Automated Red Teaming: A Proposed Framework for Military Application

Choo, Chwee Seng
DSO National Laboratories
20, Science Park Drive
Singapore 118230
65-67727125
cchweese@dso.org.sg

Chua, Ching Lian
DSO National Laboratories
20, Science Park Drive
Singapore 118230
65-67727376
chuacl@dso.org.sg

Tay, Su-Han Victor
Defence Science and Technology Agency
71, Science Park Drive
Singapore 118253
65-68795077
tsuhan@dsta.gov.sg

ABSTRACT

In this paper, we describe Automated Red Teaming (ART), a concept that uses Evolutionary Algorithm (EA), Parallel Computing and Simulation to complement the manual Red Teaming effort to uncover system vulnerabilities or to find exploitable gaps in military operational concepts. The overall goal is to reduce surprises, improve and ensure the robustness of the Blue ops concepts. The design of key components and techniques that are required to develop an ART framework are described and discussed. An experiment with a military scenario in Urban Operations (UO) was conducted and the results analyzed to demonstrate the capability of the ART framework. Results showed that Red Force survivability can be improved by 27% just by modifying behavioral parameters alone. These findings could be used by Blue Force to refine their tactics and strategy thereby ensuring robustness of plans and higher mission success.

Categories and Subject Descriptors

D.m Software, Miscellaneous.

General Terms

Experimentation.

Keywords

Red Teaming, Evolutionary Algorithms.

1. INTRODUCTION

Red teaming is a technique commonly used in the military to uncover system vulnerabilities or to find exploitable gaps in operational concepts, with the overall goal of reducing surprises, improving and ensuring the robustness of the Blue ops concepts. It is currently a manually intensive technique that typically brings together experts relevant to the system under consideration and who are then charged with identifying weaknesses. However, the vulnerability assessments made are usually "bounded" by the knowledge of these subject matter experts (SME).

By leveraging on the advancing technologies of Parallel Computing and Evolutionary Algorithms (EA), many millions of simulation runs can be generated and investigated using an automated process. The goal is to fix the Blue parameters and search for Red parameters that result in the "defeat" of Blue, within the least amount of time. Information obtained during this process can then be used to either enhance or assist the manual effort.

Preliminary attempts at Red Teaming using Evolutionary Algorithms and Evolvable Simulations were conducted by Upton et al [1-2]. Their work involved the utilization of single objective function Evolutionary Programming (EP) algorithm over a set of system parameters. They recognized the lack of flexibility in utilizing only system parameters and developed a concept called Evolvable Simulation that would alter the simulation agent's structure. The shortcoming of using single function EP is that it does not model the real world situation where multi-criteria or objectives might exist. Another issue with Evolvable Simulation is that legacy simulation models might not be easily altered and thus rebuilding those for the purpose of Red Teaming would require a substantial amount of effort and investment. Therefore in this paper, a framework is proposed to overcome this limitation of single objective-based function, and also solutions to integrate legacy simulation models into the framework.

2. OBJECTIVE

The objective of the paper is to present the design of key components and techniques that are required to develop an Automated Red Teaming (ART) framework that will provide a powerful, systematic and efficient capability to support our decision-makers. The capability of the ART framework is then demonstrated using a military scenario in Urban Operations (UO).

3. ARCHITECTURE DESIGN OF ART FRAMEWORK

The architecture consists of the following broad categories of modules:

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

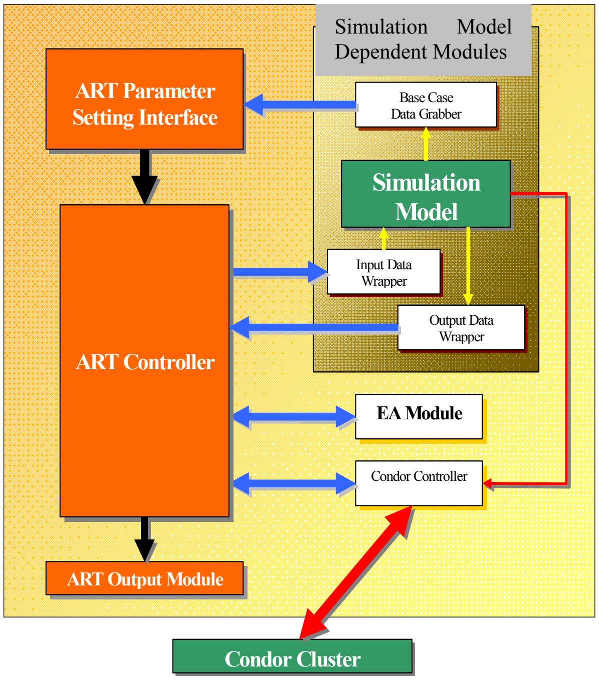

Figure 1. Architecture Design of ART Framework.

The architecture of the ART framework was coded in C++. It was designed to be modular and flexible enough to incorporate future extensions of simulation and evolutionary algorithm models. All the libraries are coded into DLL with MFC support which will be loaded during runtime. A standalone version was also developed to allow execution in a single PC environment.

3.1 Art Modules

These modules form the backbone of the ART architecture:

3.1.1 ART Parameter Setting Interface

This controls and displays the graphical interface for user to set the parameters to be considered for ART. It uses the Base Case Data Grabber to retrieve the list of parameters that can be set for the simulation model and display it in a Graphical User Interface (GUI) for ease of use. It also allows viewing, editing and creating the study profile, and view the outputs generated from EA Module.

3.1.2 ART Controller

This controls the flow and communication between the different modules. The module ensures the loading of all the libraries and database are successful and begin the ART process. The ART Controller will initialise the EA Module and get the initial population to be run in the clusters. Once it receives the outputs from the clusters, it will pass the results back to the EA Module for processing. The EA Module will evaluate whether to continue another run or to terminate. Once it is terminated, the EA Module will return the resulting population back to the controller, which will in turn pass it to the ART Output module for processing. If another run is required, the EA Module will return the next population to be run to the controller.

3.1.3 ART Output Module

This formats the output from the ART Controller Module and display the results, which are stored in a CSV file.

3.2 Simulation Model Dependent Modules

These are the modules that need to be developed for each simulation model before it can be "plugged and played" in the ART architecture. These modules act as wrappers for ART architecture to access and amend the input in the database of the simulation model. The modules are coded into the following Dynamic Link Libraries (DLLs) which will be loaded into the ART architecture during runtime:

3.2.1 Base Case Data Grabber (interface.dll)

Used by the ART Interface Module upon loading a simulation model and its base case database. To access the base case from the simulation model and returns the list of parameters that the user can set for ART.

3.2.2 Input Data Wrapper (input.dll)

Used by the ART Controller for writing data into the database of the simulation model. The wrapper will take in 2 types of parameters, namely, the attribute to be changed and the new value.

3.2.3 Output Data Wrapper (output.dll)

Used by the ART Controller for reading the output database of the simulation model. The wrapper takes in the attribute to be read and return the value to the controller.

3.3 Condor Controller Module

This controls the submission of jobs to the Condor clusters. Condor is a specialized workload management system which provides job queuing, scheduling policy and resource management for distributed computing [3]. It will consolidate the jobs to be executed from the ART controller and schedule them to the clusters. After all the jobs are completed, it will consolidate all the simulation output files back to the ART Controller for further processing.

3.4 Evolutionary Algorithm Module

This generates the population needed for the simulation runs. It also passes the terminating conditions to the ART Controller. After the all EA runs are completed, this module will return the output to the ART Controller. The architecture has been designed to allow future expansion, other EA or other possible type of algorithms can be added easily. As such, the EA Module is coded as an external library which will be loaded during runtime.

4. EVOLUTIONARY ALGORITHM

EA typically involves optimizing a certain fitness value that can be considered to be the objective function. However, restricting the analysis to a single objective is often unrealistic for conducting studies because these studies that are to be conducted often reflect real world problems which are really multi-objective in nature[1]. Therefore to improve the quality of the results obtained, it was decided to explore Evolutionary Multi-objective Optimisation (EMO) class of problem so that: [4-9]

1. The results (objectives) more closely reflect the kind of real world problems that are to be examined
2. A set of trade-offs, in contrast to an optimum solution, can be presented to the analyst to help him make better decisions.

As the key objective in our paper is to develop and demonstrate a Automated Red Teaming framework, therefore it was decided that the framework would make use of currently available EMO algorithms rather than to improve or develop a new EMO.

Studies and comparison of EMOs were conducted by Ziztler and Corne on the test functions from Deb [10-11]. Rankings of several EMOs were conducted separately by Ziztler and Corne, though the same types of test functions were used. The tests showed that Pareto-based algorithms performed better than non-Pareto ones; and for Pareto-based algorithms, those incorporating elitism clearly outperforms the rest. In the case of the three elitism-based EMOs, it was found that the Pareto Envelop-Based Selection Algorithm (PESA) had the fastest convergence due to its higher elitism intensity. However PESA, had problems with certain cases where it could be due to the fact that the algorithm does not always keep the boundary solutions. Strength Pareto Evolutionary Algorithm Version 2 (SPEA2) and Non-dominated Sorting Genetic Algorithm Version 2 (NSGA2) both showed the best performance overall though SPEA2 seems more robust than the other two algorithms when dealing with higher dimensional objective spaces. Therefore in the choice for the implementation of an EMO in ART, SPEA2, which was based on Evolutionary Programming (EP), was chosen for it's relatively simple to implement algorithm and also its best performance overall in finding the Pareto front.

5. APPLICATION: URBAN OPS

The objective of this study is to illustrate the capability of the ART framework using an UO scenario as a test-case. The problem to be investigated pertains to an UO involving the raiding and capturing of a deliberately-defended enemy key junction amidst the presence of hostile civilians. Instead of focusing on system and weapon performance factors, the focus was to explore how the various intangible characteristics of the Red Force can spoil Blue Force's plan.

[1] For example, an analyst in studying a conflict scenario is often required to explore the decision parameters that could simultaneously reduce the casualty numbers of friendly forces but maximise the damage to opposing forces.

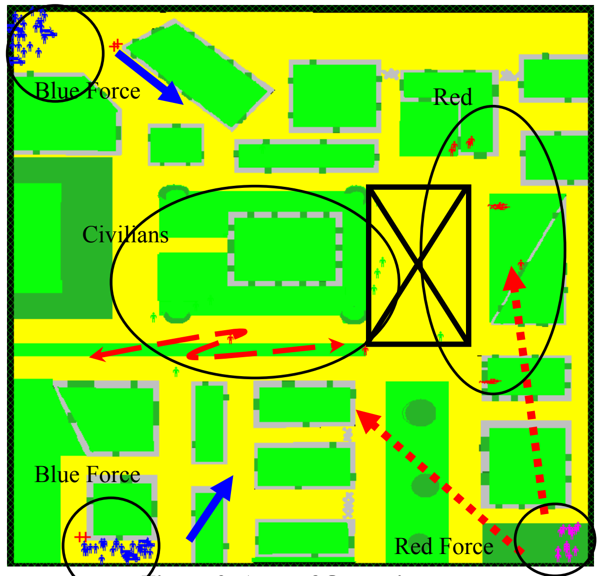

Figure 2. Area of Operations.

An Urban Area of Operations (AO) 2km by 2km in size was set up in MANA (see Figure 2). MANA, which stands for "**M**ap **A**ware **N**on-uniform **A**utomata", is an agent-based simulation tool developed by Defence Technology Agency, New Zealand. MANA was integrated into the ART framework.

In the AO, there were 2 platoons of Blue Infantry soldiers (21 soldiers per platoon). Each platoon was supported by 3 MG-mounted soft-skin vehicles, and attempting to overrun a key objective (Black crossed box) held by a section of Red Infantry soldiers (7 soldiers). Their general direction of movement is shown by the bold arrow. The Red's defence was assisted by two groups of Observation Posts (Ops, shown as crosses) which acted as early warning, two teams of snipers (4 snipers in total identified by red agent in prone position) and an additional reinforcement of 14 infantry soldiers that would be called up if the Red were outnumbered. The reinforcement take out position are shown in dotted arrow. The Red Force also positioned a group to patrol along the main axis as shown by the dashed zigzag line. The Blue agents' task was made more difficult by obstacles blocking some paths, and the hostile civilians congregating over the objective and randomly attacking the Blue agents.

Red and Blue Infantry agents were modelled slightly differently. The Blue Infantry agents were more mobile and were focused on reaching the objective, i.e. to occupy the key junction. The Red Infantry agents were more static and occupied defence positions around the objective. The Blue Infantry agents had a higher probability to kill at shorter range and a higher rate of fire. The Red Infantry agents were given higher concealment rates, as they were considered to be more familiar with their environment. The Red sniper agents were given higher sensor range and probability to kill to reflect their enhanced sighting capability and longer range weapons. Furthermore, the Red Infantry agents are scattered and hidden within the compounds of buildings under cover and concealment, and the Red snipers are located within buildings around the defending site. The Blue MG-mounted soft-skinned vehicles supporting the Blue Infantry agents were given higher protection and require greater number of hits to kill.

Furthermore, their weapons were accorded higher probability to kill simulating the higher lethality of the machine guns.

The civilian agents were dispersed within the AO, and they had the tendency to congregate at the objective, especially when Blue attacked the objective. They were also naturally hostile to Blue agents and would attack Blue upon sighting, although they were configured to be of low lethality. Their hostilities and behaviours towards the Blue agents were subjected to investigation in this study. Blue's Rule Of Engagement (ROE) against hostile civilians would be to fire back only when attacked.

6. SCOPE OF STUDY

As mentioned earlier, the intent was to explore how intangibles could lead Red to break Blue. Therefore, we short listed the parameters below (Table 1):

Table 1. Red Team Parameters Studied

Red Infantry Farming Parameters	Min	Max
Red Inf Clustering	-100	100
Red Inf Response To Injured Red	-100	100
Red Inf Individual Aggression	-100	100
Red Inf Squad Aggressiveness	-100	100
Red Inf Squad Cohesion	-100	100
Red Inf Stealthiness	0	99
Additional Civilian Parameters		
Civilian Clustering	-100	100
Civilian Aggression towards Blue	-100	100

A negative value for the parameter denotes an aversion to the particular attribute. For instance, -100 for clustering means the agents prefers to spread out than sticking as a group. Neutral value, 0, would mean that the agent is indifferent. For stealth, the value ranges between 0 and 100, however, the value 100 was not taken as it would mean the unit is completely invisible.

Additionally, questions also arose about how hostile civilians would affect the Blue force. Therefore we also examined the effects of aggression and clustering (as surrogate to rioting) on the overall performance level of the Blue.

For the purpose of this study, the Measures of Effectiveness (MOEs) to be collected for analysis were: Maximise Blue Attrition and Minimise Red Attrition.

7. RESULTS AND DISCUSSION

The MANA model was put through its paces in ART utilizing a 26 nodes processor cluster. The process took ~8 hours for a total of 40 iterations, with 30 evolving (child) and 30 archive (parent) sized populations. A total of 39,600 runs were executed, with each run taking less than 1s.

The following graph shows the evolution of the solutions through the 40 iterations. Note that set of "Final" series of points represents the set of possible solutions. However, in red teaming context, i.e. maximising Blue attrition and minimising Red attrition, the focus should be on the points within the circle.

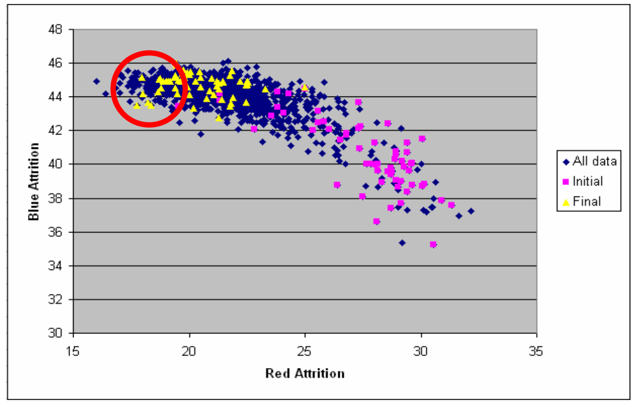

Figure 3. Results after iterations

7.1 Analysis of Results

The data were then analysed using the in-housed developed Clustering and Outlier Analysis Tool to identify the parameters associated with the best Red cluster, i.e. the cluster with the lowest Red attrition and highest Blue attrition [12]. Below is a summary of the results (Table 2):

Table 2. Results of Red Teaming Run

Best Characteristics	Base	Final (Mean)	Final (Var)
Red Inf Clustering	0	-31.98	+/- 41.14
Red Inf Response To Injured Red	0	11.04	+/- 1.58
Red Inf Individual Aggression	0	6.879	+/- 0.51
Red Inf Squad Aggressiveness	0	16.30	+/- 0.13
Red Inf Squad Cohesion	0	-14.54	+/- 0.24
Red Inf Stealthiness	0	89.15	+/- 0.67

Table 2 lays side-by-side the parameters values for the Base scenario, which will be used as basis of comparison, and the aggregated final values of the parameters after the Red Teaming run. The variances of the aggregated final values are also shown for an appreciation of their distribution.

The above results indicated that an effective Red Force would be highly stealthy (Red Inf Stealthiness = 89.15), slightly aggressive individually (Red Inf Individual Aggression = 6.879) and as a group (Red Inf Squad Aggressiveness = 16.30), with a propensity to move towards fellow injured Red (Red Inf Response To Injuried Red = 11.04) and tend not to move cohesively (Red Inf Squad Cohesion = -14.54). However, whether they cluster or not might not have significant impact on the outcome as evident from the large variance (+/- 41.14) associated with Red Inf Clustering.

Translating these results to the context of the scenario, this suggested that a high performance and dangerous Red Force would be one that is:

1. Elusive by making use of the concealment offered by the urban environment.
2. Mildly aggressive such that they would sustain their engagement with a numerically superior Blue Force, but yet not be overly aggressive to ensure that they would not be drawn out of their defence locations.
3. Well positioned and spread out in the defended positions to avoid being localized targets.
4. Ready to cover the positions of fellow injured Red infantry so that further attacks on the injured that can lead to high casualties is reduced.

The Red parameter values recommended by ART were adopted in the scenario and the results were generated and compared with that of the Base scenario. The results, as shown in Table 3, indicate a decrease in Red Attrition by 27.2% while the Blue attrition values increased by 6.3% (Vehicle) and 10.1% (Infantry).

Table 3. Comparison between Base Case Run and Red Teaming Results

	BASE CASE	RED TEAMING	EFFECT CAUSED BY RED TEAMING
RED FORCE			
ORBAT	29	**29**	
Mean Attrition & Percentage	19.69 (67.9%)	**11.80 (40.7%)**	↓ by 27.2%
CIVILIANS			
ORBAT	10	**10**	
Mean Attrition & Percentage	7.94 (79.4%)	**7.88 (78.8%)**	↓ by 0.6%
BLUE FORCE			
ORBAT (Infantry)	42	**42**	
ORBAT (Vehicle)	6	**6**	
Mean Infantry Attrition & Percentage	35.24 (83.9%)	**39.48 (94.0%)**	↑ by 10.1%
Mean Vehicle Attrition & Percentage	5.52 (92%)	**5.90 (98.3%)**	↑ by 6.3%

The Red Force recommended by ART has shown to achieve higher Blue attrition and lower own attrition. Therefore we see that by applying ART, we have effectively degraded performance of Blue's plan which would otherwise be not so easily identified.

Based on the indications of the red teaming results, the Blue should be prepared to face a possibly challenging Red Force and hence improve their capability and plans to counter the following red characteristics:

7.1.1 Stealth

Using better or more sophisticated sensors to identify stealthy Red agents hiding within buildings, can greatly aid in survivability of Blue. This is to ensure that the Red Force would not be elusive.

7.1.2 Cohesion

In order to counter the dispersion of the Red defending forces, it is important to derive plans to force the defence to cluster or co-locate at known positions to Blue. Carefully planted support fire and deceptive tactics can help Blue achieve this effect.

7.1.3 Aggression

Behavioural techniques to reduce aggression can also reduce Red's effectiveness. For instant, using a show of force (shock and awe) to intimidate the enemy.

With the results obtained, we have demonstrated the ability of using ART to search for associated parameter values that improved red force performance. In understanding what constitutes a potent Red Force, the Blue Force then has the ability to refine their plans and capability to ensure a more favourable and robust outcome when engaging an unpredictable Red Force.

7.2 Further Analysis with Civilian Parameters

To test the ART framework further, the effects of civilians' behaviour on the outcome were also explored to see if there were additional insights. 2 parameters - civilian's aggression against Blue Force and civilians clustering were considered. The following cases involving civilian parameters were put through the ART framework:

1. Base-Case. This serves as the baseline for comparisons (already done in the previous section).
2. Case A. Red Force parameters same as Base-Case, only evolve the civilian parameters.
3. Case B. Civilian parameters same as Base-Case, only evolve the Red Force parameters (already done in the previous section).
4. Case C. Red Force parameters adopt values recommended by ART in Case B; evolve only civilian parameters.
5. Case D. Evolve both Red Force and Civilian parameters starting from the Base-Case values.

The final values of the parameters recommended by the ART for each case were shown in Table 4. Note that the parameter values in italic are those with large variances and are judged to have insignificant impact on the outcomes. Shaded values are those not involved in the ART process.

Table 4. Table of Final Parameter Values obtained for all cases

PARAMETER VALUES	BASE CASE	CASE A	CASE B	CASE C	CASE D
Red Inf Clustering	0	0	*-31.98*	*-31.98*	*55.56*
Red Inf Response To Injured Red	0	0	11.04	11.04	*-58.89*
Red Inf Individual Aggression	0	0	6.879	6.879	41.75
Red Inf Squad Aggressiveness	0	0	16.30	16.30	9.7
Red Inf Squad Cohesion	0	0	-14.54	-14.54	0.31
Red Inf Stealthiness	0	0	89.15	89.15	98.22
Civilian Aggression	-100	*-75.64*	-100	*42.79*	-56.45
Civilian Clustering	0	*-12.81*	0	*-12.03*	63.78

The parameters values in Table 4 were adopted for each case and the respective attrition results were obtained and shown in Table 5.

Table 5. Comparison of attrition figures between all cases

ORBAT & ATTRITION FIGURES	BASE CASE	CASE A	CASE B	CASE C	CASE D
RED FORCE					
ORBAT	29	29	29	29	29
Mean Attrition (Percentage)	19.69 (67.9%)	20.12 (69.4%)	11.80 (40.7%)	10.46 (36.1%)	7.3 (25.2%)
CIVILIANS					
ORBAT	10	10	10	10	10
Mean Attrition (Percentage)	7.94 (79.4%)	8.7 (87.0%)	7.88 (78.8%)	7.24 (72.4%)	6.2 (62.0%)
BLUE FORCE					
ORBAT (Infantry)	42	42	42	42	42
ORBAT (Vehicle)	6	6	6	6	6
Mean **Infantry** Attrition (Percentage)	35.24 (83.9%)	35.22 (83.9%)	39.48 (94.0%)	39.1 (93.1%)	39.6 (94.3%)
Mean **Vehicle** Attrition (Percentage)	5.52 (92%)	5.44 (90.7%)	5.90 (98.3%)	5.84 (97.3%)	5.76 (96.0%)

7.2.1 Evolving Only Civilian Parameters

From the values shown in Table 4, it was observed that the civilian parameter values in Case A had evolved from the Base-Case. However, a look at their variances revealed that they were large and hence unlikely to have much impact on the results. This was also clearly apparent when the attrition figures in Table 5 for Case A and Base-Case were compared. The attrition figures for Case A were very similar to those in Base-Case. This implies that the Civilian Parameters, by themselves, may not have much effect on the outcome of the scenario.

Parameter values for Case C in Table 4 also show that although the civilian values have evolved, their variances are large. Similarly, the attrition figures for Case C in Table 5 also show little differences. This reiterated the belief that the two civilian parameters affecting the civilians' aggression and clustering behaviour would not have much impact on the outcome of the scenario.

7.2.2 Evolving both Red and Civilian Parameters

However, it becomes interesting when both the Red Force and Civilian parameters were evolved together, as in Case D. The civilian parameters for Case D had small variances and hence had impact on the attrition figures. The Red Force parameters were also slightly different from those obtained in Case B. The Red Force described in Case D was even more stealthy and aggressive than in Case B, and need not move cohesively when engaging the Blue Force. The civilians also tend to cluster and move away after their contact with the Blue Force.

In fact, the attrition figures in Table 5 showed that the Red Force in Case D is performing slightly better than Case B. Although the Blue attrition figures were similar than those in Case B, the Red Force and civilian attrition had both decreased by 15%. A more stealthy and elusive Red Force is expected to be more survivable and can afford to be more aggressive. The increased aggressiveness makes it more likely for the Blue agents to encounter Red agents compared to the civilian agents. Hence the civilian agent's attrition level is likely to drop. This shows that with the inclusion of civilian parameters, another solution point better than the pure Red Force parameters case was derived by the ART framework.

Therefore, it was observed that the civilian parameters studied, on their own, were not capable of affecting the outcome of the scenario. However, when they were grouped together with the Red parameters, ART was able to derive another solution point that proved to be better than the case when only Red parameters were studied. Nevertheless the relationship between the civilian parameters and the Red Force parameters was not apparent from the results of the ART runs.

8. CONCLUSION

We have described the concept of ART using EA in military context. In the first part, the architecture of ART was articulated and implemented. In the second part, an UO scenario was used to demonstrate use of ART as a framework for red teaming. Results showed that Red Force survivability can be improved by 27% just by modifying behavioral parameters alone. These findings could be used by Blue force to refine their tactics and strategy thereby ensuring robustness of plans and mission success.

It is important to note here that we are still experimenting with the ART framework. We will be testing it further with more military scenarios, and will be engaging military subject matter experts to perform manual red teaming, prior to the execution of ART, so that comparison can be made and the benefits of ART can be assessed. Further work will also be done to refine the implementation of the ART framework. In particular, constraints will be included to represent some trade-offs, e.g. there will be a potential "cost" to pay if we go stealth all the way. We will continue to explore other EAs and incorporate other simulation models in the ART framework.

Finally, we will continue to explore other potential applications can be spinned-off from this work. An obvious one is in the context of Blue Teaming, i.e. how the Blue breaks the Red's plan. This will naturally lead to the concept of Co-Evolution, i.e. Red Teaming vs Blue Teaming. Another potential application is in the calibration of model, e.g. what values should be assigned to the parameters that will result in certain desired outcomes.

9. ACKNOWLEDGMENTS

We would like to thank the following organizations and people that helped to make this R&D work possible:

Directorate of Research & Development in the Defence Science & Technology Agency (DRD-DSTA), Singapore, for funding this research.

Defence Technology Agency, New Zealand Defence Force, for sharing the Agent Based Model, MANA.

Steve Upton, Referentia, whose idea on Automated Red Teaming using Evolutionary Algorithm inspired us to work on this project.

10. REFERENCES

[1] Upton, S.C., and McDonald, M. J., Automated Red Teaming Using Evolutionary Algorithms, *WG31 – Computing Advances in Military OR*, 2003.

[2] Upton, S.C., Johnson, S.K., and McDonald, M. J., Breaking Blue: Automated Red Teaming Using Evolvable Simulations, *GECCO 2004*.

[3] Thain, D., Tannenbaum, T., Livny, M., Distributed Computing in Practice: The Condor Experience, *Concurrency and Computation: Practice and Experience*, volume 17, number 2-4, 2005, 323-356.

[4] Heitkoetter, J., and Beasley, D., eds. The Hitchhiker's Guide to Evolutionary Computation: A List of Frequently Asked Questions (FAQ), *Usenet: comp.ai.genetic*, 2001.

[5] Coello Coello, C.A., A short tutorial on evolutionary multiobjective optimisation, *In the Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimisation, Lecture Notes in Computer Science*, volume 1993, Springer, 2001, 21-40.

[6] Fonseca, C.M. and Fleming, P.J., *An overview of evolutionary algorithms in multiobjective optimisation*, Technical Report, 1994.

[7] Ziztler, E., Laumann, M., and Bieuler,S., *A tutorial on evolutionary Multiobjective optimisation* MOMH Workshop, 2002.

[8] Ziztler, E., Laumann, M., and Thiele, L., *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, TIK Report 103, 2001.

[9] Corne, D.W., Knowles, J.D., Oates, M.J., The Pareto Enveloped-based Selection Algorithm for Multiobjective Optimisation, *Parallel Problem Solving from Nature VI*, Springer, 2000, 839-848.

[10] Deb, K., Multiobjective Genetic Algorithms: Problem Difficulties and Construction of Test Problems, *Evolutionary Computation, volume 7, issue 3*, 1999, 205-230.

[11] Ziztler, E., Thiele, L., and Deb, K., Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, *Evolutionary Computation, volume 8, issue 2*, 2000, 173-195.

[12] Chua, C.L., *Clustering and Outlier Analysis for Data Mining*, 2005.