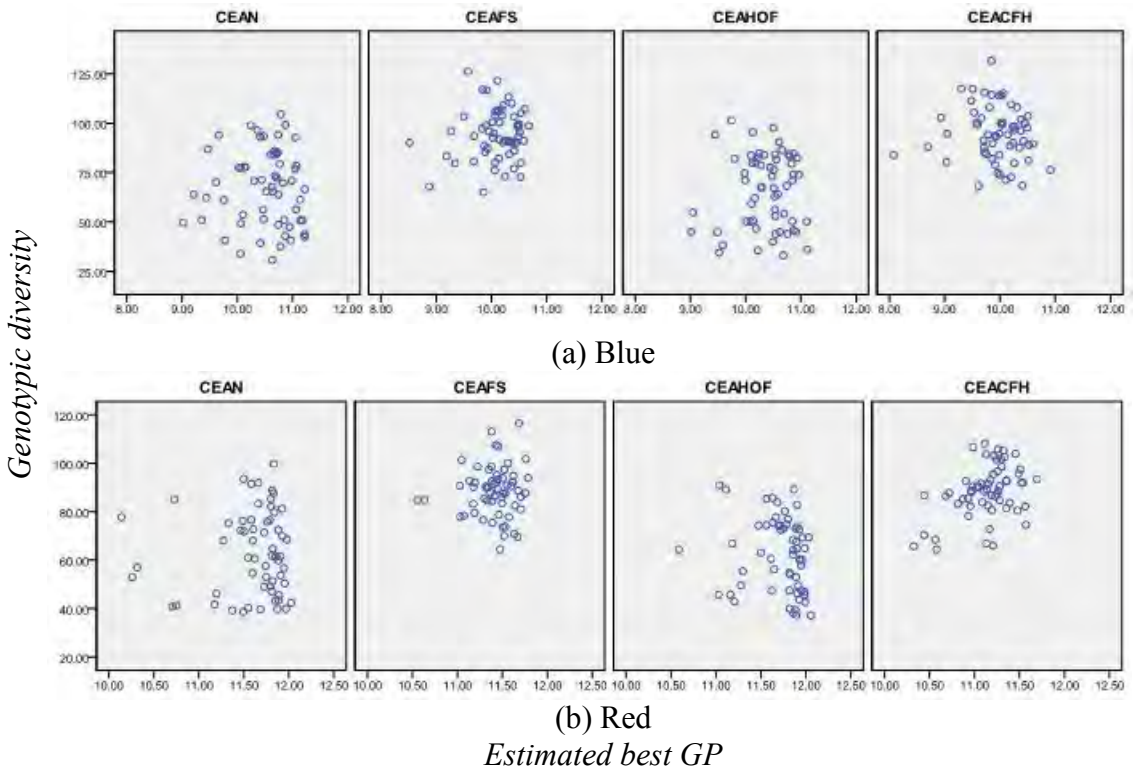

Figure 7.27: Scatter plots of genotypic diversity versus the estimated best GP in the CEAN, CEAFS, CEAHOF and CEACFH for (a) blue and (b) red team. Each point is a mean of the last 10 generations over 15 runs in 4 varieties of mutation rate which made 60 points in each plot.

7.4.4.2 Relationship between Phenotypic Diversity and Estimated Best GP

The relationship between the *phenotypic diversity* and *estimated best GP* for the blue and red teams are depicted in Figure 7.28 (a) and (b) respectively. In both teams, there was negative relationship between these two variables in CEAN and CEAHOF whereas the relationship between these two variables was unclear in CEAFS and CEACFH which support the result obtained from the correlation analysis.

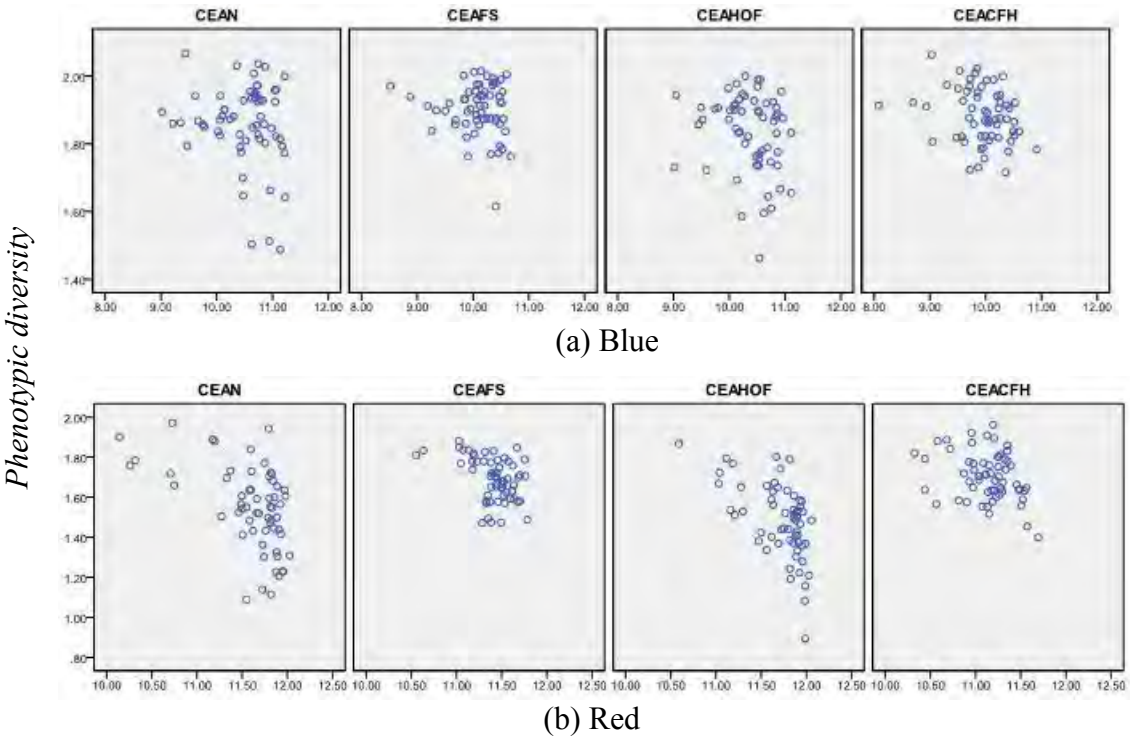

Figure 7.28: Scatter plots of phenotypic diversity versus the estimated best GP in the CEAN, CEAFS, CEAHOF and CEACFH for (a) blue and (b) red team. Each point is a mean of the last 10 generations over 15 runs in 4 varieties of mutation rate which made 60 points in each plot.

7.4.4.3 Relationship between Genotypic Diversity and Estimated Average GP

Similar to Figure 7.28, the relationship between the *genotypic diversity* and *estimated average GP* is depicted in Figure 7.29 (a) and (b) for the blue and red team respectively. The figures also demonstrate weak relationship between these two variables. As in the relationship between the *estimated best GP* and *genotypic diversity*, the CEAN and CEAHOF achieve higher best GP and low *genotypic diversity*. Contrary, the CEAFS and CEACFH achieve higher diversity but low *estimated best GP*.

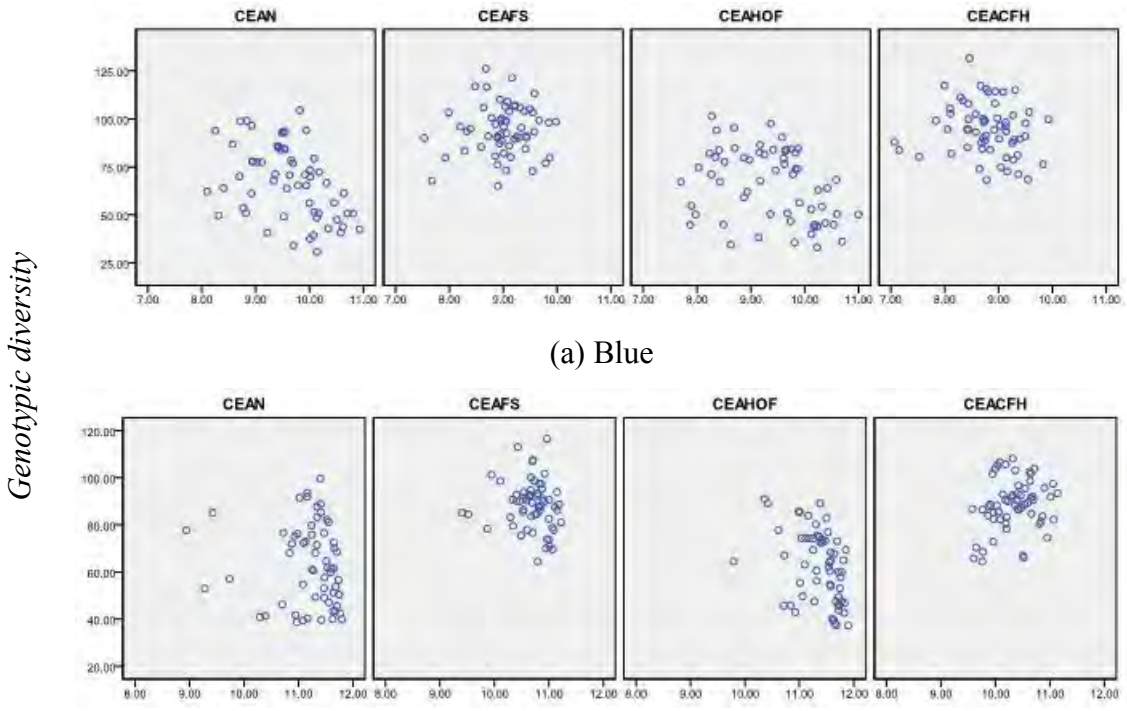

Figure 7.29: Scatter plots of genotypic diversity versus the estimated average GP in the CEAN, CEAFS, CEAHOF and CEACFH for (a) blue and (b) red team. Each point is a mean of the last 10 generations over 15 runs in 4 varieties of mutation rate which made 60 points in each plot.

7.4.4.4 Relationship between Phenotypic Diversity and Estimated Average GP

Similar to Figure 7.28, the relationship between *genotypic diversity* and *estimated average GP*s for the blue and red team are depicted in Figure 7.29 (a) and (b) respectively. The figures also demonstrate weak relationship between these two variables.

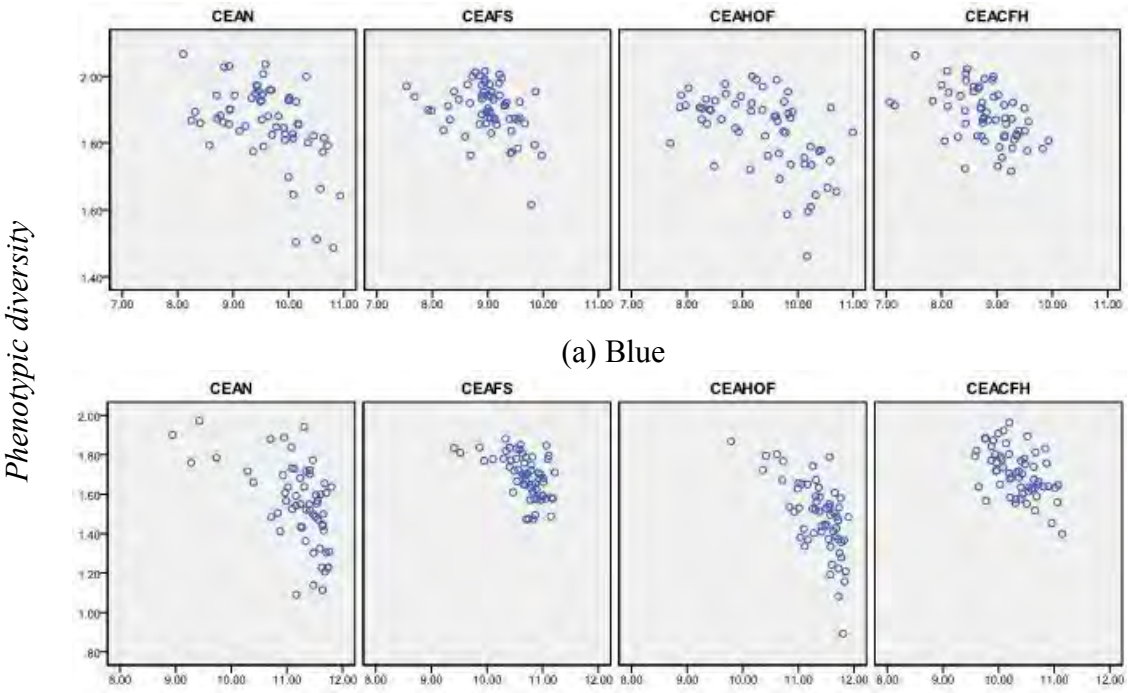

Figure 7.30: Scatter plots of phenotypic diversity versus the estimated average GP in the CEAN, CEAFS, CEAHOF and CEACFH for (a) blue and (b) red team. Each point is a mean of the last 10 generations over 15 runs in 4 varieties of mutation rate which made 60 points in each plot.

The above section described information about all four algorithms indicating the relationship between diversity and GP in this study. Scatter plots displayed that there were negative relationship between *phenotypic diversity* and *GP* (both the *estimated best* and *estimated average*). However, the relationship between *genotypic diversity* and *GP* (both) was not so strong.

In order to visualize the outcomes as strategies obtained from the four algorithms, the following section provides the discussion of the evolved tactics.

7.4.5 Evolved Strategies for the Coastline Protection Scenario

Similar to the anchorage protection optimization, the coastline protection was also optimized using four variants of CEAs with the population size 15. Thus, there were 15 sets of parameters in each team which evolved in each generation in each of the four algorithms. Each set of these parameters represents the blue or red team‟s strategy. Due

to the high number of strategies, it is not possible to discuss all strategies here. Therefore, from the last generation, a set of parameters with the highest fitness value from each team were put into the scenario and it was executed in the simulator to investigate the evolved strategy. From each of the four algorithms, the evolved set of parameters at a *mutation rate* of 40% from both teams was chosen. The evolved tactics for the blue and red team are presented below.

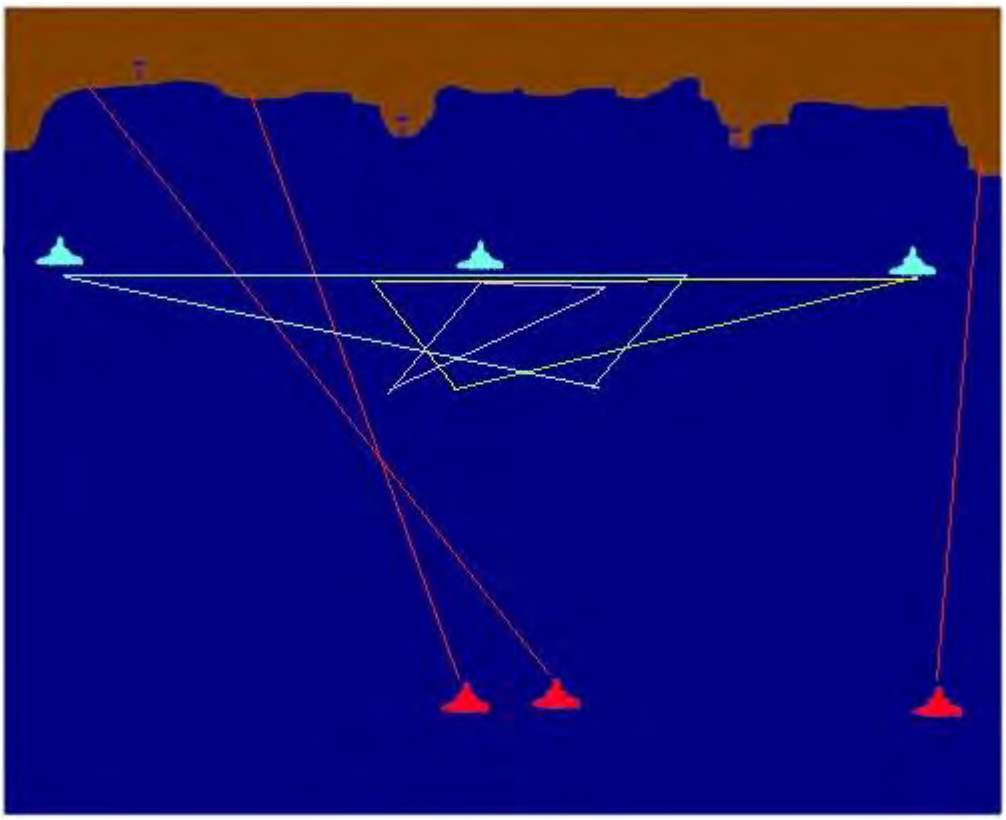

Figure 7.31: The red and blue emerged tactics when the scenario was optimized using the CEAN at a mutation rate of 40%

Figure 7.31 shows the blue and red team‟s tactics that were evolved when the scenario was optimizing using the CEAN. The red team uses deception tactics as one of the red boat sneaks through the right corner while two others grab the attention of surveillance patrols. This strategy may maximize their aim of reaching the coastline as one red boat uses a flanking strategy. In order to counteract the red team‟s strategy, the blue team expanded the surveillance area. This may be a wise strategy for the blue boats as they cannot focus on only a specific area which could increase risks of penetration.

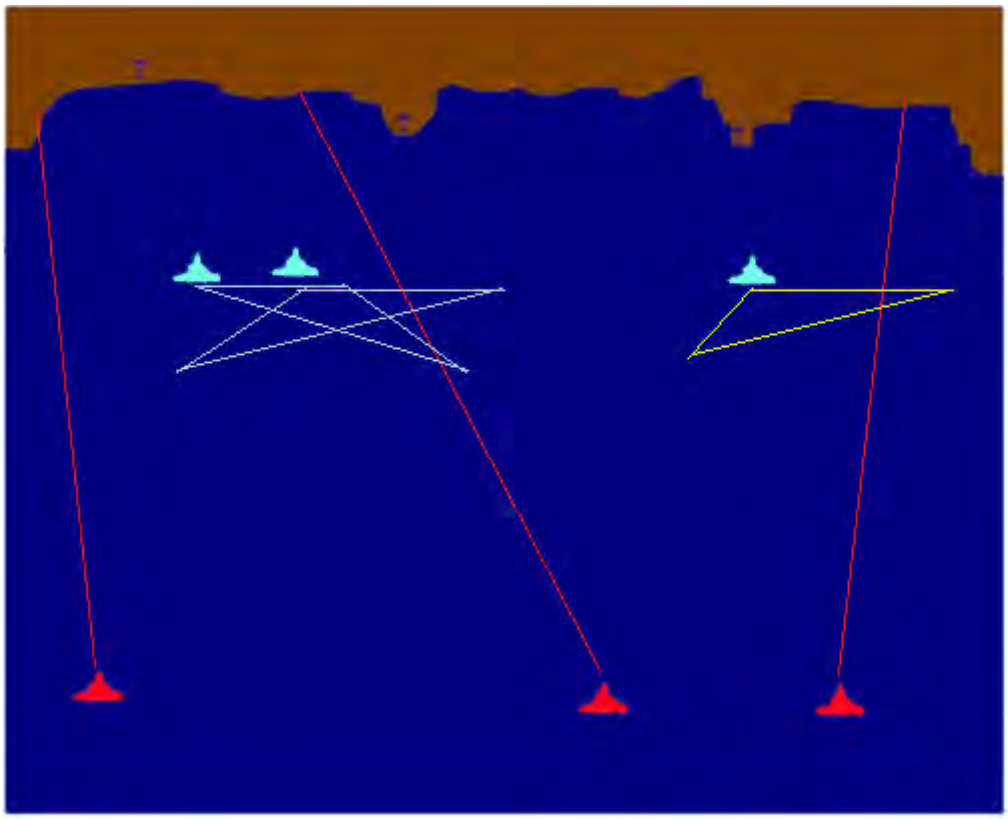

Figure 7.32: The red and blue emerged tactics when the scenario was optimized using the CEAFS at a mutation rate of 40%

Likewise, Figure 7.32 shows the evolved strategies for the blue and red teams when the scenario was optimized using CEAFS. The red team again followed deception tactics in which two boats distract the blue patrols and another red boat sneaks from the left side. As mentioned in the CEAN scenario analysis, flanking tactics are always a strong strategy for the red boats especially when they are unarmed and have to face armed opponents. The blue boat targeted the potential route of the red boats and did not widen the patrolling area. In this scenario, the blue team‟s strategy could be economic; however, in reality, their surveillance would not be effective to stop red boats if similar to the evolved route shown above were used by the red boats.

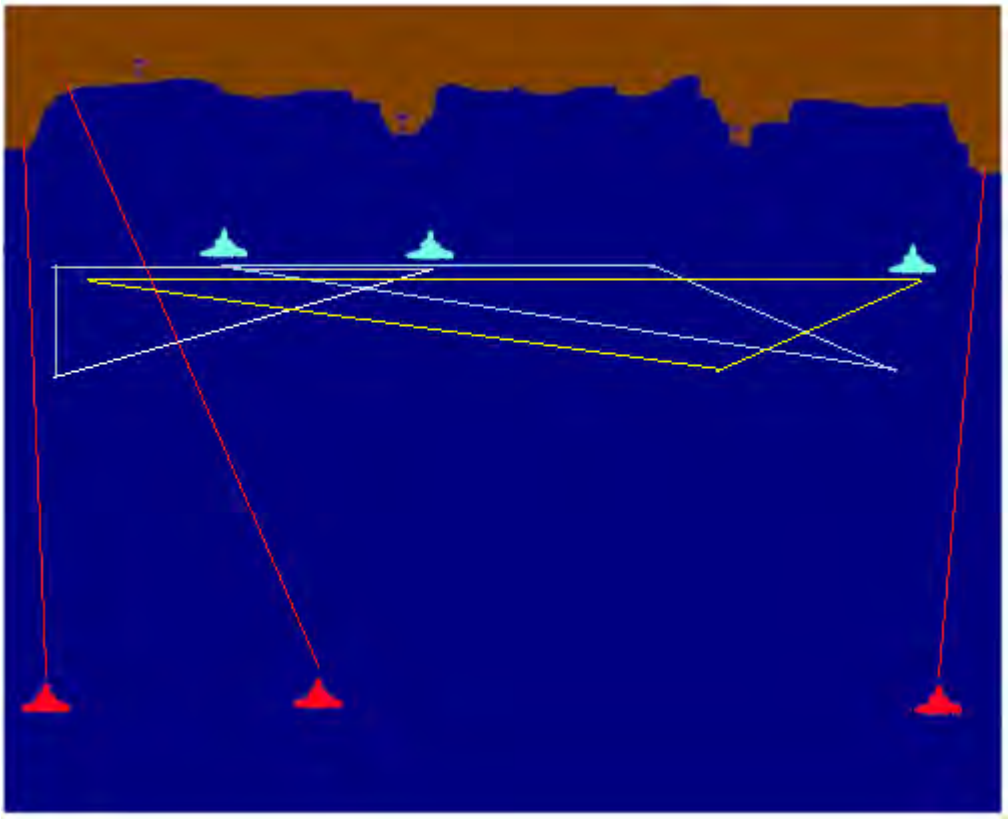

Figure 7.33: The red and blue emerged tactics when the scenario was optimized using the CEAHOF in 40% mutation rate

Figure 7.33 shows the evolved tactics when the scenario was optimized using the CEAHOF. Similar to the previous two scenarios, the red boats follow a deception strategy in which two boats sneak from two opposing sides when another red boat distracts the patrolling blue boats. This was an effective tactics for the red boats that maximize their aim of reaching the coastline. The blue boats widen their surveillance area to stop sneaking boats. This strategy could best address the red team‟s flanking strategy.

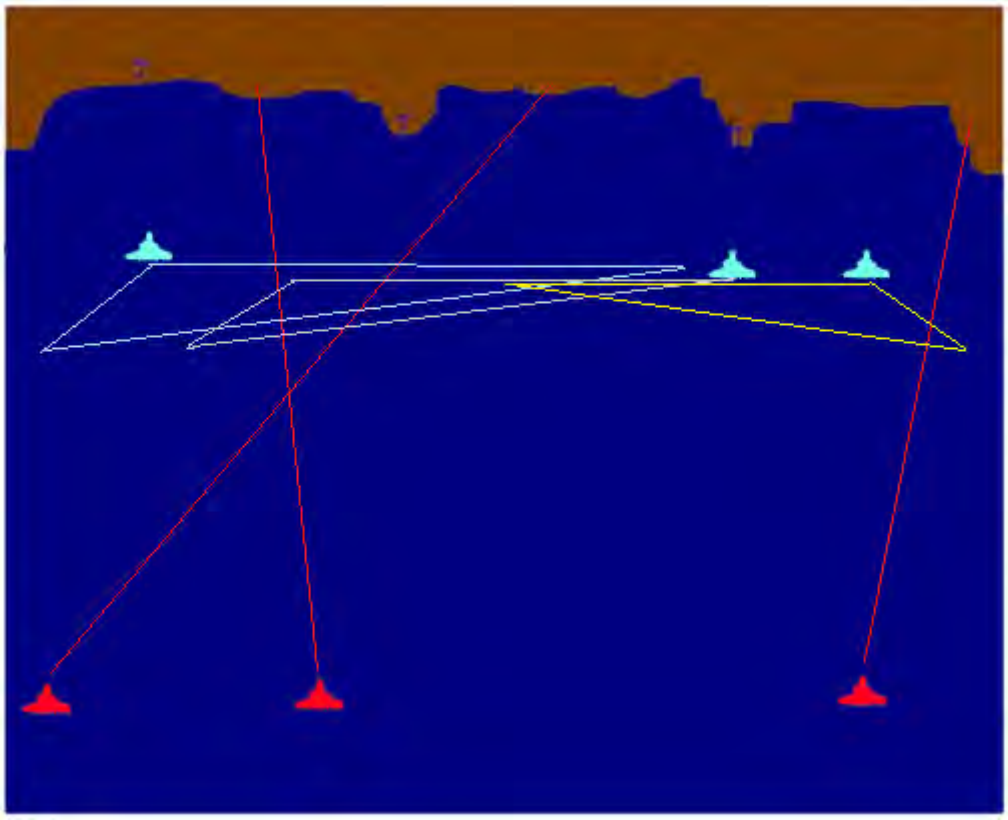

Figure 7.34: The red and blue emerged tactics when the scenario was optimized using the CEACFH in 40% mutation rate

Figure 7.34 shows the red boat and blue boat‟s tactics that emerged from the CEACFH. The red teams followed the penetration tactics by following a relatively direct route from different locations. This strategy may lead them to reaching the coastline if they could trick the patrolling boats. However, this red strategy could increase their attrition rate if they are caught by the blue surveillance. The blue boats extended their surveillance area focussing their patrol on locations from where there could be more attempts at penetration. This strategy could be less expensive in comparison to the strategy obtained from CEAHOF but if the red boats use deception tactics, this blue strategy may not be effective to stop their penetration attempt.

The analysis of the four scenario from each of the four algorithms shows that the red boats use flanking strategy (avoid direct confrontation) to reach the coastline. Some red boats were found to be using a direct confrontation strategy which may be to distract the attention of the blue boats. The blue boats, except in the scenario from CEAFS, widen their surveillance area in order to counter the red teams‟ flanking strategy. However, the

blue boats in CEAFS detect the anticipated route of the red boats and patrol only in that area.

7.5 Conclusion

Experiments were conducted to optimise two RT scenarios, anchorage protection and coastline protection, using a naïve CEA and three variants in this study. The challenge for these algorithms was to find optimum strategies for the team that could best counteract the opposing strategies. Each of these four algorithms has been evaluated on the intransitive number problem and multimodal problem described in chapter 5 and 6 respectively. In this chapter, the performance of the algorithms was also measured by their generalising ability and also their ability to locate multiple optimal solutions.

When measuring the performance of the algorithms via their generalising ability, each of the four algorithms appeared to be similar except CEACFH in the anchorage protection scenario for the blue team. Additionally, the same algorithm has low performance in the coastline protection scenario for the red team. It was expected that fitness sharing would perform better than other algorithms; however, it seems that higher diversity is not the most important factor in evolving solution in both these RT problem scenarios. The reason may be that fitness sharing was based on the genomes and a small change in genome makes a large influence to a team‟s strategy. The behavioural diversity could be more supportive; however, it is not practicable as there are no suitable methods to distinguish strategies except manual observation via simulation.

However, in terms of locating multiple optimal solutions, CEAFS and CEAHOF appeared to be the better algorithms in both scenarios studied. It was expected that CEAFS would locate many local optimal solution as a higher diversity encourages a more complete exploration of the search space. However, the good performance of CEAHOF was unexpected and an explanation of this result could be a possible subject for future research.

In addition, some evolved strategies were also explained in this chapter. Those strategies provide alternatives to analysts to address existing vulnerability in their security plans.

In this empirical study a maximum of only 5 red boats and 3 blue boats in the coastline protection scenario and a maximum of only 10 green boats, 5 red boats and 3 blue boats in the anchorage protection scenario is considered. If ever required to optimize a scenario that includes large number of boats (MANA allows up to 1000 number of agents in each team), the developed optimization tool does not require any changes to optimize the scenario. Despite varying the number of boats in the scenario, the search space remains the same. Due to the stable search space, the optimization tool can optimize scenarios with a small to large number of boats. As depicted in the result in the pilot study in chapter 3, when the number of boats involved in the scenario varied, the approach produced different outcomes in terms of the strategies incorporated by the boats. Thus, when a large numbers of boats are involved in a scenario, the strategy for the invaders and defenders will be obviously different. Additionally, the number of boats in a scenario heavily influence the computation time for running simulations. If the number of boats is symbolized by n, computational time is expected to be $O(n^2)$, which indicates that more boats makes the optimization process much slower. However, the computational time can be reduced by using one or more of the following options:

1) Adding more workstations on the Shoal cluster
2) Reducing the number of simulations
3) Using a one-to-some interaction approach in the CEA rather than all-to-all
4) Facilitating with a cloud computing.

All the above options have some drawbacks such as, each additional computer system would increased computing cost. If the number of simulation is reduced, the level of noise increases in the fitness of individuals, that is, the same individual may receive high or low fitness in different runs. The one-to-some interaction approach reduces computational time; however, it allows individuals to compete against limited strategies. The performance of cloud computing is heavily dependent on the internet or intranet service. The cloud computing would be hard to use in the area where the internet speed is slow.

8 Conclusions and Future Directions

This chapter summarises the key findings of this research relating to RT applications and other related areas. Additionally, during the course of this study, some potential areas for additional investigation were identified. The first section presents the conclusion of this thesis, followed by a discussion of limitations of the study and suggestions for future work.

8.1 Conclusion

The literature demonstrated that RT techniques have long been used in various applications including military applications. Traditional RT is expensive and time consuming. Computerized military RT was easy to use effectively; however, finding the optimal strategy that best counteracts the opponent‟s strategy was still a tough challenge. This thesis provides contributions to the area of optimization for RT and other associated CEA applications. This study aimed to:

1. Investigate approaches incorporating EAs, specifically GAs and CEAs, for finding good solution sets for RT scenarios and other similar applications.
2. Identify suitable techniques that enhance CEAs. Incorporate the identified variants in CEAs for investigating the issues of intransitivity and multimodality in RT scenarios and other similar domains.
3. Investigate suitable measures to evaluate CEAs‟ performance in various problems, including RT.

In order to achieve these purposes, RT applications and evolutionary algorithms were investigated. This thesis presented a systematic study of CEAs, with and without common enhancements, for finding good solution sets in the context of RT. During analysis, this study also evaluated the general applicability of competitive CEAs using two other problems, the intransitive number problem and the multimodal problem. Details of results for each of the aims of the thesis are now summarised.

In addressing the first and second aims, this study extends knowledge of factors affecting the application and performance of evolutionary algorithms, in the context of

RT and other similar applications, by conducting a systematic study involving GA, a basic CEA and three variants (CEAFS, CEAHOF and CEACFH). The study involving GA was described in Chapter 3, showing the limitation associated with GAs supporting the optimization of a single population at a time. An observation from this study showed that by fixing the strategy of one competing team, various strategies capable of defeating the opposition can be evolved for the optimized team. However if in turn, the optimal strategy of the optimised team is then used as the fixed strategy and a GA is used to optimised strategies for the former team, it is possible to find some good strategies for the former team that is able to defeat that fixed optimal strategy. This indicated that even "optimal" strategies may be defeated when the opponent team is optimized against them. The optimization tool, described in Chapter 3, searched for a best strategy that could counteract one known fixed opponent‟s strategy. However, in combat it is never enough to consider or hypothesize just one, or even a few, opponents‟ strategies and practice to defeat those plans. In reality, teams in combat need to adjust strategies to react to different strategies that opponent may utilize at different points. CEAs were identified as suitable algorithms which were capable of optimizing two teams simultaneously for RT applications.

The incorporation of CEAs for RT is still in its infancy and existing studies that used CEAs have yet to explore issues associated with the pathologies associated with CEAs and characteristics such as intransitivity and multimodality. This thesis presents a more complete and systematic study of these issues in the following way. This study carried out a systematic study of variants of CEAs (i.e. CEAN, CEAFS, CEAHOF and CEACFH) on four different test problems; one with intransitivity, followed by one with multimodality and lastly two RT scenarios. In terms of the variants of CEA, they are: naïve CEA (CEAN), FS integrated into the naïve CEA (CEAFS), HOF integrated with the naïve CEA (CEAHOF) and lastly HOF and FS were both integrated into the naïve CEA (CEACFH). It has been shown that individually, FS and HOF can enhance the performance of CEAs. However, there is an absence of literature in which these two well known techniques are jointly integrated in CEAs (or EAs). FS was included as an example of an implicit diversity maintenance technique. In addition, the performance of these algorithms in these test problems was investigated by increasing mutation rate, an explicit diversity maintenance technique. Chapter 5, 6 and 7 outlined these

investigations and described associated results. The general observations from the application of the four algorithms in these four different problems show that the performance of these algorithms in terms of GP and diversity measures appears to differ in these domains.

Chapter 5 described the investigation to test whether CEA variants are capable of addressing intransitivity. They were evaluated on a well-known intransitive number problem. The performance of these algorithms was measured in terms of the GP and also the objective quality of the solutions was measured. The diversity of the populations was also evaluated on the basis of genomes and fitness of individuals. The experiments showed that in order to achieve higher objective quality, a naïve CEA‟s performance could be much improved when combined with a fitness sharing approach. In addition, it was found that the use of higher mutation rates in the naïve CEA could achieve higher GPs. The relationship between the two types of diversity indicated that if individuals are diverse genetically, they receive diverse fitness. Additionally, the relationship between quality and diversity shows that more diverse populations perform better in terms of achieving higher quality.

In the context of RT, there may be more than one good strategy to defeat an opposition‟s plans. In addition, in the literature, it is stated that multimodality commonly appears in most domains. Therefore, a technique developed to test for multiple optima in an evolved population was described in Chapter 4. The result of the pilot study was used for this analysis. It was found that RT demonstrated more than one locally optimal solution. This led to the development of a scalable multimodal problem, *n-peak*, that has been described in Chapter 6. Subsequent investigation involved using the 5-peaks multimodal problem. The challenge for CEAs was to identify a pre-defined set of peaks. Subsequently, the performance of the four CEA variants was measured using GP, CEMD, PR and SR. The diversity of the populations was also measured. It was found that, rather than using only a naïve CEA, the algorithm‟s performance could be improved if it was used with a combination of the HOF and the FS approaches in this domain. The relationship between quality and diversity shows that diverse populations can achieve higher quality and also find more peaks than non-diverse population in a multimodal domain problem.

The optimization tool, which was used in the intransitive number and multimodal problems, was subsequently used in two RT scenarios. This is described in Chapter 7. These two scenarios were created using the MANA simulator and have different environments and objectives. In the pilot study, the GA was used to optimize a RT scenario. However, GAs cannot evolve two teams simultaneously. CEAs, with and without common enhancements, were used to optimize the previously studied scenario as they can evolve two populations simultaneously. For the enhancement of the CEA‟s performance, variants such as the HOF and FS were integrated. The quality of the algorithms used in this study was measured using GP. This thesis has also offered an in-depth analysis of the strategies which emerged for the blue and red teams in the RT application. A multimodal test was also conducted on the evolved strategies to check the number of local optima that existed in the evolved populations.

It was found that both teams performed better with a certain amount of diversity in both scenarios. The populations in the CEAFS were highly diverse as well as including high numbers of local optima, i.e. many good solutions rather than one the best solution existed. However, in terms of GP, the CEAFS‟s performance was relatively poorer than CEAN. The CEACFH also highly diversified the populations; however, its performance in achieving GP was relatively low and it failed to produce populations with high numbers of local optima.

This empirical study demonstrated that the naïve algorithm, which suffers from various pathologies including cycling and forgetting, may perform well with high mutation rates. The results suggested that the higher *mutation rate* was beneficial, not only in diversifying the populations but also to improve the performance of the algorithms. The CEAN, when used with higher mutation rate, produced good quality populations that scored high GP. However, the multimodal test showed that the population did not evolve as many local optima as the other algorithms evaluated.

Another finding from this thesis relates to the application of high mutation rates (a range of 2.5% to 100%). A higher mutation rate forces higher diversity; however, a higher diversity caused by extreme mutation may not give a favourable outcome in

terms of finding the optimal solution. Wright (1986) has stressed for the balance between genetic homogeneity and heterogeneity, which also support the argument that extreme diversity may not always be good. This study tries to examine the effect, of systematically increasing the mutation rate to its extreme, on the performance of CEAs. The results from Chapter 5 and 6 provide support that extreme mutation does not result in favourable outcomes as in many instances the performances of the algorithms starts to deteriorate when the mutation rates exceeds some specific values.

In terms of addressing the third aim, this study investigated suitable measures to evaluate the performance of CEAs. In CEAs, the populations evolve by evaluating each individual against the opposing population. When a population improves its performance against its opposing population, this in turn forces the competing population to improve its performance. An „arms race" occurs in which the populations eventually get better performing individuals, which may be seen through comparison with an external criterion, such as a fixed test population. However, their subjective fitness will remain unchanged or, at least, very similar because the opposing populations are also evolving simultaneously. Therefore, CEAs" performance cannot be measured on the basis of their subjective fitness. In order to measure the quality of CEAs" population, a technique called „generalisation performance" (GP) was used. According to this technique, archive populations were created for each team. The archive population was a non-evolving fixed set of solutions which were selected on the basis of their performance against the randomly generated set of solutions. In every generation, evolving teams were evaluated against their respective archive population which showed the eventual progress of the evolving population. Therefore, the GP represents the quality of the population.

The GP as a quality measurement technique was used in all domains studied in this thesis. Additional performance measures were also used in specific domains. In the intransitive number problem, the performance of the algorithms was also measured on the basis of their objective fitness. This thesis also introduced a probability distribution method, circular earth movers" distance (CEMD), to measure the performance of CEAs in their ability to detect multiple peaks. CEMD has been widely used in image processing applications. In addition, peak ratio (PR) and success ratio (SR) were

utilized to rate algorithms on the basis of the proportion of the total number of peaks detected.

The diversity of the population was also measured in two ways: genotypic and phenotypic. Genotypic diversity was measured by calculating distance between the individuals‟ genes within a population. Likewise, phenotypic diversity was measured on the basis of fitness of individuals within a population. After measuring the quality and diversity of populations, a relationship between diversity and quality was also analysed in each of domain problem studied in this thesis.

8.2 Limitations and Future Research

The previous section has described the conclusions associated with this research. This section details the limitations of this study and suggests possible options for ongoing research into automated red teaming. Intuitively, coevolution involves a number of individuals‟ interactions to calculate their fitness done by simulating an interaction scenario. Additional scenario simulations are required when performance needs to be measured with a fixed test set. This characteristic of CEAs enormously increases the computational time to conduct RT optimization. Therefore, to economize the computational time, certain parameters in the study associated with the RT scenarios have been constrained as follows:

- Population size of 15 was chosen for each team in both RT scenarios evaluated
- Only 10 simulation runs were performed for evaluating each pair of strategies
- Each run of each algorithm used only 50 generations
- For statistical variation, only 15 runs of each algorithm were executed at each mutation rate
- RT optimization was tested only at 4 levels of mutation rate.

An increase in population size might suggest more realistic strategies. An increase of simulation runs would reduce the noise and an increase in the number of generations could provide more optimum results. Since the RT outcomes were highly fluctuating, if mutation rates can be varied in stepwise increments of 2.5%, a clearer picture of the algorithms‟ performances could be obtained.

Due to time constraints, this research was limited to investigations involving the integration of HOF and FS in CEAs. Other techniques addressing the pathologies of CEA can be investigated for RT applications.

An additional limitation of this study would be the issue of scalability in RT scenario. This thesis considered only a small number of boats in both scenarios studied. If the number of boats is increased, it will increase the computational time.

References

Abbass, H. A., Bender, A., Gaidow, S., & Whitbread, P. (2011). Computational red teaming: Past, Present and Future. *IEEE Computational Intelligence Magazine*, 30-42.

Abbass, H. A., Sarker, R., & Newton, C. (2001). PDE: A pareto-frontier differential evolution approach for multi-objective optimization problems. *Proceedings of the Congress on Evolutionary Computation 2001, 2*, 971-978.

Abraham, A., Jain, L., & Goldberg, R. (2005). Evolutionary multi-objective optimization: Theoretical advances and applications X. Wu & L. Jain (Eds.), *Springer*

Alcala, R., Alcala-Fdez, J., Gacto, M. J., & Herrera, F. (2007). A multi-objective evolutionary algorithm for rule selection and tuning on fuzzy rule-based systems. *International Journal of Uncertainty: Fuzziness and Knowledge-based Systems, 15*(5), 539-557.

Andrews, P. (2005). Red teams: Toward radical innovation *IBM Advanced Business Institute in Palisades*. New York: IBM business consulting services.

Avery, P., & Louis, S. (2010). *Coevolving team tactics for a real-time strategy game*. Paper presented at the WCCI 2010 IEEE World Congress on Computational Intelligence, Barcelona, Spain.

Avery, P., Michalewicz, Z., & Schmidt, M. (2008). Short and long term memory in coevolution. *International Journal of Information Technology and Intelligent Computing, 3*(1).

Axelrod, R. (1987). The evolution of strategies in the iterated prisoner's dilemma. *Genetic Algorithms and Simulated Annealing, 3*, 22-41.

Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. *In Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and their Applications*, 14-21.

Barlow, M., & Easton, A. (2002). Corcadile - An open, extensible agent-based distillation engine. *Inform. Secur., 8*(1), 17-51.

Beasley, D., Bull, D. R., & Martin, R. R. (1993). A sequential niche technique for multimodal function optimization. *Evolutionary Computation, 1*(2), 101-125.

Buckle, P., Moore, T., Robertshaw, S., Treadway, A., Tarkoma, S., & Poslad, S. (2002). Scalability in multi-agent systems: The FIPA-OS perspective. *Foundations and applications of multi-agent systems, 2403*, 110-130.

Bui, L. T., & Barlow, M. (2003). *Performance of path planning agents in agent-based distillation.* Paper presented at the SimTect T-2003 Conference, Adelaide, Australia.

Buro, M. (2003). Real-Time Strategy Games: A New AI Research Challenge. *In Proceedings of the 18th International Joint Conference on Artificial Intelligence*.

Casillas, J., Cordon, O., Herrera, F., & Merelo, J. J. (2002). A cooperative coevolutionary algorithm for jointly learning fuzzy rule bases and membership functions. *Artificial Evolution, 2310*, 1075-1105.

Chong, S. Y., Tino, P., & Yao, X. (2008). Measuring generalization performance in coevolutionary learning. *IEEE Transactions on Evolutionary Computation, 12*(4), 479-505.

Chong, S. Y., Tino, P., & Yao, X. (2009). Relationship between generalization and diversity in coevolutionary learning. *IEEE Transactions on Computational Intelligence and AI in Games, 1*(3), 214-232.

Choo, C. S., Chua, C. L., Low, K. M. S., & Ong, W. S. D. (2009). *A coevolutionary approach for military operational analysis*. Paper presented at the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation, New York, USA.

Choo, C. S., Chua, C. L., & Tay, S. V. (2007). *Automated red teaming: A proposed framework for military appliaction.* Paper presented at the 9th Annual Conference on Genetic and Evolutionary Computation, London, England, United Kingdom.

Chua, C. L., Sim, C. W. C., Choo, C. S., & Tay, V. (2008). *Automated red teaming: An objective-based data farming approach for red teaming*. Paper presented at the 40th Conference on Winter Simulation.

Chung, M., Buro, M., & Shaeffer, J. (2005). *Monte-Carlo planning in RTS games.* . Paper presented at the In proceedings of the Computational Intelligence and Games, UK.

Churchill, D., & Buro, M. (2011). Build order optimization in StarCraft. *In Proceedings of AIIDE*.

Churchill, D., & Buro, M. (2012). Incorporating search algorithms into RTS game agents. *Artificial Intelligence in Adversarial Real-Time Games*.

Cioppa, A. D., Stefano, C. D., & Marcelli, A. (2004). On the role of population size and niche radius in fitness sharing. *IEEE Transactions on Evolutionary Computation, 8*(6), 580-592.

Claypool, M. (2005). The effect of latency on user performance in real-time strategy games. *Computer Networks, 49*, 52-70.

Cleary, P., Ball, D., Madahar, M., & Thorne, S. (2008). Investigating the use of software agents to reduce the risk of undetected errors in strategic spreadsheet applications. *Computing research repository - CORR*.

Cliff, D., & Miller, G. F. (1995). Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations *Advances in Artificial Life* (Vol. 929, pp. 200-218).

Coello, C. A. C. (1999a). *An updated survey of evolutionary multi-objective optimization techniques: State of the art and future trends*. Paper presented at the Congress on Evolutionary Computation.

Coello, C. A. C. (1999b). A comprehensive survey of evolutionary-based multi-objective optimization techniques. *Knowledge and Information Systems, 1*(3), 129.

Coello, C. A. C., Lamont, G. B., & Veldhuizen, D. A. V. (2007). *Evolutionary algorithm for saolving multi-objective problems*. New York: Springer Science + Business Media.

Coello, C. A. C., & Pulido, G. T. (2001). *A micro genetic algorithm for multi-objective optimization*. Paper presented at the Evolutionary Multi-criterion Optimization.

Danculovic, J. P. (2007). Clustering with the Shoal Framework. Retrieved from http://today.java.net/pub/a/today/2007/12/11/clustering-with-shoal-framework.html

Davidsson, P., & Boman, M. (2000). A multi-agent system for controlling intelligent buildings. *4th IEEE International Conference on Multi-agent Systems*, 377-378.

Davidsson, P., & Wernstedt, F. (2002). *A multi-agent system architecture for coordination of just-in-time production and distribution*. Paper presented at the APM Symposium on Applied Computing.

Deb, K., & Goldberg, D. E. (1989). *An investigation of niche and species formation in genetic function optimization*. Paper presented at the 3rd International Conference on Genetic Algorithms, San Mateo, CA.

Deb, K., & Goyal, M. (1996). A combined genetic adaptive search (gene AS) for Engineering Design. *Computer Science and Informatics, 26*(4), 30-45.

Decraene, J., Chandramohan, M., Malcolm, Y. H.-L., & Choo, C. S. (2010). *Evolvable simulations applied to automated red teaming: A preliminary study*. Paper presented at the Proceedings of the 2010 Winter Simulation Conference.

DeJong, E. D. (1975). An analysis of the behavior of a class of genetic adaptive systems. *Dissertation Abstract International, 36*(10).

DeJong, E. D. (2004). *Intransitivity in coevolution*. Paper presented at the Parallel Problem Solving from Nature PPSN VIII.

DeJong, E. D., Stanley, K., & Wiegand, P. (2007). Introductory tutorial on coevolution. *Genetic and Evolutionary Computation Conference*, 3133-3157. doi: 10.1145/1274000.1274108

Department of Defence. (2003). The role and status of DoD red teaming activities. *Defense Science Board Task Force*.

Eiben, A. E., & Smith, J. E. (2003). *Introduction to evolutionary computing*. Verlag Berlin Heidelberg New York: Springer.

Eiben, A. E., & Smith, J. E. (2007). *Introduction to evolutionary computing: Natural computing series*: Springer.

Epitropakis, M. G., Plagianakos, V. P., & Vrahatis, M. N. (2011). *Finding multiple global optima exploiting differential evolution's niching capability*. Paper presented at the IEEE Symposium on Differential Evolution, Paris, France.

Ficici, S. G. (2004). *Solution concepts in coevolutionary algorithms.* Doctor of Philosophy, Brandeis University.

Ficici, S. G., & Pollack, J. B. (2001). *Pareto optimality in coevolutionary learning*. Paper presented at the 6th European Conference on Advances in Artificial Life, London, Uk.

Ficici, S. G., & Pollack, J. B. (2003). *A game-theoretic memory mechanism for coevolution*. Paper presented at the Genetic and Evolutionary Computation Conference.

Figueiraa, J. R., Liefooghe, A., Talbi, E.-G., & Wierzbickie, A. P. (2010). A parallel multiple reference point approach for multi-objective optimization. *European Journal of Operational Research, 205*(2), 390-400.

Flores-Mendez, R. A. (1999). Towards a standardization of multi-agent system frameworks. *Crossroads, 5*.

Fontenot, G. (2005). Seeing red: Creating a red team capability for the blue force. *Military Review, 85*(5), 4.

Garcia-Pedrajas, N., Castillo, J. a. R. d., & Ortiz-Boyer, D. (2009). A cooperative coevolutionary algorithm for instance selection for instance-based learning. *Machine Learning, 78*(3).

Gilbert, E. N. (1965). Random Minimal Trees. *Journal of the Society for Industrial and Applied Mathematics, 13*(2), 376-387.

Giles, L. (2005). *The art of war by Sun Tzu*. Texas: El Paso Norte Press.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co.

Goldberg, D. E., & Richardson, J. (1987). *Genetic algorithms with sharing for multimodal function optimization*. Paper presented at the 2nd International Conference on Genetic Algorithms, Hillsdale NJ.

Guo, Y., Cao, X., & Yin, H. (2007). Coevolutionary optimization algorithm with dynamic sub-population size. *International Journal of Innovative Computing, Information and Control, 3*(2), 435-448.

Han, A. W. C., Chung, S. W., Lian, C. C., Kiat, L. Y., Chin, K. S., Lijun, C. T., . . . Abbott, B. (2007). *Applying automated red teaming in a maritime scenario*. Paper presented at the 15th International Data Farming Workshop, Montery, CA.

Hansen, N., & Kern, S. (2004). *Evaluating the CMA evolution strategy on multimodal test functions*. Paper presented at the Parallel Problem Solving from Nature.

Herrera, F., & Lozano, M. (1996). *Adaptation of genetic algorithm parameters based on Fuzzy Logic controllers*. Paper presented at the Genetic Algorithms and Soft Computing.

Herz, J. C., & Macedonia, M. R. (2002). Computer games and the military: Two views. *Defense Horizons: Center for Technology and National Security Policy*.

Higashi, N., & Iba, H. (2003). Particle Swarm Optimization with Gaussian Mutation. *IEEE Swarm Intelligence Symposium*, 72-79.

Hillis, W. D. (1990). Coevolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena, 42*(1), 228-234.

Hingston, P., & Preuss, M. (2011). *Red Teaming with coevolution*. Paper presented at the 2011 IEEE Congress on Evolutionary Computation.

Hingston, P., Preuss, M., & Spierling, D. (2010). RedTNet: A network model for strategy games. *IEEE Congress on Evolutionary Computation*, 1-9.

Hinterding, R., Michalewicz, Z., & Peachey, T. C. (1996). Self adaptive Genetic Algorithm for numeric functions. *Parallel Problem Solving from Nature, IV*, 420-429.

Hiroyasu, T., Miki, M., Iwahashi, T., Okamoto, Y., & Dongarra, J. (2003). *Dual individual distributed genetic algorithm for minimizing the energy of protein tertiary structure*. Paper presented at the SICE Annual Conference, Fukui, Japan.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*: University of Michigan Press.

Horn, J., & Nafpliotis, N. (1994). *Multiobjective optimization using the niched Pareto genetic algorithm*. Paper presented at the 1st IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Piscataway, NJ.

Ilachinski, A. (2000). Irreducible semi-autonomous adaptive combat (ISAAC): An artificial-life approach to land warfare. *Military Operations Research, 5*(3), 29-46.

Ilachinski, A. (2003). Exploring self-organized emergence in an agent-based synthetic warfare lab. *Kybernetes, 32*(1/2), 38-76.

Innocenti, B., Lopez, B., & Salvi, J. (2003). *Multi-agent system architecture with planning for a mobile robot*. Paper presented at the 10th Conference of the Spanish Association for Artificial Intelligence, CAEPIA.

Jacob, C. (2001). *Illustrating evolutionary computation with mathematica*. San Francisco, USA: Morgan Kaufmann.

Jang, S.-H., Yoon, J.-W., & Cho, S.-B. (2009). Optimal strategy selection of non-player character on real time strategy game using a speciated evolutionary algorithm. *Computational Intelligence and Games*.

Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems, 1*(1), 7-38.

Johnson, L., Yannakakis, G. N., & Togelius, J. (2010). *Cellular automata for real-time generation of infinite cave levels*. Paper presented at the PCGames Monterey, CA, USA.

Koper, K. D., Wysession, M. E., & Wiens, D. A. (1999). Multimodal function optimization with a niching Genetic Algorithm: A seismological example. *Bulletin of the Seismological Society of America, 89*(4), 978-988.

Kouchmeshky, B., Aquino, W., Bongard, J. C., & Lipson, H. (2007). Coevolutionary algorithm for structural damage identification using minimal physical testing. *International Journal for Numerical Methods in Engineering, 69*, 1085-1107.

Lanchester, F. W. (1916). *Aircraft in Warfare: The dawn of the fourth arm*. London: Constable Press.

Lauren, M. K. (1999). Characterising the difference between complex adaptive and conventional combat models *DOTSE Report 169*.

Lauren, M. K. (2002). A metamodel for describing the outcomes of the MANA cellular automaton combat model based on lauren's attrition equation *DTA Report 205*.

Lauren, M. K., & Stephen, R. (2002). MANA: Map aware non-uniform automata. *A New Zealand approach to scenario modeling, J. Battlefield Technol, 5*(1), 27-31.

Levin, S. A. (2002). Complex adaptive systems: Exploring the known, the unknown and the unknowable. *New Series of the American Mathematical Society, 40*(1), 3-19.

Levina, E., & Bickel, P. (2001). The earth mover's distance is the mallows distance: Some insights from statistics. *8th IEEE International Conference on Computer Vision*, 251-256.

Ling, H., & Okada, K. (2006). An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*(5), 840-853.

Liss, C. (2007). The Privatisation of Maritime Security - Maritime Security in Southeast Asia: Between a rock and a hard place? *Asia Research Centre*.

Liu, Y., & Zhang, A. (2008). Multi-Agent System and Its Application in Combat Simulation. *2008 International Symposium on Computational Intelligence and Design*, 448-492.

Mahfoud, S. W. (1995). *Niching method for Genetic Algorithm.* PhD, University of Illinois, Urbana-Champaign.

McIntosh, G. C., Galligan, D. P., Anderson, M. A., & Lauren, M. K. (2007). MANA: Map Aware Non-Uniform Automata Version 4 User Manual.

Mckay, R. I. (2000). *Fitness sharing in genetic programming*. Paper presented at the Genetic and Evolutionary Computation Conference, Las Vegas, USA.

Meehan, M. K. (2007). Red Teaming for Law Enforcement. *Police Chief, 74*(2).

Miles, C. E. (2007). *Co-evolving real time strategy game players.* University of Nevada.

Miller, B. L., & Shaw, M. J. (1995). *Genetic algorithms with dynamic niche sharing for multimodal function optimization*. Paper presented at the Evolutionary Computation-Proceedings of IEEE International Conference, Nagoya Japan.

Miller, G. F., & Todd, P. M. (1993). *Evolutionary wanderlust: Sexual selection with directional mate preferences*. Paper presented at the 2nd International Conference on Simulation of Adaptive Behaviour, Cambridge, MA

Mora, A. M., Fernandez-Ares, A., Merelo, J. J., Garca-Sanchez, P., & Fernandes, C. M. (2012). Effect of noisy fitness in real-time strategy games player behaviour optimisation using evolutionary algorithms. *Journal of Computer Science and Technology 27*(5), 1007-1023.

Munir, N., Kitchin, D., & Crampton, A. (2010). Monte-Carlo planning for pathfinding in real time strategy games. *In Proceedings of PlanSIG*.

Nolfi, S., & Floreano, D. (1998). Coevolving predator and prey robots: Do Arms Races arise in artificial evolution? *Artificial Life, 4*(4), 311-335.

Otani, T., Suzuki, R., & Arita, T. (2011). *A new mutation operator for multimodal optimization with differential evolution*. Paper presented at the Australasian Joint Conference on Artificial Intelligence, Perth, Australia.

Othman, N., Decraene, J., Cai, W., Hu, N., Low, M. Y. H., & Gouaillard, A. (2012). Simulation-based optimization of StarCraft Tactical AI through evolutionary computation. *IEEE Conference on Computational Intelligence and Games*.

Parsopoulos, K. E., & Vrahatis, M. N. (2004). On the computation of all global minimizers through particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*(3), 211-224.

Parunak, H. V. D. (2007). MAS combat simulation. *Whitestein Series in Software Agent Technologies*, 131-150.

Pohlheim, H. (2006). Genetic and evolutionary algorithm toolbox for use with MATLAB. *GEATbx, 3*(8).

Poli, R., & Langdon, W. B. (1997). A new schema theorem for genetic programming with one-point crossover and point mutation. *In Proceedings of 2nd Annual Conference of Genetic Programming*, 278-285.

Ponsen, M. (2004). *Improving adaptive game AI with evolutionary learning.* Delft University of Technology.

Porter, M. A., & DeJong, K. A. (1994). A cooperative coevolutionary approach to function optimization. *In Proceedings from the 3rd Parallel Problem Solving from Nature, 886*, 249-257.

Poslad, S., Buckle, P., & Hadingham, R. (2000). *The FIPA-OS agent platform: Open source for open standards*. Paper presented at the 5th International Conference on the Practical Application of Intelligent Agents and Multi-agent Technology PAAM, Manchester, UK.

Puppala, N., Sen, S., & Gordin, M. (1998). *Shared memory based cooperative coevolution*. Paper presented at the IEEE World Congress on Computational Intelligence, Anchorage Alaska USA.

Rabin, J., Delon, J., & Gousseau, Y. (2008). *Circular earth mover's distance for the comparison of local features*. Paper presented at the International Conference on Pattern Recognition.

Ranjeet, T., Hingston, P., Lam, C.-P., & Masek, M. (2012). *Evaluating coevolution on a multimodal problem*. Paper presented at the Genetic and Evolutionary Computation Conference, Philadelphia, Pennsylvania, USA.

Ray, T. S. (1993). Evolution, complexity, entropy and artificial reality. *Physica D: Nonlinear Phenomena, 75*(1-3), 239-263.

Ronkkonen, J., Li, X., Kyrki, V., & Lampinen, J. (2008). A generator for multimodal test functions with multiple global optima. *In Proceedings of the 7th International Conference on Simulated Evolution and Learning*.

Rosin, C. D. (1997). *Coevolutionary search among adversaries.* PhD, University of California, San Diego.

Rosin, C. D., & Belew, R. K. (1997). New methods for competitive coevolution. *Evolutionary Computation, 5*(1), 1-29.

Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision, 40*(2), 99-121.

Sameer, A., Zhao, W., Tang, J., Lokan, C., Ellejmi, M., Kirby, S., & Abbass, H. (2012). Discovering delay patterns in arrival traffic with dynamic continuous descent approaches using co-evolutionary red teaming. *Air Traffic Control Quarterly, 20*(1), 47.

Sareni, B., & Krahenbuhl, L. (1998). Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation, 2*(3), 97-106.

Schadd, F., Bakkes, S., & Spronck, P. (2007). Opponent modeling in real-time strategy games. *In 8th International Conference on Intelligent Games and Simulation*.

Schumacher, M. (2001). *Objective coordination in multi-agent system engineering: Design and implementation*. Berlin - Heidelberg: Springer-Verlag.

Seredynski, F., Zomaya, A. Y., & Bouvry, P. (2003). Function optimization with coevolutionary algorithms. *Advances in Soft Computing, Springer*, 13-22.

Shakshuki, E., & Kajonpotisuwan, P. (2002). Multi-agent system architecture for computer-based tutoring systems. *IAI '02 Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial intelligence*.

Sidran, D. E. (2004). A calculated strategy: Readings directed towards the creation of a strategic artificial intelligence. *Computer Science Readings for Research Spring 2004*.

Singh, G., & Deb, K. (2006). *Comparison of multi-modal optimization algorithms based on evolutionary algorithms*. Paper presented at the In Proceedings of Genetic Evolutionary Computational Conference (GECCO '06), Washington USA.

Smith, R. E., Forrest, S., & Perelson, A. S. (1993). Searching for diverse cooperative populations with genetic algorithms. *Evolutionary Computation, 1*(2).

Talbi, E.-G. (2009). *Metaheuristics: From design to implementation*: John Wiley & Sons, Inc., Hoboken, NJ, USA.

Thiele, L., Miettinen, K., Korhonen, P. J., & Molina, J. (2009). A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation 17*(3), 411-436.

Thomsen, R. (2004). Multimodal optimization using crowding-based differential evolution. *IEEE Congress on Evolutionary Computation, 2*, 1382-1389.

Togelius, J., Preuss, M., Beume, N., Wessing, S., Hagelb¨ack, J., & Yannakakis, G. N. (2010). Multiobjective exploration of the StarCraft map space. *IEEE Symposium*.

Upton, S. C., & McDonald, M. J. (2003). Automated Red Teaming using evolutionary algorithm. *WG31-Computing Advances in Military and Security Applications of Evolutionray Computation, GECCO*.

Ursem, R. K. (2002). Diversity-guided evolutionary algorithms. *Lecture Notes in Computer Science, 2439*, 462-472.

Veldhuizen, D. A. V. (1999). *Multi-objective evolutionary algorithms: Classifications, analysis and new innovations.* PhD, Air Force Institue of Technology Air University.

Watson, R. A., & Pollack, J. B. (2001). *Coevolutionary dynamics in a minimal substrate*. Paper presented at the Genetic and Evolutionary Computation Conference GECCO-01, San Francisco.

Wiegand, R. P. (2003). *An analysis of cooperative coevolutionary algorithms.* Doctor of Philosophy, George Mason University, Virginia.

Wright, S. (1986). *Evolution: selected papers*: University of Chicago Press.

Xu, Y. L., Low, M., & Choo, C. S. (2009). *Enhancing automated red teaming with evolvable simulation.* Paper presented at the First ACM/SIEGVO Summit on Genetic and Evolutionary Computation, ACM, New York, USA.

Yang, A. (2006). A Networked multi-agent combat model: Emergency explained. *University of New South Wales, New South Wales, Australia*.

Yang, A., Abbass, H. A., & Sarker, R. (2006). Characterizing warfare in Red Teaming. *IEEE Transactions on systems, MAN and Cybernetics-Part B: Cybernetics, 36*(2), 268-285.

Yu, E. L., & Suganthan, P. N. (2010). Ensemble of niching algorithms. *Information Sciences, 180*, 2815-2833.

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications* (Vol. 63): Shaker.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation, 8*(2), 173-195.

Zitzler, E., & Kuzli, S. (2004). Indicator-based selection in multiobjective search. *Parallel Problem Solving from Nature-PPSN VIII*.

Appendices

Appendix A

Chapter 5 Related Appendices

Appendix A.1

ANOVA Test for the Estimated Best GP from the Intransitive Number Problem (Section 5.4.1.1)

Dependent Variable: Estimated Best GP

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	2.670[a]	159	.017	10.580	.000	.151
Intercept	9161.525	1	9161.525	5772843.501	.000	.998
Mutation	.706	39	.018	11.402	.000	.045
Algorithm	.892	3	.297	187.306	.000	.056
Mutation * Algorithm	1.072	117	.009	5.775	.000	.067
Error	14.981	9440	.002			
Total	9179.176	9600				
Corrected Total	17.651	9599				

R Squared = .151 (Adjusted R Squared = .137)

Appendix A.2

ANOVA Test for the Estimated Average GP from the Intransitive Number Problem (Section 5.4.1.2)

Dependent Variable: Estimated Average GP

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	146.028[a]	159	.918	227.437	.000	.793
Intercept	5322.628	1	5322.628	1318103.775	.000	.993
Mutation	.689	39	.018	4.376	.000	.018
Algorithm	142.402	3	47.467	11754.861	.000	.789
Mutation * Algorithm	2.937	117	.025	6.216	.000	.072
Error	38.120	9440	.004			
Total	5506.776	9600				
Corrected Total	184.147	9599				

R Squared = .793 (Adjusted R Squared = .790)

Appendix A.3

ANOVA Test for the Objective Best Quality from the Intransitive Number Problem (Section 5.4.2.1)

Dependent Variable: Objective Best Quality

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	97307.022[a]	159	611.994	8.357	.000	.123
Intercept	7.494E7	1	7.494E7	1023324.162	.000	.991
Mutation	10984.038	39	281.642	3.846	.000	.016
Algorithm	65837.387	3	21945.796	299.690	.000	.087
Mutation * Algorithm	20485.597	117	175.091	2.391	.000	.029
Error	691275.869	9440	73.228			
Total	7.572E7	9600				
Corrected Total	788582.892	9599				

R Squared = .123 (Adjusted R Squared = .109)

Appendix A.4

ANOVA Test for the Objective Average Quality from the Intransitive Number Problem (Section 5.4.2.2)

Dependent Variable: Objective Average Quality

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	154244.229[a]	159	970.089	11.141	.000	.158
Intercept	5.014E7	1	5.014E7	575820.886	.000	.984
Mutation	3363.494	39	86.243	.990	.487	.004
Algorithm	134943.821	3	44981.274	516.597	.000	.141
Mutation * Algorithm	15936.914	117	136.213	1.564	.000	.019
Error	821961.630	9440	87.072			
Total	5.111E7	9600				
Corrected Total	976205.859	9599				

R Squared = .158 (Adjusted R Squared = .144)

Appendix A.5

ANOVA Test for the Genotypic Diversity from the Intransitive Number Problem (Section 5.4.3.1)

Dependent Variable: Genotypic Diversity

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	1.527E6	159	9601.134	209.281	.000	.779
Intercept	1.646E7	1	1.646E7	358730.935	.000	.974
Mutation	121062.407	39	3104.164	67.663	.000	.218
Algorithm	1342305.200	3	447435.067	9752.961	.000	.756
Mutation * Algorithm	63212.623	117	540.279	11.777	.000	.127
Error	433077.386	9440	45.877			
Total	1.842E7	9600				
Corrected Total	1959657.615	9599				

R Squared = .779 (Adjusted R Squared = .775)

Appendix A.6

ANOVA Test for the Phenotypic Diversity from the Intransitive Number Problem (Section 5.4.3.2)

Dependent Variable: Phenotypic Diversity

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	369.653[a]	159	2.325	91.086	.000	.605
Intercept	29834.236	1	29834.236	1168879.307	.000	.992
Mutation	174.003	39	4.462	174.802	.000	.419
Algorithm	153.886	3	51.295	2009.711	.000	.390
Mutation * Algorithm	41.764	117	.357	13.985	.000	.148
Error	240.945	9440	.026			
Total	30444.834	9600				
Corrected Total	610.598	9599				

R Squared = .605 (Adjusted R Squared = .599)

Appendix B

Chapter 6 Related Appendices

Appendix B.1

ANOVA Test for the Estimated Best GP from the Multimodal Problem (Section 6.3.1.1)

Dependent Variable: Estimated Best GP

Source	Type IV Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	4.108[a]	159	.026	79.192	.000	.572
Intercept	2977.304	1	2977.304	9126690.180	.000	.999
Mutation	.990	39	.025	77.825	.000	.243
Algorithm	.832	3	.277	850.194	.000	.213
Mutation * Algorithm	2.285	117	.020	59.879	.000	.426
Error	3.080	9440	.000			
Total	2984.491	9600				
Corrected Total	7.187	9599				

R Squared = .572 (Adjusted R Squared = .564)

Appendix B.2

ANOVA Test for the Estimated Average GP from the Multimodal Problem (Section 6.3.1.2)

Dependent Variable: Estimated Average GP

Source	Type IV Sum of Squares	Df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	3.825[a]	159	.024	159.706	.000	.729
Intercept	1787.624	1	1787.624	1.187E7	.000	.999
Mutation	2.452	39	.063	417.480	.000	.633
Algorithm	.350	3	.117	774.073	.000	.197
Mutation * Algorithm	1.023	117	.009	58.028	.000	.418
Error	1.422	9440	.000			
Total	1792.870	9600				
Corrected Total	5.246	9599				

R Squared = .729 (Adjusted R Squared = .724)

Appendix B.3

ANOVA Test for the CEMD from the Multimodal Problem (Section 6.3.2.1)

Dependent Variable: Circular Earth Movers' Distance

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	7085.988[a]	159	44.566	296.413	.000
Intercept	63616.181	1	63616.181	423117.388	.000
Mutation	2335.828	39	59.893	398.354	.000
Algorithm	2083.759	3	694.586	4619.762	.000
Mutation * Algorithm	2666.400	117	22.790	151.577	.000
Error	1419.315	9440	.150		
Total	72121.483	9600			
Corrected Total	8505.302	9599			

R Squared = .833 (Adjusted R Squared = .830)

Appendix B.4

ANOVA Test for Genotypic Diversity from the Multimodal Problem (Section 6.3.3.1)

Dependent Variable: Genotypic Diversity

Source	Type IV Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	10.282[a]	159	.065	99.836	.000	.627
Intercept	959.077	1	959.077	1480613.204	.000	.994
Mutation	4.043	39	.104	160.029	.000	.398
Algorithm	1.974	3	.658	1015.856	.000	.244
Mutation * Algorithm	4.266	117	.036	56.284	.000	.411
Error	6.115	9440	.001			
Total	975.474	9600				
Corrected Total	16.397	9599				

R Squared = .627 (Adjusted R Squared = .621)

Appendix B.5

ANOVA Test for the Phenotypic Diversity (Section 6.3.3.2)

Dependent Variable: Phenotypic Diversity

Source	Type IV Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	157.414[a]	159	.990	283.082	.000	.827
Intercept	24620.603	1	24620.603	7039860.699	.000	.999
Mutation	54.411	39	1.395	398.922	.000	.622
Algorithm	15.666	3	5.222	1493.156	.000	.322
Mutation * Algorithm	87.337	117	.746	213.441	.000	.726
Error	33.015	9440	.003			
Total	24811.032	9600				
Corrected Total	190.429	9599				

R Squared = .827 (Adjusted R Squared = .824)

Appendix C

Chapter 7 Anchorage Protection Scenario Related Appendices

Appendix C.1

ANOVA for the Estimated Best GPs of the Blue Team in the Anchorage Protection (Section 7.3.1.1)

Dependent Variable: Estimated Best GP of the Blue Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	106.568[a]	15	7.105	4.835	.000	.245
Intercept	114783.825	1	114783.825	78115.208	.000	.997
Mutation	2.611	3	.870	.592	.621	.008
Algorithm	101.520	3	33.840	23.030	.000	.236
Mutation * Algorithm	2.436	9	.271	.184	.996	.007
Error	329.149	224	1.469			
Total	115219.542	240				
Corrected Total	435.717	239				

R Squared = .245 (Adjusted R Squared = .194)

Appendix C.2

ANOVA for the Estimated Best GPs of the Red Team in the Anchorage Protection (Section 7.3.1.1)

Dependent Variable: Estimated Best GP of the Red Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	17.718[a]	15	1.181	.973	.485	.061
Intercept	90129.131	1	90129.131	74222.717	.000	.997
Mutation	1.888	3	.629	.518	.670	.007
Algorithm	12.363	3	4.121	3.394	.019	.043
Mutation * Algorithm	3.467	9	.385	.317	.969	.013
Error	272.005	224	1.214			
Total	90418.853	240				
Corrected Total	289.723	239				

R Squared = .061 (Adjusted R Squared = -.002)

Appendix C.3

ANOVA for the Estimated Average GPs of the Blue Team in the Anchorage Protection (Section 7.3.1.2)

Dependent Variable: Estimated Average GP

Source	Type III Sum of Squares	Df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	155.321[a]	15	10.355	6.483	.000	.303
Intercept	92654.641	1	92654.641	58014.198	.000	.996
Mutation	11.699	3	3.900	2.442	.065	.032
Algorithm	130.355	3	43.452	27.206	.000	.267
Mutation * Algorithm	13.268	9	1.474	.923	.506	.036
Error	357.751	224	1.597			
Total	93167.714	240				
Corrected Total	513.072	239				

R Squared = .303 (Adjusted R Squared = .256)

Appendix C.4

ANOVA for the Estimated Average GPs of the Red Team in the Anchorage Protection (Section 7.3.1.2)

Dependent Variable: Estimated Average GP

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	35.255[a]	15	2.350	2.124	.010	.125
Intercept	76273.102	1	76273.102	68941.692	.000	.997
Mutation	10.018	3	3.339	3.018	.031	.039
Algorithm	15.109	3	5.036	4.552	.004	.057
Mutation * Algorithm	10.128	9	1.125	1.017	.427	.039
Error	247.821	224	1.106			
Total	76556.177	240				
Corrected Total	283.075	239				

R Squared = .125 (Adjusted R Squared = .066)

Appendix C.5

ANOVA for the Genotypic Diversity in the Anchorage Protection for the Blue Team (Section 7.3.3.1)

Dependent Variable: Genotypic Diversity for the Blue Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	133131.910[a]	15	8875.461	100.334	.000	.870
Intercept	2613108.837	1	2613108.837	29540.407	.000	.992
Mutation	64253.124	3	21417.708	242.121	.000	.764
Algorithm	62147.133	3	20715.711	234.185	.000	.758
Mutation * Algorithm	6731.653	9	747.961	8.455	.000	.254
Error	19814.771	224	88.459			
Total	2766055.517	240				
Corrected Total	152946.680	239				

R Squared = .870 (Adjusted R Squared = .862)

Appendix C.6

ANOVA for the Genotypic Diversity in the Anchorage Protection for the Red Team (Section 7.3.3.1)

Dependent Variable: Genotypic Diversity for the Red Team

Source	Type III Sum of Squares	Df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	212919.387[a]	15	14194.626	64.167	.000	.811
Intercept	4914668.248	1	4914668.248	22216.741	.000	.990
Mutation	89549.286	3	29849.762	134.936	.000	.644
Algorithm	93027.753	3	31009.251	140.177	.000	.652
Mutation * Algorithm	30342.349	9	3371.372	15.240	.000	.380
Error	49552.078	224	221.215			
Total	5177139.714	240				
Corrected Total	262471.466	239				

R Squared = .811 (Adjusted R Squared = .799)

Appendix C.7

ANOVA for the Phenotypic Diversity in the Anchorage Protection for the Blue Team (Section 7.3.3.2)

Dependent Variable: Phenotypic Diversity for the Blue Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	.209[a]	15	.014	3.882	.000	.206
Intercept	898.319	1	898.319	250319.925	.000	.999
Mutation	.145	3	.048	13.428	.000	.152
Algorithm	.033	3	.011	3.051	.029	.039
Mutation * Algorithm	.032	9	.004	.977	.460	.038
Error	.804	224	.004			
Total	899.332	240				
Corrected Total	1.013	239				

R Squared = .206 (Adjusted R Squared = .153)

Appendix C.8

ANOVA for the Phenotypic Diversity in the Anchorage Protection for the Red Team (Section 7.3.3.2)

Dependent Variable: Phenotypic Diversity for the Red Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	.030[a]	15	.002	.815	.660	.052
Intercept	899.042	1	899.042	360809.180	.000	.999
Mutation	.010	3	.003	1.353	.258	.018
Algorithm	.014	3	.005	1.818	.145	.024
Mutation * Algorithm	.007	9	.001	.302	.974	.012
Error	.558	224	.002			
Total	899.631	240				
Corrected Total	.589	239				

R Squared = .052 (Adjusted R Squared = -.012)

Appendix D

Chapter 7 Coastline Protection Scenario Related Appendices

Appendix D.1

ANOVA for the Estimated Best GPs of the Blue Team in the Coastline Protection (Section 7.4.1.1)

Dependent Variable: Estimated Best GP of the Blue Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	12.446[a]	15	.830	3.404	.000	.186
Intercept	25068.384	1	25068.384	102844.795	.000	.998
Mutation	.622	3	.207	.851	.467	.011
Algorithm	11.747	3	3.916	16.064	.000	.177
Mutation * Algorithm	.077	9	.009	.035	1.000	.001
Error	54.600	224	.244			
Total	25135.431	240				
Corrected Total	67.046	239				

R Squared = .186 (Adjusted R Squared = .131)

Appendix D.2

ANOVA for the Estimated Best GPs of the Red Team in the Coastline Protection (Section 7.4.1.1)

Dependent Variable: Estimated Best GP of the Red Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	13.834[a]	15	.922	9.289	.000	.383
Intercept	31533.584	1	31533.584	317593.075	.000	.999
Mutation	1.921	3	.640	6.448	.000	.079
Algorithm	11.358	3	3.786	38.132	.000	.338
Mutation * Algorithm	.555	9	.062	.622	.778	.024
Error	22.241	224	.099			
Total	31569.659	240				
Corrected Total	36.075	239				

R Squared = .383 (Adjusted R Squared = .342)

Appendix D.3

ANOVA for the Estimated Average GPs of the Blue Team in the Coastline Protection (Section 7.4.1.2)

Dependent Variable: Estimated Average GP of the Blue Team

Source	Type III Sum of Squares	Df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	34.624[a]	15	2.308	5.489	.000	.269
Intercept	20266.109	1	20266.109	48196.751	.000	.995
Mutation	5.139	3	1.713	4.074	.008	.052
Algorithm	24.713	3	8.238	19.591	.000	.208
Mutation * Algorithm	4.772	9	.530	1.261	.259	.048
Error	94.189	224	.420			
Total	20394.922	240				
Corrected Total	128.813	239				

R Squared = .269 (Adjusted R Squared = .220)

Appendix D.4

ANOVA for the Estimated Average GPs of the Red Team in the Coastline Protection (Section 7.4.1.2)

Dependent Variable: Estimated Average GP of the Red Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	45.249[a]	15	3.017	15.617	.000	.511
Intercept	28500.550	1	28500.550	147549.314	.000	.998
Mutation	2.951	3	.984	5.093	.002	.064
Algorithm	39.484	3	13.161	68.138	.000	.477
Mutation * Algorithm	2.813	9	.313	1.618	.111	.061
Error	43.268	224	.193			
Total	28589.066	240				
Corrected Total	88.516	239				

R Squared = .511 (Adjusted R Squared = .478)

Appendix D.5

ANOVA for the Genotypic Diversity of the Blue Team in the Coastline Protection (Section 7.4.3)

Dependent Variable: Genotypic Diversity of the Blue Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	88218.501[a]	15	5881.233	63.371	.000	.809
Intercept	1562821.350	1	1562821.350	16839.491	.000	.987
Mutation	36646.005	3	12215.335	131.621	.000	.638
Algorithm	46640.497	3	15546.832	167.518	.000	.692
Mutation * Algorithm	4931.999	9	548.000	5.905	.000	.192
Error	20788.751	224	92.807			
Total	1671828.601	240				
Corrected Total	109007.252	239				

R Squared = .809 (Adjusted R Squared = .797)

Appendix D.6

ANOVA for the Genotypic Diversity of the Red Team in the Coastline Protection (Section 7.4.3)

Dependent Variable: Genotypic Diversity of the Red Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	68628.036[a]	15	4575.202	71.122	.000	.826
Intercept	1373228.330	1	1373228.330	21347.103	.000	.990
Mutation	24989.949	3	8329.983	129.491	.000	.634
Algorithm	39524.196	3	13174.732	204.804	.000	.733
Mutation * Algorithm	4113.891	9	457.099	7.106	.000	.222
Error	14409.597	224	64.329			
Total	1456265.963	240				
Corrected Total	83037.633	239				

R Squared = .826 (Adjusted R Squared = .815)

Appendix D.7

ANOVA for the Phenotypic Diversity of the Blue Team in the Coastline Protection (Section 7.4.3)

Dependent Variable: Phenotypic Diversity of the Blue Team

Source	Type III Sum of Squares	Df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	.931[a]	15	.062	8.598	.000	.365
Intercept	838.537	1	838.537	116135.179	.000	.998
Mutation	.658	3	.219	30.359	.000	.289
Algorithm	.152	3	.051	7.030	.000	.086
Mutation * Algorithm	.121	9	.013	1.868	.058	.070
Error	1.617	224	.007			
Total	841.086	240				
Corrected Total	2.549	239				

R Squared = .365 (Adjusted R Squared = .323)

Appendix D.8

ANOVA for the Phenotypic Diversity of the Red Team in the Coastline Protection (Section 7.4.3)

Dependent Variable: Phenotypic Diversity of the Red Team

Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	3.031[a]	15	.202	8.578	.000	.365
Intercept	620.073	1	620.073	26325.179	.000	.992
Mutation	.552	3	.184	7.809	.000	.095
Algorithm	2.164	3	.721	30.622	.000	.291
Mutation * Algorithm	.315	9	.035	1.487	.154	.056
Error	5.276	224	.024			
Total	628.379	240				
Corrected Total	8.307	239				

R Squared = .365 (Adjusted R Squared = .322)

Appendix E

Chapter 7 Genomes of the Last Populations

Appendix E.1

Last Population of the Coastline Protection Scenario in CEAN (Section 7.4.2)

Blue														
Home_x1	Home_x2	Home_x3	WayPx11	WayPx12	WayPx13	WayPx21	WayPx22	WayPx23	Alive Enemy	Alive Friends	En Threat 1	Next WayPoint	Alt WayPoint	Movement Speed
87.93	185.97	158.84	191.98	85.58	47.28	176.06	0	175	100	-4.53	100	-45.11	95.89	100
50.93	191.8	158.84	199	66.61	63.64	175.06	0	178.96	100	-4.53	100	-45.11	100	100
0	194.32	175.81	199	66.61	107.32	180.58	0	199	100	7.93	100	-39.84	100	100
20.78	185.52	105.3	199	54.35	199	189.39	68.57	95.38	1.2	-27.53	96.31	3	100	97.05
0	144.96	0	169.79	13.19	38.69	169.14	49.23	49.93	-20.92	-22.97	100	37.54	49.91	100
0.68	183.87	175.81	199	66.61	107.32	176.8	0	199	100	14.65	100	-39.84	100	100
4.52	199	164.2	199	85.33	97.48	165.21	0	138.46	1.2	2.19	100	3	100	100
0	199	180.77	174.03	52.3	129.79	174.85	0	199	45.9	-32.75	91.96	8.29	94.2	100
0	199	108.21	199	126.43	184.88	151.5	16	138.46	1.2	-24.85	100	-39.84	100	99.61
63.2	131.18	92.65	67.41	11.39	0	139.8	52.88	176.61	53.73	-7.47	64.09	3	82.32	100
104.91	130.36	107.72	151.45	41.89	18.44	199	10.76	125.37	86.89	17.66	36.25	-10.28	100	93.25
0	194.32	175.81	160.78	66.61	93.12	182.31	0	157.05	100	-30.73	100	3	100	92.66
2.16	148.55	181.31	120.64	63.57	67.7	199	7.26	199	97.44	-73.99	100	-22.24	71.85	85.35
51.87	77.66	9.88	151.02	34.37	29.7	199	28.36	140.47	0.47	-48.81	82.4	94.79	92.93	98.56
0	199	108.21	199	88.52	130.08	136.76	77	101.35	1.2	-47.54	73.6	25.53	100	89.41
Red														
Home_x1	Home_x2	Home_x3	WayPx11	WayPx12	WayPx13	Alive Enemy	Alive Friends	En Threat 1	Next WayPoint	Alt WayPoint	Movement Speed			
186.54	17.17	57.9	173.73	74.89	75.12	-23.44	-56.26	-100	26.18	-3.6	99.42			
198.84	6.08	57.9	199	74.89	83.2	-23.44	-56.26	-100	26.18	-3.6	100			
168.29	16.06	27.77	196.43	60.87	85.38	-41.09	-23.89	-82.84	30.56	-63.04	100			
145.79	6.08	15.19	132.4	60.87	110.22	-41.9	-91.86	-48.2	41.16	-87.3	100			
194.69	50.49	0	86.31	133.47	70	-62.18	-34.78	-49.87	65.23	-100	100			
199	67.55	113.5	63.13	73.9	104.59	-98.16	8.21	-43.73	42.54	44.55	98.66			
199	3.39	0.86	198.05	34.55	134.29	-30.16	-23.61	-82.84	54.6	-68.99	100			
179.41	6.08	82.5	199	36.42	151.5	-31.01	-73.7	0.05	65.79	-90.59	100			
199	102.5	17.91	100.77	70.21	104.26	-100	0.64	-79.32	32.37	-58.76	91.56			
176.93	140.93	0	77.5	87.5	74.18	-36.27	-91.86	-12.32	41.16	-97.43	100			
177.57	6.08	6.28	194.5	60.87	85.38	-41.9	19.75	-49.47	69.95	-87.3	100			
199	175.02	0	79.36	110.73	79.15	-34.68	9.13	-49.47	21.3	-100	100			
183.89	13.45	0	199	60.87	85.38	-71.72	-10.58	-77.87	18.88	-95.31	85.18			
198.84	6.08	26.25	156.62	38.35	191.04	-49.06	-51.95	2.55	4.37	90.3	100			
199	0	35.75	175.5	77.18	85.38	-41.09	-25.65	-100	10.81	-2.5	100			

Home_x1 = x coordinate of home for boat 1

Home_x2 = x coordinate of home for boat 2

Home_x3 = x coordinate of home for boat 3

WayPx11 = x coordinate of first waypoint for boat 1

WayPx12 = x coordinate of first waypoint for boat 2

WayPx13 = x coordinate of first waypoint for boat 3

WayPx21 = x coordinate of second waypoint for boat 1

WayPx22 = x coordinate of second waypoint for boat 2

WayPx23 = x coordinate of second waypoint for boat 3

Alive Enemy, Alive Friends, Enemy Threat, Next Waypoint, Alternate Waypoint, Movement Speed = as mentioned in Table 3.2

Appendix E.2

Last Population of the Coastline Protection Scenario in CEAFS (Section 7.4.2)

Blue														
Home_x1	Home_x2	Home_x3	WayPx11	WayPx12	WayPx13	WayPx21	WayPx22	WayPx23	Alive Enemy	Alive Friends	En Threat 1	Next WayPoint	Alt WayPoint	Movement Speed
71.51	145.44	18.62	199	129.29	77.47	0	31.24	99.97	100	-100	76.49	4.44	-99.93	98.59
125.74	177.3	2.19	199	160.36	98.16	0	0	0	42.67	-56.77	56.57	-3.53	-94.13	100
199	121.62	15.78	5.53	188.57	46.55	16.14	70.64	124.21	100	-95.02	100	-7.37	-99.93	98.59
90.57	193.71	47.02	0	188.57	46.55	16.14	50.12	114.01	100	-95.02	100	-7.37	-100	98.59
71.51	145.44	14.9	199	111.02	77.47	0	34.08	96.79	100	19.07	87.97	30.28	-29.26	93.69
99.45	184.44	86.16	136.55	199	0	21.43	30.61	0.26	82.72	-70.22	70.79	-0.63	-21.19	100
53.59	153.83	31.6	163.92	197.53	141.6	95.84	0	13.27	87.99	-63.45	95.39	33.15	-16.51	88.54
199	121.62	15.78	21.16	187.02	46.55	14.04	60.07	141.96	100	-100	100	-21.74	-29.26	100
92.79	145.64	47.53	144.77	173.87	142.18	77.51	4.68	0	81.28	-25.24	56.57	-15.06	-75.29	100
125.74	198.1	50.45	176.86	150.2	94.77	11.79	0	0	44.53	-78.34	39.31	36.72	-64.98	86.1
92.79	145.64	47.53	179.81	173.87	142.71	77.51	7.55	16.07	100	-25.24	24.8	-0.63	-75.29	94.32
199	121.62	15.78	27.65	139.09	86.59	86.46	94.96	1.33	64.67	-89.59	100	-24.74	-23.62	100
116.03	132.16	97.6	155.28	154.82	75.38	76.7	52.86	0	95.32	-33.37	30.35	-27.05	18.11	100
116.66	146.04	34.55	195.39	107.44	56.42	39.75	0	0	74.31	-100	75.74	-51.21	23.73	100
199	199	57.99	164.98	123.64	91.99	41.25	67.54	0	96.13	-92.58	83.58	-32.15	9.51	100
Red														
Home_x1	Home_x2	Home_x3	WayPx11	WayPx12	WayPx13	Alive Enemy	Alive Friends	En Threat 1	Next WayPoint	Alt WayPoint	Movement Speed			
27.99	113.13	71.76	73.56	175.11	47.65	-89.21	-46.36	-52.54	30.52	-60.46	100			
3.3	14.94	89.56	34.94	136.55	199	-74.48	-100	-100	70.07	-100	100			
11.2	158.09	131.68	0	95.92	44.06	-26.06	-100	-63.6	85.67	-88.39	100			
19.52	96.96	167.48	52.25	191.43	125.74	-100	-22	-72.76	42.43	-90.56	97.6			
27.99	113.13	48.93	170.73	176.39	47.65	-89.21	-46.36	-47.79	30.52	-60.46	100			
33.32	193.06	150.53	99.01	104.71	182.18	-7.15	-14.36	-73.53	46.03	-90.1	100			
0	125.84	97.36	97.19	101.28	161.4	-69.41	-100	-52.96	25.49	-85.4	100			
0	122.93	71.21	72.64	95.48	161.84	-79.38	-79.5	-62.01	30.52	-32.94	100			
4.12	60.62	112.99	0	158.52	173.23	17.51	-82.32	-86.71	51.48	-100	87.84			
30.77	167.54	162.42	64.58	84.84	169.05	-35.7	-63.57	-100	74.9	-41.56	100			
0	20.18	85.41	105.06	177.44	199	-57.33	-86.68	-55.47	50.5	-83.63	95.59			
60.56	53.63	158.62	0	153.83	91.64	-18.04	-98.59	-63.54	73.76	-65.91	100			
12.85	38.74	115.57	28.41	163.12	184.84	-100	-45.26	-96.05	43.71	-99.85	83.75			
40.8	129.49	52.75	65.05	159.99	196.9	6.1	-81.99	-86.71	51.48	-89.2	87.84			
17.5	159.21	162.42	90.19	103.46	131.25	-60.69	-81.54	-100	100	-22.52	95.94			

Appendix E.3

Last Population of the Coastline Protection Scenario in CEAHOF (Section 7.4.2)

Blue														
Home_x1	Home_x2	Home_x3	WayPx11	WayPx12	WayPx13	WayPx21	WayPx22	WayPx23	Alive Enemy	Alive Friends	En Threat 1	Next WayPoint	Alt WayPoint	Movement Speed
151.91	150.22	61.83	86.44	199	120.46	100.32	0	109.83	63.06	-40.07	67.86	-10.78	-57.49	100
172.13	151.32	61.83	156.36	199.99	120.46	100.32	5.56	124.02	63.06	-40.07	67.86	-10.78	-57.49	100
151.05	150.22	60.34	86.44	186.99	120.94	115.46	0	109.83	80.47	-40.14	67.86	-15.99	-15.96	100
183.12	86.66	66.18	77.03	181.54	100.7	84.42	0	67.49	63.06	23.32	100	-4.27	-72.22	100
138.19	150.22	61.83	25.02	184.22	143.83	80.42	0	109.83	63.06	-40.07	67.86	63.48	-57.49	100
199	86.33	0	89.91	140.05	98.91	27.7	0	75.34	60.43	-99.97	100	-51.66	-88.09	97.32
133.78	195.38	42.91	84.38	199	99.74	130.08	59.57	199	63.06	-63.01	64.34	-43.45	-59.46	100
151.91	154.94	61.83	86.44	199	120.46	91.44	0	109.83	63.06	-40.14	67.86	39.76	-13.44	95.15
151.31	148.45	90.13	65.53	181.54	120.27	149.23	0	38.94	95.27	52.89	100	-56.3	-92.79	100
167.06	140.56	60.66	22.26	186.99	141.14	171.92	0	53.74	-38.19	13.43	85.45	-15.44	-91.93	97.43
199	182.43	0	166.69	137.57	111.81	171.77	0	192.54	79.25	-51.31	79.32	-63.08	-80.52	99.42
125.87	187.31	56.9	16.97	0	135.76	77.28	26.06	95.64	-100	-29.09	100	9.92	29.59	100
199	86.33	0.66	89.91	146.32	98.91	54.94	43.16	57.58	82.9	-84.6	100	-38.22	-88.09	70.87
136.89	122.32	0.12	179.9	3.02	152.42	76.66	27.24	99.7	-44.19	-30.09	51.59	1.14	-1.53	99.3
129.78	161.51	67.08	16.97	19.37	145.96	78.86	108.48	22.18	-47.21	11.7	100	-29.95	23.04	94.03
Red														
Home_x1	Home_x2	Home_x3	WayPx11	WayPx12	WayPx13	Alive Enemy	Alive Friends	En Threat 1	Next WayPoint	Alt WayPoint	Movement Speed			
79.18	161.72	68.95	79.63	164.35	55.19	-54.92	-14.52	-100	31.21	73.86	100			
79.18	199	105.85	94.63	180.38	87.12	-54.92	-14.93	-100	31.21	41.31	100			
18.26	197	81.94	65.93	199	26.39	-84.47	-29.28	-60.43	24.44	100	96.74			
97.9	199	89.78	121.29	183.29	55.98	-98.28	-11.51	-100	24.44	96.37	100			
74.67	160.03	61.74	121.29	199	0	-54.92	7.74	-98.07	31.21	60.26	93.26			
130.33	160.54	131.83	93.61	164.5	64.85	-83.36	-5.66	-100	59.85	84.28	100			
94.02	199	68.95	68.63	199	0	-100	73.79	-92.57	34.18	55.1	100			
108.41	199	128.57	81.67	199.5	66.35	-83.39	4.46	-88.5	73.34	89.95	100			
174.53	194.32	38.14	74.77	199	61.62	-80.57	-71.48	-81.51	74.88	99.23	100			
84.96	189.56	0	100.22	157.79	192.55	-85.46	-62.81	-32.49	67.44	64.39	100			
148.55	170.4	0	96.76	177.37	199	-92.95	-85.28	-100	24.44	96.37	100			
119.72	199	56.59	181.81	199	0	-100	-68.44	-84.18	43.95	53.7	91.81			
69.43	199	80.62	93.99	199	0	-100	30.1	-81.19	66.06	50.75	100			
97.9	199	89.78	121.29	176.6	55.98	-58.79	-43.86	60.07	25.11	86.4	100			
147.02	199	65.84	37.09	199	70.5	-63.14	-65.05	-100	90.11	33.03	85.77			

Appendix E.4

Last Population of the Coastline Protection Scenario in CEACFH (Section 7.4.2)

Blue														
Home_x1	Home_x2	Home_x3	WayPx11	WayPx12	WayPx13	WayPx21	WayPx22	WayPx23	Alive Enemy	Alive Friends	En Threat 1	Next WayPoint	Alt WayPoint	Movement Speed
148.94	57.4	74.78	38.82	120.01	0	158.95	193.5	107.1	100	-24.72	99.19	42.71	43.79	96.82
0	123.65	163.47	63.54	102.14	42.07	66.8	157.66	150.01	25.41	-24.55	100	0.55	-82.75	90
195.06	64.98	162.54	12.09	103.74	90.34	134.77	199	199	30.97	-55.61	80.77	-22.54	-94	89.72
12.46	41.17	122.92	133.27	0	198.89	98.43	71.41	132.75	100	-10.73	26.98	42.42	-26.28	100
199	80.68	199	31.34	183.98	144.7	101.23	193.81	182.19	42.95	-9.75	94.2	57.9	-58.54	96.11
59.26	0	114.84	199	68.82	193.1	92.93	56.29	110.03	-28.34	-28.84	100	-13.13	-67.83	93.41
0	115.1	167.47	116.61	47.21	34.87	110.88	191.77	148.76	27.1	29.74	93	-24.99	-67.83	100
177.65	78.39	75.49	81.5	47.77	43.34	134.77	140.64	199	96.16	-32.76	100	-21.56	61.81	100
6.63	84.73	163.47	83.39	90.38	82.64	93.17	199	143.07	70.35	-6.26	100	-3.87	-100	59.1
0	112.99	163.47	106.13	27.87	75.44	136.61	160.49	103.39	100	57.71	96	-74.6	-52.27	100
199	27.22	48.93	161.15	51.85	151.16	172.47	167.74	164.68	100	-22.6	62.72	94.77	-29.97	78.06
42.31	9.05	119.83	68	86.27	2	179.35	199	189.75	27.14	54.12	35.7	12.95	-25.14	85.28
199	66.32	198.07	31.34	183.98	149.63	128.21	191.54	199	75.64	-29.3	100	-18.62	61.81	100
0	31.87	199	121.64	117.47	84.87	184.18	141.48	190.39	-67.95	92.35	74.88	-8.18	-13.14	100
30.8	33.74	114.84	199	68.82	198.65	92.93	56.29	116.1	-45.82	-28.84	100	-13.13	-67.83	72.54
Red														
Home_x1	Home_x2	Home_x3	WayPx11	WayPx12	WayPx13	Alive Enemy	Alive Friends	En Threat 1	Next WayPoint	Alt WayPoint	Movement Speed			
130.64	63.96	110.29	189.36	82.32	52.24	-61.4	11.23	-79.04	92.05	59.43	100			
134.74	144.08	125.57	158.78	199	16.02	-100	7.56	-96.66	100	-82.48	100			
128.97	0	127.73	102.06	125.78	77.47	-92.54	-31.71	-72.8	61.21	-82.22	98.61			
140.07	0	199	199	43.93	91.92	-91.57	45.01	-94.26	91.05	-47.31	96.1			
146.3	48.41	31.35	166.2	53.84	175.33	-40.99	-91.17	-79.04	56.05	12.25	100			
125.32	0	69.94	169.38	42.65	174.91	-25.51	-88.27	-24.76	100	49.54	94.25			
106.44	7	199	138.61	41.93	138.31	-100	4.03	-85.95	52.2	-82.48	100			
199	0	113.34	186.61	104.27	94.33	-28.44	100	-58.67	51.29	-59.37	96.49			
70.54	53.39	112.14	165.39	57.59	128.19	-4.13	18.05	-16.19	76.74	-20.1	86.09			
189.66	5.2	199	131.78	182.25	194.22	-74.81	-24.72	-96.89	88.73	-14.76	79.58			
199	0	158.87	185.28	104.27	92.11	-44.56	100	-65.11	17.12	-48.58	100			
152.09	0	199	140	85.8	106.21	-86.04	-100	-87.8	61.29	-81.54	99.63			
157.24	0	69.94	169.38	59.78	165.64	-58.26	-88.27	-56.9	19.95	-67.31	72.74			
199	53.59	24.15	131.73	83.98	1.62	-4.57	14.48	-83.34	-0.7	-100	76.36			
122.8	15.96	199	138.61	41.93	132.16	-100	4.03	-100	7.06	-82.92	100			

Appendix E.5

Last Population of the Anchorage Protection Scenario in CEAN (Section 7.3.2)

Blue																													
Home_ x1	Home_ x2	Home_ x3	Home_ y1	Home_ y2	Home_ y3	WayPx 11	WayPx 12	WayPx 13	WayPx 21	WayPx 22	WayPx 23	WayPy 11	WayPy 12	WayPy 13	WayPy 21	WayPy 22	WayPy 23												
202.08	180.43	243.15	54.62	130.72	60.88	108.3	256.64	105.93	144.45	264.32	246.49	154.6	97.94	96.83	88.58	84.09	117.13												
202.08	180.43	243.49	63.18	152.21	72.3	108.3	254.49	105.93	144.45	329	297.6	153.2	105.73	128.74	99.52	82.59	93.64												
203.15	181.85	242	62.31	152.21	66.16	110.36	292.64	100.44	144.45	329	238.18	159	121.73	72.64	65.55	88.61	95.8												
221.34	209.25	205.48	58.76	156.02	53.43	188.52	219.3	70	159.66	248.67	220.94	146.97	90.05	93.6	117.55	40	111.96												
192.64	245.22	243.15	48.31	131.46	59.97	70	117.24	70	104.81	329	236.4	130.55	103.03	40	114.85	73.5	123.15												
198.57	264.59	301.99	78.29	127.96	40	134.34	223.97	89.07	116.2	311.41	217.76	152.77	124.41	110.82	109.13	90.4	92.45												
151.27	198.84	160.3	53.32	128.47	51.13	108.3	241.63	105.93	144.45	264.32	278.75	154.6	97.94	102.7	86.42	84.09	130.4												
179.08	231.79	226.2	52.81	154.94	53.13	174.77	308.47	89.59	98.91	329	264.64	135.52	133.34	96.83	120.39	86.23	60.07												
178.92	286.94	192.18	52.81	146.51	45.68	165.62	251.34	85.79	91.76	277.98	166.73	159	67.92	53.2	80.13	56.64	137.64												
166.13	202.75	130.42	75.9	157.4	57.5	112.4	294.81	76.9	86.37	216.06	211.23	143.62	149.23	84.19	90.75	111.28	135.76												
164.98	146.72	243.15	55.72	155.67	55.97	87.66	260.72	70	70	276.91	275.83	152.89	99.18	93.81	108.25	58.5	89.84												
151.27	224.09	181.58	60.5	150.74	49.63	158.05	171.82	109.07	129.7	228.46	220.94	149.6	119.59	105.9	109.13	67.14	81.51												
198.83	135.62	234.82	40	147.19	40	86.37	154.77	70	139.63	241.49	293.25	159	106.83	110.89	114.5	64.91	133.13												
147.74	239.65	329	47.06	149.71	40	168.55	300.68	70	182.52	300.7	208.42	137.13	40.4	71	92.43	81.69	100.14												
153.64	217.89	86.75	69.24	155.79	41.95	119.95	328.4	165.61	98.73	323.89	239.33	155.72	81.83	81.64	142.76	83.66	105.44												
Red																													
Home_ x1	Home_ x2	Home_ x3	Home_ x4	Home_ x5	Home_ y1	Home_ y2	Home_ y3	Home_ y4	Home_ y5	WayPx 11	WayPx 12	WayPx 13	WayPx 14	WayPx 15	WayPx 21	WayPx 22	WayPx 23	WayPx 24	WayPx 25	WayPy 11	WayPy 12	WayPy 13	WayPy 14	WayPy 15	WayPy 21	WayPy 22	WayPy 23	WayPy 24	WayPy 25
358.26	95.09	357.29	270.07	359.46	23.68	34.31	33.07	161.24	195.34	212.85	146.5	399	154.89	127.37	164.94	11.14	193.51	376.97	139	139.85	110.64	58.39	99.17	139.34	119	83.47	102.8	159	84.64
393.79	95.82	399	259.37	359.46	19.61	33.43	31.88	170.86	192.71	205.32	146.5	399	153.87	70.95	172.17	47.6	193.51	399	141.33	130.78	112.63	58.39	99.13	140.12	111.43	66.04	107.33	159	79
358.26	95.09	399	298.13	375.74	28.47	31.58	31.88	161.24	199	205.32	154.23	399	153.87	127.37	164.94	7.67	193.6	376.97	139	139.85	110.64	58.39	105.85	129.79	119	86.7	102.07	159	80.64
358.26	95.09	339.75	293.26	277.05	23.68	33.74	39	161.24	195.34	212.85	146.5	399	154.89	39.82	164.94	11.14	193.51	376.97	139	136.18	110.64	59.6	98.08	139.34	118.13	83.47	102.8	159	84.64
398.33	0	379.78	33.89	399	1.37	37.99	1.04	175.93	190.94	329.1	208.74	384.18	162.23	124.54	162.16	317.17	139	117.28	139	159	105.88	90.52	86.48	141.16	106.88	110.83	89.88	45.44	95.12
399	95.09	346.42	315.77	347.75	29.83	36.34	31.88	161.24	195.34	205.32	146.5	399	180.02	198.32	181.06	11.14	193.51	376.97	139	139.85	110.64	58.39	105.95	139.34	119	99.97	105.02	155.43	81.34
360.41	0	306.98	245.99	354	23.68	32.84	30.55	164.93	198.54	163.02	154.19	399	149.66	103.21	157.55	4.12	186.36	350.94	139	158.82	100.69	50.41	104.14	135.26	115.48	95.49	99.28	159	92.96
399	0	244.59	371.72	341.69	33.84	33.63	30.11	166.3	199	236	178.19	148.57	180.5	91.68	160.51	0	211.4	399	139	147.13	113.09	58.39	105.35	149.79	119	84.02	102.73	147.04	91.15
366.44	44.51	309.23	293.55	259.67	31.15	36.87	30.11	182.48	199	236	184.24	171.64	168.08	63.37	157.33	0	200.35	399	139	140.66	100.13	42.46	105.71	140.84	119	88.13	104.23	154.61	91.87
346.88	126.99	226.39	266.92	307.26	18.97	39	37.69	166.54	189.57	184.9	142.93	274.43	145.91	304.27	166.47	10.03	177.68	399	147.79	106.66	95.45	111.3	105.09	131.46	97.4	53.14	96.57	156.43	89.71
305.93	13.29	399	263.9	301.41	33.11	33.56	37.76	160	195.81	77.56	145.89	184.37	259	186.62	155.8	131.26	188.78	387.54	139	123.89	115.41	71.01	116.54	149.3	108.09	101.31	106.94	159	94.81
367.27	102.14	399	288.04	393.39	19.61	39	28.11	168.33	191.02	294.28	180.49	123	209.36	231.79	141.18	129.53	187.2	296.37	210	78.57	119	117.11	110.9	94.68	88.99	59.57	101.89	138.88	79.21
243.8	8.04	370.46	260.92	399	19.81	39	21.51	175.78	194.17	266.16	139	399	162.7	346.08	161.79	153.46	199.59	72.87	142.7	72.12	84.61	53.95	115.07	145.23	102.21	90.92	83.12	156.74	82.23
332.7	198.07	269.79	216.4	258.46	15.72	39	38.57	164.26	189.3	239.18	139	316.95	149.49	180.35	152.48	101.55	159.07	314.86	139	101.88	92.23	70.68	105.02	118.77	112.27	63.45	99.57	135.7	88.32
338.26	0	399	69.63	207.14	8.47	23.57	28.72	165.92	199	175.11	144.66	399	146.35	50.78	219.23	82.41	139	237.73	203.21	112.21	93.82	80.89	112.72	128.83	116.22	40	102.49	159	88.09

Home_x1 = x coordinate of home for boat 1

Home_x2 = x coordinate of home for boat 2

Home_x3 = x coordinate of home for boat 3

Home_x4 = x coordinate of home for boat 4

Home_x5 = x coordinate of home for boat 5

Home_y1 = y coordinate of home for boat 1

Home_y2 = y coordinate of home for boat 2

Home_y3 = y coordinate of home for boat 3

Home_y4 = y coordinate of home for boat 4

Home_y5 = y coordinate of home for boat 5

WayPx11 = x coordinate of first waypoint for boat 1

WayPx12 = x coordinate of first waypoint for boat 2

WayPx13 = x coordinate of first waypoint for boat 3

WayPx14 = x coordinate of first waypoint for boat 4

WayPx15 = x coordinate of first waypoint for boat 5

WayPx21 = x coordinate of second waypoint for boat 1

WayPx22 = x coordinate of second waypoint for boat 2

WayPx23 = x coordinate of second waypoint for boat 3

WayPx24 = x coordinate of second waypoint for boat 4

WayPx25 = x coordinate of second waypoint for boat 5

WayPy11 = y coordinate of first waypoint for boat 1

WayPy12 = y coordinate of first waypoint for boat 2

WayPy13 = y coordinate of first waypoint for boat 3

WayPy14 = y coordinate of first waypoint for boat 4

WayPy15 = y coordinate of first waypoint for boat 5

WayPy21 = y coordinate of second waypoint for boat 1

WayPy22 = y coordinate of second waypoint for boat 2

WayPy23 = y coordinate of second waypoint for boat 3

WayPy24 = y coordinate of second waypoint for boat 4

WayPy25 = y coordinate of second waypoint for boat 5

Appendix E.6

Last Population of the Anchorage Protection Scenario in CEAFS (Section 7.3.2)

Blue																													
Home_ x1	Home_ x2	Home_ x3	Home_ y1	Home_ y2	Home_ y3	WayPx 11	WayPx 12	WayPx 13	WayPx 21	WayPx 22	WayPx 23	WayPy 11	WayPy 12	WayPy 13	WayPy 21	WayPy 22	WayPy 23												
302.3	184.14	189.5	82.18	40	128	213.75	87.06	250.92	214.78	305.37	159.54	58.39	113.51	110.31	120.47	60.2	76.67												
277.51	277.2	203.91	40	40	146.28	135.84	96.45	177.76	257.4	182.58	178.16	45.69	142.54	87.97	101.72	51.32	71.31												
300.16	241.58	251.46	83.46	70.73	113.15	168.44	70	206.69	169.02	174.57	105.27	42.59	135.62	118.15	87.74	96.34	129.42												
275.24	116.56	113.28	84.18	63.57	159	171.83	110.71	250.92	234.99	286.37	221.78	53.84	107.94	98.51	104.64	60.2	80.58												
275.24	91.89	113.51	92.21	40	155.8	222.26	89.85	250.92	214.78	329	233.44	58.39	106.42	106.74	104.02	87.69	71.31												
300.01	164.98	173.33	116.65	43.11	139.18	204.39	105.14	192.87	245.06	166.34	176.07	45.69	142.54	87.97	101.72	40	71.31												
255.57	146.31	151.31	80.89	40	139.5	227.88	70	239.36	232.32	248.61	229.29	78.92	131.35	101.71	127.37	67.89	75.88												
275.24	70	97.88	105.95	40	159	202.96	121.32	225.04	225.08	255.23	191.42	73.17	122.12	100.7	104.32	43.55	71.31												
329	144.15	203.58	101.72	50.87	159	230.16	84.34	232.27	194.3	148.83	186.19	48.67	96.76	159	129.81	108.37	63.97												
259.52	132.09	135.52	59.64	43.58	142.34	193.82	91.47	209.77	214.78	216.25	327.12	96.8	113.92	157.84	43.15	95.73	100.7												
302.3	122.72	224.28	82.18	51.96	153.47	198.69	114.69	225.04	193.6	227.9	172.64	117.61	133.41	123.64	115.81	41.16	80.26												
270.77	164.98	224.04	128.94	43.11	139.18	216.91	78.24	203.84	132.47	262.41	215.6	66.18	122.12	98.72	97.76	42.43	73.87												
307.41	101.67	224.28	82.18	60.65	159	155.77	114.69	226.36	193.6	234.09	181.53	119.38	115.98	128.94	130.87	66.71	43.23												
242.35	160.41	255.87	103.44	63.28	136.34	138.76	129.84	261.2	276.14	149.89	101.27	69.21	106.12	132.89	127.36	53.93	83.85												
329	193.31	220.17	97.65	97.05	159	131.48	74.73	215.22	181.26	156.58	323.98	88.91	116.89	154.48	41.56	85.53	69.35												
Red																													
Home_ x1	Home_ x2	Home_ x3	Home_ x4	Home_ x5	Home_ y1	Home_ y2	Home_ y3	Home_ y4	Home_ y5	WayPx 11	WayPx 12	WayPx 13	WayPx 14	WayPx 15	WayPx 21	WayPx 22	WayPx 23	WayPx 24	WayPx 25	WayPy 11	WayPy 12	WayPy 13	WayPy 14	WayPy 15	WayPy 21	WayPy 22	WayPy 23	WayPy 24	WayPy 25
17.74	287.35	41.42	399	0	28.11	7.17	0	160	165.81	147.59	247.35	13.94	239.8	399	139	81.27	220.13	237.48	240.71	153.59	104.87	41.97	95.5	65.58	119	126.07	99.85	82.01	79
147.28	361.27	0	352.22	67.95	29.54	12.92	0	168.71	164.78	189.73	255.42	0	219.61	288.34	139	41.84	194.79	240.78	258.61	139.36	106.64	87.31	93.04	74.93	114.07	112.87	105.32	56.57	82.48
0	399	165.13	399	88.34	0	37.48	3.26	166.53	171.42	17.95	224.98	51.29	248.34	382.41	141.64	0	193.24	0	224.66	159	102.77	103.02	101.98	104.93	116.44	125.95	113.09	41.27	81.61
77.31	384.93	247.82	237.34	67.95	21.3	17.03	3.25	166.84	164.3	139.7	249.9	0	239.71	399	162.63	6.23	195.45	204.99	214.82	130.31	94.74	69.45	95.5	77.17	116.48	127.33	105.43	40	88.48
0	399	102.49	399	135.31	24.27	16.52	3.69	179.99	183.06	83.61	259	0	216.31	234.85	139	50.52	217.69	114.44	235.66	159	99.83	76.27	91.57	104.93	119	111.3	111.49	72.24	85.59
131.7	351.89	216.03	399	88.34	32.11	16.52	8.04	179.99	183.06	83.61	259	0	218.78	234.85	139	50.52	217.69	114.44	225.32	159	99.83	76.27	91.57	93.82	118.96	111.3	104.66	72.24	86.45
21.67	308.81	73.16	297.94	108.91	33.68	17.76	8.96	182.68	180.71	17.94	258.49	21.35	231.98	399	202.55	56.27	142.18	238.29	259	111.32	100.23	95.32	93.04	55.45	110.18	102.48	95.63	72.88	81.7
10.44	399	278.72	399	112.93	32.11	17.85	0.47	164.58	160	331.82	246.19	113.39	216.05	328.67	139	36.29	193.18	102.34	239.01	159	98.68	85.83	86.4	104.93	119	113.73	111.49	82.59	92.07
206.3	324.07	33.03	193.68	65.7	0	23.45	24.3	168.02	197.44	160.68	240.38	0	257.54	282.31	139	0	184.59	209.37	259	159	96.03	64.87	97.94	101.72	117.2	114.09	96.05	81.63	90.99
0	399	102.49	399	135.31	24.27	18.99	4.96	172.73	160	227.9	246.19	84.3	221.68	362.77	158.64	13.84	213.35	176.37	242.79	155.04	99.35	76.27	86.45	104.93	119	113.73	119	73.09	88.27
179.98	215.88	258.2	399	283.99	4.36	31.38	7.96	172.14	172.98	214.03	245.61	108.43	252.59	399	141.04	36.29	197.5	5.15	259	159	105.95	72.88	93.04	103.53	119	113.73	111.49	75.98	96.69
0	266.72	241.39	293.68	88.34	32.11	16.25	18.67	172.14	160	196.22	246.19	84.3	200.52	328.67	141.04	32.84	242.47	114.44	242.79	159	93.3	66.85	91.57	104.93	115.64	100.15	98.47	40	80.7
28.88	379.93	228.6	399	87.85	32.11	21.06	10.33	160	160	214.03	246.19	5.79	224.16	348.09	139.13	60.56	197.64	121.39	238.44	159	89.86	88.61	82.29	104.93	110.45	123.65	110.72	86	86.96
107.75	337.23	214.24	369.72	75.9	28.13	22.31	0	166.38	160	364.82	259	55.89	217.67	318.71	141.04	49.39	201.7	118.72	247.65	144.18	100.55	62.14	98.16	41.22	113.5	101.4	100.72	40	80.89
272.62	294.63	0	319.65	125.3	30.66	15.31	9.14	168.43	190.41	0	240.78	0	259	336.97	147.07	15.3	212.17	63.6	222.06	158.37	108.54	62.19	83.46	126.23	115.29	100.13	80.74	43.55	88.71

Appendix E.7

Last Population of the Anchorage Protection Scenario in CEAHOF (Section 7.3.2)

Blue																													
Home_ x1	Home_ x2	Home_ x3	Home_ y1	Home_ y2	Home_ y3	WayPx 11	WayPx 12	WayPx 13	WayPx 21	WayPx 22	WayPx 23	WayPy 11	WayPy 12	WayPy 13	WayPy 21	WayPy 22	WayPy 23												
239.4	146.11	213.6	40	158.89	61.93	148.59	266.64	134.53	162.53	215.65	150.59	59.03	93.37	76.75	81.56	95.56	108.2												
239.4	146.11	241.36	40	158.89	61.93	148.59	266.64	161.41	162.53	231.41	188.17	63.49	84.79	73.16	93.52	95.56	101.07												
287.2	147.19	232.24	42.13	159	60.72	143.21	245.5	116.91	162.53	215.65	150.59	59.03	93.37	56.14	73.83	95.56	108.97												
287.66	138.42	213.6	40	151.32	40.4	162.09	266.64	134.53	162.53	215.65	150.59	59.03	94.87	76.75	71.48	95.56	116.32												
287.2	138.42	213.6	40	159	98.67	162.09	274.72	143.62	162.53	215.65	150.59	68.91	94.87	76.75	81.56	95.56	87.17												
294.5	74.13	245.21	40	153.96	93.83	88.56	276.46	210.94	162.53	295.73	102.18	64.13	88.47	82.51	61.37	68.28	140.28												
262.15	148.84	183.18	51.76	120.79	40	128.16	258.7	155.58	188.32	275.14	98.3	40	96.8	120.49	79.37	129.25	118.38												
216.14	84.85	176.93	47.01	159	46.26	175.89	268.84	221.54	144.45	265.44	169.54	91.75	119.36	91.61	53.49	89.22	118.09												
287.2	117.24	165.91	40	153.44	51.7	74.85	280.25	143.91	156.39	160.95	132.23	40	104.21	65.19	72.16	98.96	122.18												
201.24	216.87	180.68	56	150.17	40.85	70	243.15	124.77	108.89	175.32	124	49.65	82.98	45.78	60.86	109.36	117.31												
283.15	113.66	240.04	47.9	159	57.32	120.88	236.46	163.14	80.81	222.18	199.79	54.31	94.87	76.75	69.97	83.02	87.86												
254.28	132.91	190.52	50.22	138.53	53.64	104.57	259.66	70	206.62	278.92	73.96	40.71	133.02	94.73	53.71	109.63	68.24												
254.28	152.82	183.11	40	121.85	40.2	138.36	282.37	87.32	203.14	278.92	73.96	40	106.44	72.45	69.49	108.23	109.82												
301.43	70	164.51	44.96	159	43.41	70	249.25	180.66	163.58	151.94	112.82	64.03	128.31	79.54	66.32	159	82.22												
309.53	70	184.91	72.95	134.64	117.51	110.53	310.03	206.09	180.48	266.18	115.84	40	117.09	79.92	49.41	102.52	142.42												
Red																													
Home_ x1	Home_ x2	Home_ x3	Home_ x4	Home_ x5	Home_ y1	Home_ y2	Home_ y3	Home_ y4	Home_ y5	WayPx 11	WayPx 12	WayPx 13	WayPx 14	WayPx 15	WayPx 21	WayPx 22	WayPx 23	WayPx 24	WayPx 25	WayPy 11	WayPy 12	WayPy 13	WayPy 14	WayPy 15	WayPy 21	WayPy 22	WayPy 23	WayPy 24	WayPy 25
374.96	75.36	399	204.43	110.54	0	17.26	23.9	188.67	195.08	0	160.96	369.38	149.25	21.77	193.75	79.49	232.66	249.38	139	40	86.55	68.39	79	47.02	95.65	117.22	79	104.88	99.14
399	29.85	301.07	144.05	293.17	6.4	0	30.05	195.4	178.1	116.52	225.56	351.79	193.58	0	159.44	397.01	210.61	399	143.36	109.31	85.6	86.48	84.66	91.11	106.5	159	98.64	111.72	94.2
399	29.85	368.34	177.86	293.17	6.4	0	30.05	195.4	178.1	77.69	219.2	351.79	205.29	0	159.44	393.78	229.37	399	139	109.31	84.42	72.02	88.39	91.11	106.5	159	98.22	104.93	95.63
399	43.43	368.34	177.86	293.17	3.1	0	30.05	192.28	176.34	77.69	209.14	351.79	205.29	0	185.7	393.78	222.45	332.19	139	40	89.63	118.71	105.49	81.03	111.46	144.4	103.42	104.82	94.07
399	53.36	303.12	211.6	46.99	1.53	17.56	32.29	191.08	198.95	4.26	198.2	399	139	0	165.01	5.37	238.42	268.98	139	66.26	85.86	65.05	83.4	40	114.86	141.14	81.46	87.49	100.03
399	61.7	368.34	177.86	279.84	7.7	1.33	30.05	194.58	178.1	77.69	226.68	346.45	205.29	0	159.44	393.78	229.37	399	140.79	119.98	83.34	59.35	87.97	112.23	106.5	142.93	98	104.93	83.05
350.81	254.24	387.71	157.44	55.42	0	11.66	34.39	187.77	199	0	186.14	399	139	0	183.25	291.23	216.67	318.53	142.84	52.66	94.68	66.15	104.72	146.15	102.23	112.47	97.36	118.02	114.44
296.81	117.49	399	162.5	0	4.69	21.68	29.25	173.62	193.15	0	211.94	310.67	191.2	6.14	149.5	106.53	254.78	282.64	182.58	40	94.53	73.3	79.82	47.02	103.67	151.49	94.86	92.92	88.69
340.87	125.9	399	186.96	112.94	2.12	23.21	23.9	175.35	198.36	42.02	200.61	395.02	139	21.77	176.61	79.49	220.72	278.28	139	56.56	81.18	71.1	79.01	91.4	110.83	145.62	81.53	101.24	105.3
371.81	146.01	399	168.92	95.27	2.12	24.94	15.07	170.97	198.36	0	192.57	399	139	20.95	176.61	93.32	211.54	262.97	139	54.01	84.96	68.39	79	45.53	98.11	117.66	81.53	110.86	105.3
350.81	124.84	383.69	208.73	258.11	6.11	0.16	30.05	195.19	188.23	0	226.68	301.64	213.87	0	149.36	399	249.66	389.48	140.39	122.11	91.04	92.14	91.78	124.44	98.81	159	80.62	111.88	104.74
399	88.21	352.55	171.21	285.9	8.76	9.25	29.77	180.88	165.12	0	200.03	216.25	167.42	40.15	178.18	384.58	209.33	399	148.77	89.4	79	58.89	94.09	91.11	106.5	149.44	100.73	113.6	79.67
280.71	0	399	0	255.25	20.22	25.75	13.91	199	180.97	87.55	247.95	399	196.89	0	153.4	271.73	226.35	379.86	163.41	54.51	79.13	96.43	100.69	40	116.27	159	94.17	154.97	79
399	125.05	337.79	80.2	0	4.48	9.15	17.68	196.77	169.52	85.41	224.88	298.01	195.73	206.63	244.75	399	198.51	241.3	151.94	61.37	93.84	45.28	107.5	47.4	107.66	157.02	86.52	90.41	114.89
253.48	149.25	307.57	55.63	72.16	16.51	9.01	28.77	192.32	179.14	21.91	215.23	199.22	193.07	271.2	195.96	328.14	214.28	399	139	99.51	84.42	72.02	95.32	107.51	106.5	152.71	93.62	98.82	94.91

Appendix E.8

Last Population of the Anchorage Protection Scenario in CEACFH (Section 7.3.2)

Blue																													
Home_ x1	Home_ x2	Home_ x3	Home_ y1	Home_ y2	Home_ y3	WayPx 11	WayPx 12	WayPx 13	WayPx 21	WayPx 22	WayPx 23	WayPy 11	WayPy 12	WayPy 13	WayPy 21	WayPy 22	WayPy 23												
137.72	291.61	189.08	108.36	152.22	40	227.89	244.45	70	70.4	200.06	178.61	149.9	43.72	154.74	96.48	101.01	96.1												
161.99	281.72	241.58	115.75	136.66	40	264.3	143.92	122.96	75.17	232.86	214.3	130.61	76.09	142.02	92.43	99.59	127.69												
124.38	305.33	250.1	147.31	137.71	45.05	157.67	212.58	70	109.22	307.21	164.18	106.68	73.63	137.08	159	89.33	86.92												
260.16	228.74	329	151.29	110.37	40	220.98	159.36	70	116.99	235.36	264.39	159	44.26	159	126.55	127.93	66.66												
138.53	329	222.83	139.15	159	41.26	242.2	169.21	254.87	292.99	70	226.43	59.96	104.65	104.93	131.01	149.47	111.24												
120.21	287.61	197.8	108.36	157.18	124.63	113.93	161.4	190.79	241.08	248.61	146.4	132.36	93.03	139.51	76.77	94.44	127.69												
229.58	268.75	304.97	159	133.05	58.14	93.05	306.8	215.9	205.39	268.98	192.69	123.93	54.36	72.86	93.14	93.16	86.51												
138.53	265.97	250.85	139.15	159	41.26	242.2	196.76	260.02	292.99	70	226.43	59.96	104.65	108.14	131.01	130.02	111.24												
188.03	166.49	230.67	112.5	71.15	150.14	139.82	280.06	238.47	227.37	147.5	222.4	159	84.87	42.56	92.18	120.54	75.97												
107.51	265.97	260.12	137.07	134.77	41.26	233.6	196.76	262.66	262.46	70	226.43	91.22	104.65	108.14	131.01	130.02	104.96												
89.35	242.19	234.95	158.05	138.66	52.24	190.12	162.85	70	104.45	316.37	175.21	112.65	84.59	137.55	158.32	159	107.21												
124.38	293.78	281.2	158.1	138.66	40	176.69	201.2	70	110.14	329	220.14	113.44	81.25	145.48	130.43	98.21	40												
196.53	307.54	322.58	51.5	51.69	70.07	117.26	329	100.71	70	270.12	118.14	156.61	108.3	159	96.49	108.38	125.9												
187.19	329	236.1	134.49	113.36	70.94	187.31	264.69	70	77.19	201.76	78.94	156.88	64.66	44.04	50.54	95.59	96.23												
162.37	201.71	293.6	100.38	151.73	91.45	241.7	179.09	167.07	271.66	253.02	71.24	133.75	159	159	84.34	81.52	99.43												
Red																													
Home_ x1	Home_ x2	Home_ x3	Home_ x4	Home_ x5	Home_ y1	Home_ y2	Home_ y3	Home_ y4	Home_ y5	WayPx 11	WayPx 12	WayPx 13	WayPx 14	WayPx 15	WayPx 21	WayPx 22	WayPx 23	WayPx 24	WayPx 25	WayPy 11	WayPy 12	WayPy 13	WayPy 14	WayPy 15	WayPy 21	WayPy 22	WayPy 23	WayPy 24	WayPy 25
53.97	177.54	394.92	399	386.2	19.46	10.63	2.39	178.03	187.71	290.83	259	349.17	171.79	350.8	174.89	158.42	235.06	0	162.07	159	119	98.01	100.75	129.96	86.33	77.46	102.37	159	104.61
0	0	399	399	399	7.35	4.99	0	180.79	199	268.61	213.3	234.29	150.69	84.23	190.17	273.42	164.31	292.76	174.34	121.07	79.08	159	80.17	68.63	93.16	42.73	84.63	143.19	97.42
21.46	33.56	295.1	290.38	0	10.96	1.64	13.18	172.99	176.36	399	217.64	330.63	236.44	38.86	222.98	333.59	193.39	327.11	193.64	133.94	119	68.09	119	131.7	94.39	80.62	93.52	159	117.3
46.61	135.5	395.63	399	341.56	18.53	10.86	6.98	167.48	199	399	258.56	386.65	198.19	399	163.31	139.25	240.14	0	178.39	146.99	117.64	73.32	119	126.75	90.34	92.45	104.02	150.57	109.43
17.1	10.39	380.38	321.97	265.51	22.24	26.87	0	167.37	176.94	83.09	252.84	251.58	209.41	154.17	164.15	16.94	253.14	0	172.18	140.18	118.19	133.16	85.81	75.89	82.62	60.97	92.63	134.73	80.11
44.46	138.39	252.88	399	387.81	15.94	14.73	23.05	176.28	177.89	212.51	256.25	275.77	144.37	43.47	201.27	0	258.55	343.62	220.09	135.27	102.54	71.57	107.79	142.96	94.31	74.99	97.54	159	113.38
50.77	10.81	303.21	296.1	15.17	11.07	0	13.24	177.01	172.28	399	214.4	277.03	171.79	350.8	174.89	157.01	241.96	0	162.07	159	119	98.01	100.75	129.96	86.33	77.46	102.37	159	104.61
87.32	135.08	323.67	362.42	341.56	7.78	10.63	9.33	165.66	199	399	257.4	382.54	181.93	355.78	145.78	205.36	139	378.99	199.72	75.45	87.97	138.69	98.3	137.24	102.52	83.99	104.25	119.12	79.34
0	9.73	340.5	399	354.34	14.16	10.63	10.82	165.19	195.39	399	259	399	174.19	286.41	171	62.24	238.07	0	229.06	47.17	87.43	147.48	103.55	53.35	101.81	133.3	85.37	154.27	106.02
0	120.78	375.02	160.81	0	9.41	3.92	0.99	165.86	196.36	282.72	248.25	383.22	156.34	399	223.27	176.46	233.99	9.12	163.98	63.62	99.35	114.15	85.99	127.81	93.28	60.88	111.91	104.16	86.06
37.08	41.24	399	399	228.5	6.45	15.49	25.39	195.68	172.99	365.32	215.44	100.06	201.21	0	209.74	96.31	237.16	0	171.38	104.78	117.61	50.22	119	72.43	84.05	60.97	94.56	132.66	91.81
17.95	22.47	339.67	308.06	0	25.67	9.04	0.93	163.26	188.68	399	259	345.18	176.87	292.86	251.38	54.36	154.82	201.04	254.71	73.86	81.84	159	91.5	109.7	109.61	40	96.32	141.34	79
1.68	157.2	335.5	293.43	300.5	31.53	2.77	0.44	183.7	180.61	365.45	171.12	130.06	198.32	162.2	177.18	135.45	139	53.26	217.64	145.37	79	129.96	97.64	159	103.37	140.17	102.08	138.4	109.34
0	99.71	366.81	343.78	350.06	25.76	26.71	0	162.15	181.9	83.09	252.84	217.79	209.41	167.82	169.2	16.67	235.06	24.15	182.66	152.84	119	133.16	89.97	83.34	82.62	64.99	97.57	98.11	84.85
55.64	177.54	394.92	399	386.2	11.59	8.75	2.39	178.03	190.19	265.74	259	349.17	205.06	140.8	222.98	308.75	177.13	181.38	189.26	101.46	87.43	144.37	102.72	53.35	103.96	133.3	87.53	153.1	105.58