Analysis of Key Installation Protection using Computerized Red Teaming

Tirtha R. Ranjeet, Philip Hingston, Chiou-Peng Lam and Martin Masek
School of Computer and Security Science,
Edith Cowan University
2 Bradford Street, Mount Lawley, Western Australia, 6050.

Abstract

This paper describes the use of genetic algorithms (GAs) for computerized red teaming applications, to explore options for military plans in specific scenarios. A tool called Optimized Red Teaming (ORT) is developed and we illustrate how it may be utilized to assist the red teaming process in security organizations, such as military forces. The developed technique incorporates a genetic algorithm in conjunction with an agent-based simulation system (ABS) called MANA (Map Aware Non-uniform Automata). Both enemy forces (the red team) and friendly forces (the blue team) are modelled as intelligent agents in a multi-agent system and many computer simulations of a scenario are run, pitting the red team plan against the blue team plan.

The paper contains two major sections. First, we present a description of the ORT tool, including its various components. Second, experimental results obtained using ORT on a specific military scenario known as Key Installation Protection, developed at DSO National Laboratories in Singapore, are presented. The aim of these experiments is to explore the red tactics to penetrate a fixed blue patrolling strategy.

Keywords: Red Teaming, Evolutionary algorithm, Key Installation Protection

1 Introduction

This paper presents a tool, ORT (Optimised Red Teaming), which provides automated support for red teaming. We illustrate the use of the tool by exploring potential attack plans to defeat a defensive coastline patrolling strategy designed to protect a key installation.

Red teaming is a process that assists in finding vulnerabilities in a system, whereby the organization itself takes on the role of an "attacker" to test the system. In military organizations, the red teaming concept has long been used at various levels, including organizational and tactical. Traditionally, it is a manual process using humans as actors, resulting in a process that can be expensive, time-consuming, and limited from the perspective of humans "thinking inside the box" (Andrews, 2005, DoD, 2003, Meehan, 2007).

As a possible solution to the "human" limitations in manual red teaming, computerized red teaming uses agent-based simulation (ABS) in which autonomous agents taking on the roles of attacker and defender. Using such a system, where humans are not an intrinsic part of the simulation loop, allows many iterations of the problem to be simulated in a short space of time. This allows for the exploration of a wider range of possible attack/defence strategies, in a shorter time, utilising less real personnel than the traditional manual red teaming. Promising results from the automated simulation can then be checked for legitimacy and evaluated by the expert.

2 Related Work

Upton and McDonald (2003) first suggested using evolutionary algorithms and agent-based simulation for automated red teaming, in their case for testing proposed security procedures. They used an Evolutionary Programming algorithm to evolve the parameters of a red team strategy to defeat a fixed blue team strategy for defence of a fixed structure. The idea of combining agent-based simulation with an evolutionary algorithm has been further developed into the ART framework by researchers at Singapore's DSO and Nanyang Technological University (Choo et al., 2007, Chua et al., 2008, Xu et al., 2009). Further work on computerised red teaming has also been done at The University of New South Wales' Australian Defence Force Academy (Ang, 2006) using a simple (1+1) Evolution Strategy algorithm, coupled with WISDOM, a low-resolution simulation model for military simulations.

The ART framework integrates an optimisation algorithm with an agent-based simulation. It currently supports particle swarm optimisation and a multi-objective evolutionary algorithm as the optimiser, and several simulations models, chiefly MANA (Map Aware Non-uniform Automata) (Lauren, 2002). ART has been used in a series of data farming workshops (Lee et al., 2006, Sim et al., 2006, Wong et al., 2007) for applications including urban operations, maritime defence and anchorage protection, and is claimed to be able to discover non-intuitive tactics that are superior to those obtained by manual red teaming.

The aim of (Ang et al., 2006) study was to investigate the nature of the fitness landscape taking into account the personalities of the red and blue teams. The early part of the paper provides a useful survey of computational tools and techniques that are available for defence games.

Recently, (Hingston et al., 2010) proposed RedTNet, a network based modelling framework intended to support red teaming studies for critical infrastructure protection and strategy games.

3 Optimized Red Teaming (ORT)

ORT is an automated tool developed to assist the red teaming process in finding vulnerabilities in a security plan. The tool is used to explore a situation and identify potentials penetration strategies to 'break blue'. This helps subject matter experts to recognize weaknesses in their plan, which provides the opportunity to take action to address those weaknesses. MANA, an agent-based simulation application, is used to run simulations of the scenario, and a genetic algorithm (GA) is utilized to optimize the combatants' behaviours.

In this section, we describe the design of ORT and its components. The main components of ORT are shown in Figure 1 and are discussed below:

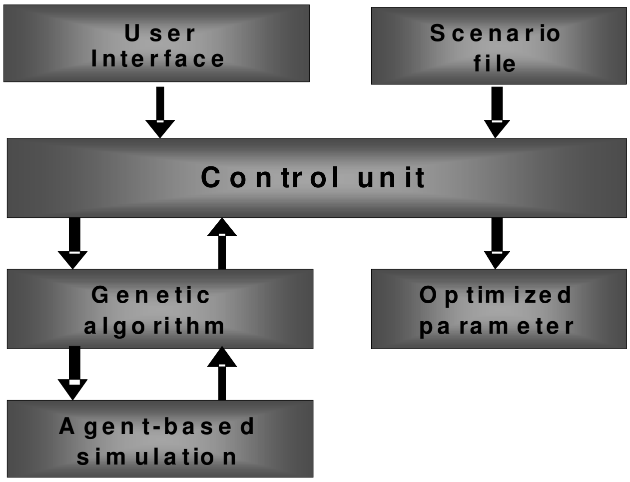

Figure 1: The framework of ORT

a. **User interface:** This is a graphical user interface which allows the user to supply required information to the program, including genetic algorithm parameters and agent personality parameters for each squad, and to select which squads to optimize.
b. **Scenario file:** This is a description of the particular scenario, in XML format, which contains details of the environment and at least two military squads with different intentions and targets. The scenario allows for different tactics to achieve the goal.
c. **Control unit:** This component controls the overall execution of the program, taking parameters as specified via the user interface. These are used to configure and execute the genetic algorithm, running simulations as needed to calculate fitness values.
d. **Genetic algorithm:** This takes parameter values from the control unit and executes a genetic algorithm, using fitness values calculated using agent-based simulations.
e. **Optimized parameters:** This is the optimized parameter values for the agents' personalities.
f. **Agent-based simulation:** The framework uses MANA as the agent-based simulation that runs scenarios in order to evaluate different parameter choices.

To use ORT to analyse a particular scenario, the user selects agent personality parameters via the user interface, which determines the structure of the chromosome to be used for the optimization process. The agent personality parameters are divided into three categories, agent situational awareness (SA), squad SA and inorganic SA. These represent personal, internal group and external group activities. Users have the option to choose which squads to optimize - the red, blue or others. The user also selects GA parameters such as population size, generation number, and crossover and mutation rate, and also the scenario file containing the details of the scenario to be examined. The optimization process can then be initiated.

The genetic algorithm is initialised using the specified GA parameters, and is executed as described in Section 3.1 below. When fitness values are required, sets of simulations are run using parameters values specified by the genome being evaluated. When the genetic algorithm terminates, the optimised parameter values are available to the user, who can then run further simulations to examine and understand the behaviours of the optimised squads.

In this way, the user may be able to identify and address weaknesses in the blue defensive strategy. The improved strategy can then be further tested in a similar way. An alternative is to use ORT to optimise the blue team against the optimised red plan. We illustrate this alternative in later sections (but note that there are potential difficulties, which are also discussed).

ORT makes use of both genetic algorithms and the MANA agent-based simulation. These are described briefly in the next subsections.

3.1 Genetic algorithm (GA)

In this subsection, we briefly explain the process of a typical genetic algorithm. The foundation of evolutionary algorithms (EAs) is evolutionary theory, which suggests that solutions to an optimisation problem can be derived by an evolutionary process that selects a best solution from a population (Abbass et al., 2001, Alcala et al., 2007, Veldhuizen, 1999, Zitzler, 1999). According to (Coello et al., 2007, Deb, 1999), EAs are adaptive heuristic search algorithms that derive the high quality solutions by using the principles of natural selection: each solution gets a chance to reproduce a certain number of times depending on its performance. Thus, quality results are achieved by selecting among the best solutions. Genetic algorithms are a specific kind of EA. The process of a typical GA may be described in pseudo code:

1. Generate an initial population.
2. Do until the termination condition is satisfied:
   a. Calculate the fitness of every individual.
   b. Start a new population.
   c. Do until new population is complete.
      i. Select two parents from the old population according to their fitness
      ii. Perform crossover and mutation to obtain two new offspring
      iii. Add the new offspring to the new generation.
3. Output optimized parameter values from the best individual in the population.

There are a number of design choices that must be made before applying a genetic algorithm to solve a specific problem of interest. Table 1 lists the GA design choices that are utilized in ORT:

Features	Name
Crossover	Simulated binary crossover (SBX)
Mutation	Polynomial mutation (PM)
Selection method	Stochastic universal selection
Elite individual	Only the best one
Initial population	Each genome is a sequence of randomly generated values from 0 to 100, representing parameter values for each agent's personality.

Table 1: GA features incorporated in ORT

3.2 Map Aware Non-uniform Automata (MANA)

MANA is a cellular automaton combat simulation model, designed at the New Zealand Defence Technology Agency (DTA). It includes a graphic user interface (GUI) that allows users to create new scenarios or loads external scenario files. The features of MANA include agents with situational awareness (SA), a terrain map, event driven personality and flexible waypoints. These features allow a rich set of parameters to be explored when running a scenario. SA influences agent behaviours in MANA. For example, an agent in a squad may detect enemy approaching near to the squad, the information they share among other agents to alert from the situation. Terrain maps are coded using colours to indicate traversabilty of the terrain. Event driven behaviours help agents to change their activity according to changes in the situation. MANA also contains its own analytical tools including a genetic algorithm (GA). These analytical tools can be used to find the suitable tactics in order to penetrate an opponents' strategy (Lauren, 2002, McIntosh et al., 2007).

4 Scenario description

The Key Installation (KIN) protection scenario was developed using MANA at DSO National Laboratories, Singapore. Basically, the scenario demonstrates the threats to KIN protection from non-military boats which try to penetrate the regular surveillance of the three blue boats. The fairly low speed blue boats patrol a specific area of the coastline with low level weapons. Conversely, the red boats without weapons try to penetrate the blue patrol to get into the land using different escaping tactics and routes (Chua et al., 2008).

In the original scenario, there are three KINs and three blue patrolling boats. Each blue boat has their patrolling route in which they constantly move to resist any penetrator. The blue surveillance route and KINs along with the initial positions of the red boats are depicted in Figure 2. The red boats are penetrators whose objective is to reach into the land by escaping from the blue patrol and destroy KINs.

4.1 Parameters

Each squad's behaviour is determined by a number of parameters. Table 2 lists these parameters. The default parameters used for the blue agents in the scenarios are depicted in Table 3.

The parameter values for the red team are evolved using the genetic algorithm. Thus the genome is a sequence of real parameter values in the ranges indicated in Table 2.

Characteristics Considered	**Description**	**Values in Range**
Movement Speed	The value determine the number of cells agents move in a given time step. Its range is 0 to 1000; however normalized to 100 so that an agent can move one cell per time step.	0 to 100
Agent SA – 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 - Agent stake 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 2: Selected agent personality parameters

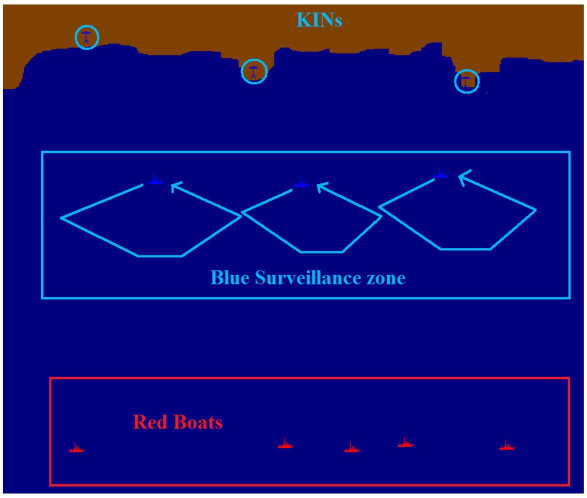

Figure 2. Scenario for Key Installation protection

Personalities			Normal Behaviour	Enemy Contact
Distance from Enemy	Agent SA	Enemy	0	100
		Enemy Threat 3	0	0
	Squad SA	Enemy Threat 3	0	0
Distance from Friends	Agent SA	Uninjured Friend	0	0
	Squad SA	Friend	0	0
Movement Speed			60	80

Table 3: Default characteristics of the blue agents

Each blue boat follows its specified route unless one of the red boats comes into their contact area. The values given under 'normal behaviour' section in Table 3 are all 0, meaning that they are neither aggressive to enemies nor affected by friendly boats. They circle their route at normal speed (movement speed is 60). When any blue boat finds a red boat within its sensor range, it switches to the 'enemy contact' parameters values, and its behaviour will become aggressive (the value of 100 indicates that it will chase after the enemy, and will do so with the greater speed of 80).

4.2 Measures of Effectiveness (MOE)

Two factors are considered as measures of effectiveness (MOEs) to evaluate the performance of the red team:

1. Maximizing the goal achievement – that is, breaking the blue boat patrolling tactics by getting at least one boat to the land.
2. Minimizing red casualties

These are combined to define the fitness function to guide selection in the genetic algorithm, using the formula:

Fitness = Red Goal Success Proportion * (Number of red agents)^2 - Mean red casualties + Number of red agents.

5 Initial experimentation

In order to explore the strategies available to each side in this scenario, we consider variations with different numbers of attacking boats. Every scenario has the same number of the blue agents, patrolling strategy and mission (which is to prevent the red boats attacking the key installation). The numbers of red boats is varied between two and five. 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.

For these experiments, following some preliminary testing, we set the GA parameter values as listed in Table 4 below:

Properties	**Values**
Agent-based simulation	MANA
Evolutionary algorithm	GA
Simulations per individual	20
Population size	20
Generations	50
Crossover Rate	60%
Mutation rate	1/population size
Number of experiments	20

Table 4: GA parameter values

In MANA, the simulation termination condition was set to 1000 simulation steps, or all red agents destroyed, or achieving the goal by any red agent (reaching the land). ORT executed 20000 (= 20 x 50 x 20) simulation runs in each experiment. It takes less than 1 second to evaluate each individual on a standard personal computer with 1GB RAM and 1.6 GHz CPU.

6 Discussion

Convergence graphs showing the fitness values over generations in the scenarios with the red agents ranging from two to five are depict in Figure 3, Figure 4, Figure 5 and Figure 6 respectively. Each graph shows the minimum, maximum and median values of best fitness values for each generation over 20 repeats of the genetic algorithm. Not unexpectedly, 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 experiments, we can see that the search converges quickly, in less than 20 generations. With five attackers, the genetic algorithm reliably converges to a solution with a fitness value close to 30. However, with only two attackers, convergence is much less reliable, with a range of final fitness values between 5 and 6. This may indicate that it is more difficult to find good solutions for the red team when there are fewer agents available. There is nonlinear relationship between the agent numbers and the number of red casualties, as casualties increase when there are many agents involved in the penetration process.

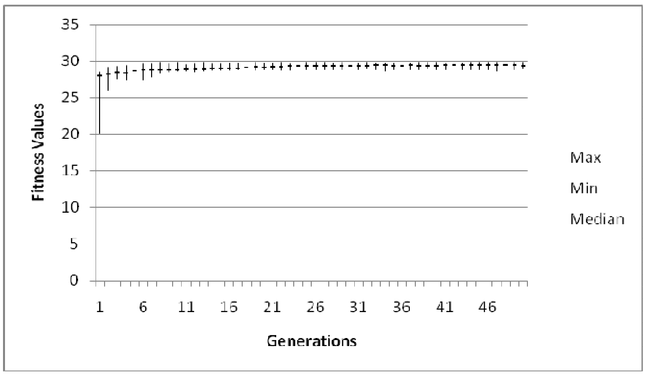

Figure 3. Minimum, maximum and median fitness values of the red team while considering five red boats.

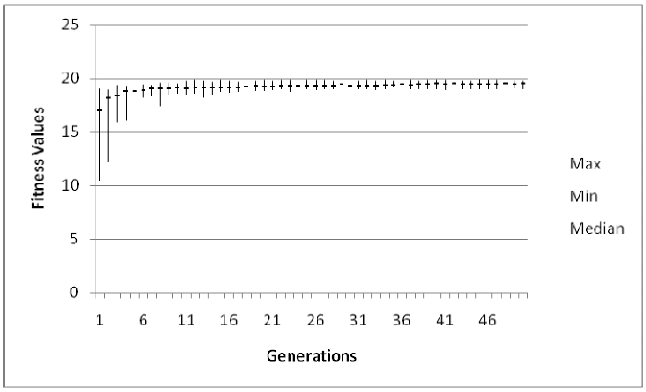

Figure 4. Minimum, maximum and median fitness values of the red team while considering four red boats.

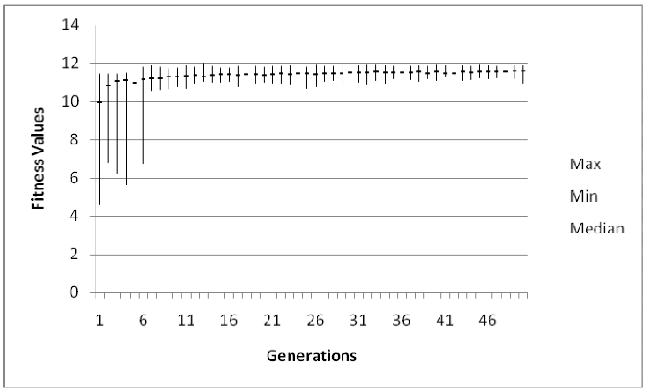

Figure 5. Minimum, maximum and median fitness values of the red team while considering three red boats.

The experiments show that the red teams alter their tactics and behaviours as the number of penetrator boats changes. Example tactics incorporated by the red team when different numbers of red boats are involved in blue penetration are depicted in Figure 7, Figure 8, Figure 9 and Figure 10 below.

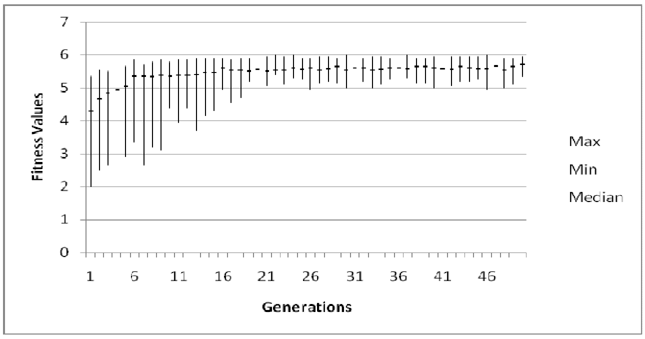

Figure 6. Minimum, maximum and median fitness values of the red team while considering two red boats.

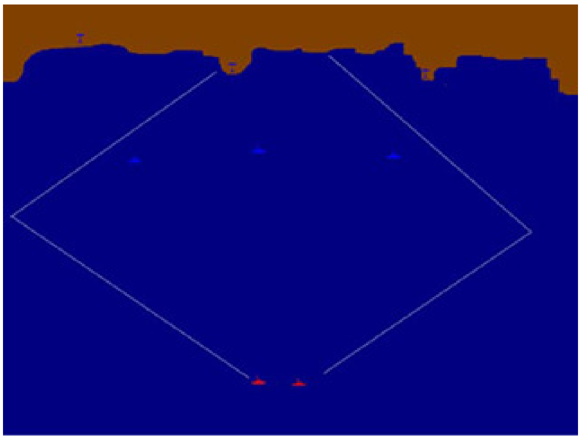

Figure 7. The route suggested by ORT for two red boats to penetrate the three blue patrolling boats.

In Figure 7, the red boats avoid confrontation with the blue boat and find a secure way to the land. When there are three red boats as in Figure 8, the tactics use one boat as a distraction so that other two can easily pass through the blue patrol formation.

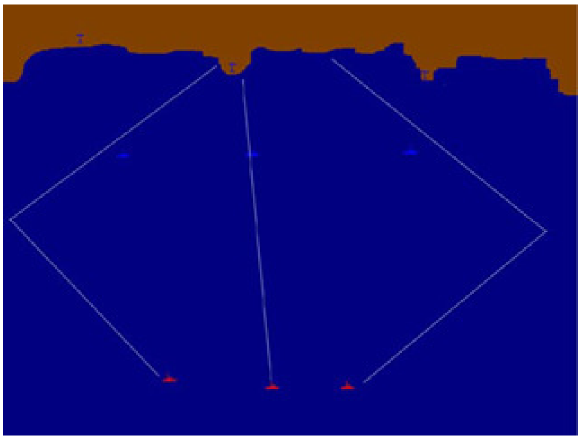

Figure 8. The route suggested by ORT for three red boats to penetrate the three blue patrolling boats.

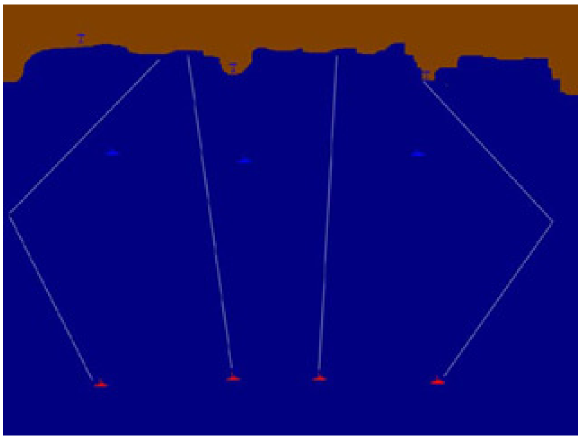

Figure 9. The route suggested by ORT for four red boats to penetrate the three blue patrolling boats.

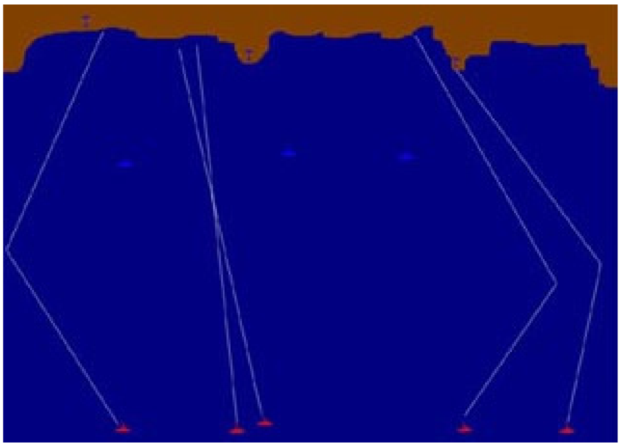

Figure 10. The route suggested by ORT for five red boats to penetrate the three blue patrolling boats.

Figure 9 and Figure 10 show a mixed strategy. The boats at the corner avoid confrontation whereas the others move towards the patrolling area by maintaining distance with friendly boats. Similar to the scenario with three red boats, the tactics use some distraction boats in order to allow the rest of the boat to achieve the goal.

The tactics show that with a smaller number of red boats, the red team should follow flanking tactics to achieve the goal. The result demonstrates the behaviour of internal cooperation among the red agents when the red boats are varied. The cooperation among the red boats is strong when a large number of red agents are involved in the penetration process. Conversely, they maintain distance if there is a smaller number of agents involved, which leads them to follow flanking strategies.

As the number of the red agents increases, their tactics change from flanking to direct confrontation. However, they avoid conflict and try to find a narrow escape between the blue patrolling routes to avoid casualties.

The personality values for the red team with two, three, four and five agents as suggested by ORT are depicted in Table 5, which indicate that the red agents stay away from the blue boats and they maintain distance between friendly agents also. The flanking tactics and increased speed help the red agents to avoid confrontation with the blue agents and reach the goal. The red teams with the given characteristics succeed almost 100% to achieve the goal while minimizing their casualties. The negative value under 'Enemy' shows they fear of the blue and stay away from their contact. The positive and negative value in 'Friend' rows show closeness and distance with the friendly boats.

<table>
<tr><td colspan="3">Personalities/ Boat No.</td><td>2</td><td>3</td><td>4</td><td>5</td></tr>
<tr><td rowspan="3">Distance from Enemy</td><td rowspan="2">Agent SA</td><td>Enemy</td><td>-90</td><td>-60</td><td>-83</td><td>-93</td></tr>
<tr><td>Enemy Threat 3</td><td>-95</td><td>-98</td><td>-99</td><td>-98</td></tr>
<tr><td>Squad SA</td><td>Enemy Threat 3</td><td>-85</td><td>-75</td><td>-90</td><td>-87</td></tr>
<tr><td rowspan="3">Distance from Friends</td><td>Agent SA</td><td>Uninjured Friend</td><td>-96</td><td>-35</td><td>30</td><td>50</td></tr>
<tr><td>Squad SA</td><td>Friend</td><td>-65</td><td>-20</td><td>22</td><td>35</td></tr>
<tr><td colspan="2">Inorganic SA</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="3">Movement Speed</td><td>100</td><td>100</td><td>100</td><td>100</td></tr>
</table>

Table 5: Personality of the red suggested by ORT for a red team with two agents

Scenarios with agent personalities as listed in Table 5 were further analysed to evaluate their effectiveness. For this, an additional 50 repetitions of each scenario were run in MANA. Table 6 tabulates the mean MOE and fitness values for different numbers of red agents.

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

Table 6: Mean casualties and success rate of optimized red team

The results in Table 6 indicate that there is a direct relation between the number of agents involved in penetration and their success rate. Conversely, there is negative relation between the number of agents and their attrition.

7 Further Experimentation

To further explore the strategy options, in response to the evolved red team, another experiment was devised to consider the blue agents to be optimized against the optimized red agents. For this, only the scenario with two red boats is considered. The default personality values for the red boats are shown in the second column of Table 5. GA parameters were the same as in the previous experiments, as depicted in Table 4.

Two factors are considered as MOEs, to evaluate the individuals: maximizing the red casualties and stopping the red boats to pass through the patrolling area. The formula used in fitness function is:

Fitness = Mean red casualties – Red goal success proportion + number of blue agents

ORT suggests characteristics for the blue team to stop the red boats as depicts in Table 7. The emerged tactics for the blue boats to respond the optimized red boats alter the blue behaviours make them more aggressive and active. Despite the use of flanking tactics by the optimized red boats, the later optimized blue boats are capable of taking action against them.

Against the default blue strategy, optimized red boats would reach destination almost 100% of the time; however, when the blue team are optimized their winning ratio is reduced by one third. The fall of the red winning ratio after blue optimization indicates that improved tactics can address the weaknesses of the plan if they are identified in advance.

Personalities			Normal Behaviour	Enemy Contact
Distance with Enemy	Agent SA	Enemy	0	100
		Enemy Threat 3	0	100
	Squad SA	Enemy Threat 3	0	25
Distance with Friends	Agent SA	Uninjured Friend	0	-84
	Squad SA	Friend	0	-71
Movement Speed			60	100

Table 7: Optimized personality of the blue agents

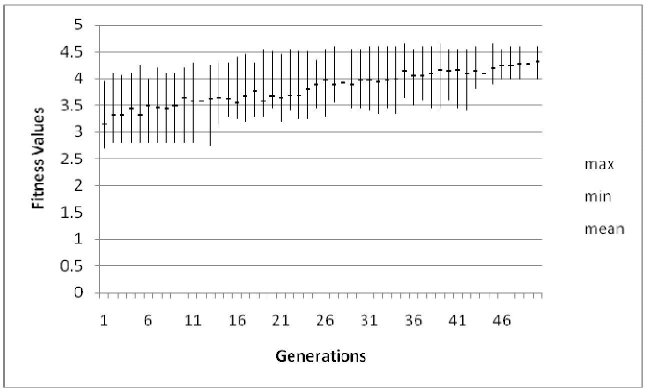

Figure 11. Maximum, minimum and mean fitness values of the blue team while considering two red boats trying to penetrate three blue patrolling boats in the scenario.

In order to monitor the progress of the GA, Figure 11 depicts 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 acquire when optimizing the blue team against already optimized the red team.

A word or warning is in order here – it would be wrong for the blue side to assume that its plans will now be effective against red attacks. It may be that different red tactics would defeat these blue tactics, which are only optimised against one specific type of red tactic. A comprehensive analysis would have to consider the range of possible red tactics and their likelihood.

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

Table 8: Mean casualties and success rate of red boats after optimizing the blue boats.

8 Conclusion

In this paper we demonstrated the use of ORT, as a tool to assist the red teaming process for detecting weaknesses in tactical security plans. We have seen different tactics emerge in response to the blue patrolling boats in different scenarios, and shown that we can develop blue tactics to respond to optimised red tactics. While the simple approach illustrated here can be used to gain valuable insights into a scenario, in general, the situation is very complicated and ventures into the realms of game theory. We intend to explore this in future work using co-evolutionary algorithms.

Acknowledgment

The authors would like to thank Defence Technology Agency (DTA) New Zealand for providing MANA and DSO National Laboratories, Singapore for sharing the scenarios.

References

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.*

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. *Proc. of the 16th IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE'07*, 1367-1372, London.

ANDREWS, P. 2005. Red Teams: Toward radical innovation. *IBM Advanced Business Institute in Palisades, New York.*

ANG, Y. 2006. A Networked Multi-Agent Combat Model: Emergency Explained. *University of New South Wales, New South Wales, Australia.*

ANG, Y., ABBASS, H. A. & SARKER, R. 2006. Characterizing warfare in red teaming *Systems, Man and Cybernetics, Part B: Cybermetics, IEEE Transactions,* 36**,** 268-285.

CHOO, C. S., CHUA, C. L. & TAY, S. V. Year. Automated Red Teaming: A Proposed Framework for Military Appliaction. *In: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007 London, England, United Kingdom. 1936-1942.

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.

Proceedings of the 2008 Winter Simulation Conference.
COELLO, C. A. C., LAMONT, G. B. & VELDHUIZEN, D. A. V. 2007. Evolutionary Algorithm for Solving Multi-Objective Problems, Kluwer, Boston.
DEB, K. 1999. Multi-objective Genetic Algorithms: problem Difficulties and Construction of Test Problems. *Evolutionary Computation,* 7**,** 205-230.
DOD 2003. The Role and Status of DoD Red Teaming Activities. *Defense Science Board Task Force*.
HINGSTON, P., PREUSS, M. & SPIERLING, D. 2010. RedTNet: A Network Model for Strategy Games. *IEEE Congress on Evolutionary Computation, Barcelona*.
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*.
LEE, M., ANG, D. & HUNG, L. F. 2006. Applying automated red teaming in an urban ops scenario *In scythe 1: Proceedings and Bulletin of the International Data Farming Community, Montery, CA, USA, Naval Postgraduate School***,** 2430.
MCINTOSH, G. C., GALLIGAN, D. P., ANDERSON, M. A. & LAUREN, M. K. 2007. MANA: Map Aware Non-Uniform Automata Version 4 User Manual.
MEEHAN, M. K. 2007. Red Teaming for Law Enforcement. *Police Chief,* 74.
SIM, W. C., CHOO, C. S., M-TIBURCIO, F., LIN, K. & SHEE, M. Year. Applying automated red teaming in a maritime scenario. . *In: In Scythe 2: Proceedings and Bulletin of the International Data Farming Community 2006 Montery*, CA, USA, NavalPostgraduate School. 2629.
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*.
VELDHUIZEN, D. A. V. 1999. Multi-Objective Evolutionary Algorithms: Classifications, Analysis and New Innovations. Ph.D. thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Tech., Wright-Patterson AFB, Ohio,
WONG, A. C. H., CHUA, C. L., LIM, Y. K., KANG, S. C., TEO, C. L. J., LAMPE, T. & ABBOTT, P. H. B. 2007. Applying automated red teaming in a maritime scenario. *In Scythe 3: Proceedings and Bulletin of the International Data Farming Community*, Montery, CA, USA, NavalPostgraduate School. 35.
XU, Y. L., LOW, M. & CHOO, C. S. Year. Enhancing automated red teaming with evolvable simulation. *In: GEC '09: Proceedings of the first ACM/SIEGVO Summit on Genetic and Evolutionary Computation*, ACM, 2009 New York, USA. 687-694.
ZITZLER, E. 1999. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. thesis, Swiss Federal Institute of Technology, Zurich.