Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds.

APPLICATION OF MULTI-OBJECTIVE BEE COLONY OPTIMIZATION ALGORITHM TO AUTOMATED RED TEAMING

Malcolm Yoke Hean Low
Mahinthan Chandramohan

School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798

Chwee Seng Choo

DSO National Laboratories
20 Science Park Drive
Singapore 118230

ABSTRACT

Automated Red Teaming (ART) is an automated process for Manual Red Teaming which is a technique frequently used by the Military Operational Analysis community to uncover vulnerabilities in operational tactics. The ART makes use of multi-objective evolutionary algorithms such as SPEAII and NSGAII to effectively find a set of non-dominated solutions from a large search space. This paper investigates the use of a multi-objective bee colony optimization (MOBCO) algorithm with Automated Red Teaming. The performance of the MOBCO algorithm is first compared with a well known evolutionary algorithm NSGAII using a set of benchmark functions. The MOBCO algorithm is then integrated into the ART framework and tested using a maritime case study involving the defence of an anchorage. Our experimental results show that the MOBCO algorithm proposed is able to achieve comparable or better results compared to NSGAII in both the benchmark function and the ART maritime scenario.

1 INTRODUCTION

Red Teaming is a technique frequently used by the Military Operational Analysis (OA) community to uncover vulnerabilities and breaches in operational tactics with the intention to improve them. When carried out manually, Red Teaming demands close collaboration from a group of subject matter experts, whose knowledge and experiences dictate the success of the process. This is especially so in view of the complicated and multi-faceted nature of military operational tactics.

Automated Red Teaming (ART) (Choo et al. 2007) is a concept that complements the Manual Red Teaming (MRT) effort via the automated discovery of vulnerabilities and breaches in the targeted system through simulation. The technique works by accessing the (friendly) Blue Team's targeted system using a series of rigorous tests or strategies (used by the Red Team) and retaining those strategies that perform exceedingly well against the Blue's operational tactics. These retained strategies provide the subject matter experts with alternative views regarding the various vulnerabilities in the Blue's operational tactics so that they can be made more robust.

For ART to be effective, the simulation model used must be instrumented with different parameters for the strategies used by the Red Team so that a large and varied search space that consists of different possible combination of strategies can be found. At the same time, applying these strategies by the Red Team on the Blue Team may lead to multiple outcomes which can be used to evaluate the effectiveness of these strategies. Finding exact solution to such multi-objective problem with a large search space is computationally expensive and such problem cannot be solved within a polynomially bounded computation times. For this reason, the ART framework by Choo et al. (2007) employed evolutionary algorithms such as SPEA2 and MOPSO to efficiently find a set of non-dominated solutions from a large search space.

In this paper, we extend the ART framework by integrating a biologically inspired multi-objective bee colony optimization (MOBCO) algorithm to improve its performance. Our proposed MOBCO algorithm uses the non-dominated selection and crowding distance ranking approach.

This paper is organized as follows, Section 2 of the paper provides a brief summary of the related works followed by an introduction to MOBCO algorithm in Section 3. A brief description of the ART framework is presented in Section 4, while the experiments and the results are included in Section 5. Finally, Section 6 concludes with a summary of the paper.

2 RELATED WORK

2.1 Bee Colony Optimization

Bee colony algorithm is currently used in many areas to solve combinatorial optimization problems. This section describes the existing work done on this area of research.

Nakrani and Tovey (2004) first proposed the use of a honey bee algorithm for dynamic allocation of Internet servers. In their algorithm, servers and HTTP request queues in an Internet server colony are modeled as foraging bees and flower patches respectively. The experimental results show that the algorithm performs reasonably well in the dynamic allocation problem.

Bee colony algorithm has also been used to solve the travelling salesman problem (TSP). In Wong et al. (2008), the authors showed that the TSP could be solved efficiently using a bee colony optimization algorithm compared to other approaches. Here the benchmark problems EIL51, KROA100 and LIN318 were studied and the dimension of these problems were 51, 100 and 318 cities respectively. It was also identified that the performance of the algorithm generally dropped when the size of the problem instances increases. The main reason given for this problem was that the parameter settings needs to be fine-tuned to cater for different scenarios.

In Wong et al. (2008), the authors showed that a bee colony optimization algorithm with big valley landscape exploitation (BCBV) was able to achieve a comparable performance to the Taboo Search (TS) by Nowicki and Smutnicki (1996) using small number of iterations for the typical Job Shop Scheduling Problem (JSSP). It was stated that given ample computation time, BCBV was able to deliver performance better than TS on a set of Taillard JSSP benchmark data set.

In all the above work described, the problems solved by the bee colony optimization algorithm have only a single objective. The MOPSO algorithm proposed in this paper aims to address the optimization of multi-objective problems.

2.2 Evolutionary Algorithms for military operation research

Current work using EA for military operations research (OR) encompasses many different aspects of domain such as tactics design (Liang and Wang 2006), path planning for Unmanned Arial Vehicles (UAVs) (De la Cruz et al.2008, Mittal and Deb 2007) and mission planning (Ridder and HandUber 2005, Rosenberg et al.2008). This section describes existing work in the literature on the use of EA in military OR. Liang and Wang (2006) showed that their custom-made genetic algorithm was capable of designing evasive tactics for submarines in anti-torpedo warfare. They utilized a variable length individual representation to represent an anti-torpedo tactic.

Unlike conventional EAs, where both the recombination and mutation operators are applied to all variables of a solution, only the mutation operator is applied to submarine-related variables due to the variable-length of submarine's related variables. Comparatively, both the mutation and recombination operators are used on the decoys and jammers variables. The quality of the tactics generated by the EA were tested by attacking the submarine from 19 different directions and averaging the ten lowest values of the submarine's survivability. Although the tactics derived were robust against attacks from any of the 19 different directions simultaneously, they were not tested against attacks from multiple directions.

In the domain of flight planning for UAVs, Mittal and Deb (2006) successfully tested their hybrid algorithm to generate flight paths for UAVs through open as well as rough terrains. The hybrid algorithm comprises three stages: global search stage, solutions reduction stage and a fine tuning stage. NSGAII (Deb et al. 2002) was used in the global search stage to obtain a set of optimal solutions for the problem. In the solutions reduction stage, the solutions obtained by NSGAII were analyzed and reduced through the k-means clustering technique so as to make the amount of solutions found manageable for the decision makers. Lastly, a local search algorithm will be used for fine tuning the reduced set of solutions. De la Cruz et al. (2008) extended the work through a series of optimization and new ideas such as incorporation of refuelling and more complex modelling of UAVs. The EA utilized in their paper includes several new operators in addition to the usual operators such as crossover and mutation. An immigration operator was used to insert random individuals into the population so as to preserve diversity. Additionally, at every 25 generations interval, the EA uses a local search algorithm to optimize the best solutions found. The algorithm was tested to be effective in four different scenarios.

In the domain of mission planning, Ridder and HandUber (2005) successfully applied a non-elitist variant of NSGAII, Dynamic Non-Dominated Sorting GA (DNSGA) for Joint Suppression of Enemy Air Defences (JSEAD) mission planning. JSEAD is the application of lethal and non-lethal means to neutralise enemy air defences. Successful employment of JSEAD enhances the survivability of friendly aircrafts by preventing detection and engagement from enemy aircrafts. The DNSGA uses two forms of mutation: one for real-valued genes and another for discrete-valued genes. In their paper, Ridder and HandUber assessed the performance of DNSGA on two 3-objective scenarios. The first scenario requires the suppression of an operationally unrealistic placement of enemy air defences and the second scenario involves suppression of an operationally realistic air defence strategy for the enemy. Simulation results showed that EA is capable of discovering innovative and operationally useful ideas in real mission plans.

2.3 Automated Red Teaming (ART)

Automated Red Teaming has been applied on several military based scenarios. Choo et al. (2007) demonstrated the capability of ART using an urban operations scenario which involves the defence of an urban area controlled by the Red Team. Their work showed that ART was able to discover solutions which were useful for analyst to refine and design their strategies and thereby ensuring robustness of plans and higher mission success rates. Another work on ART was performed by Sim et al. (2006) on a maritime defence scenario. The maritime scenario involves the defence of a coastline by three Blue ships against attacks from five Red ships. Experimental results showed that ART was able to generate tactics that were unintuitive to the authors when performing MRT. Wong et. al. (2007) extended the work by Sim et al. (2006) by evaluating ART's effectiveness using an anchorage protection scenario. Similarly, their findings showed that ART is a useful tool for complementing the manual red teaming effort by providing useful and non-intuitive tactics.

3 HONEY BEE COLONY

The self-organized and collective behavior of colony insects enables them to solve multi-objective problems that are beyond the capability of individual members functioning alone. In the case of honey bees, this behavior helps them to explore the environment in search of flower patches and then pass the information of food source to the other bees of the colony when they return to the hive.

The foraging behavior in a bee colony remains mysterious for many years until Von Frisch (1974) translated the language embedded in the waggle dances of bees. Bees use waggle dance as a communication medium to describe the quality, distance and direction of the food source (flower patches) to other follower bees in the hive. Distance is conveyed by the type and duration of the waggle dance. Profitability rating of a patch is determined by the nectar quality, nectar bounty and distance from the hive. The follower bees are sent to the patches based on the profitability ratings. More bees are sent to patches that have higher profitability ratings. Further details of waggle dance can be found in (Dyer 2002, Biesmeijer and Seeley 2005).

3.1 Multi-Objective Bee Colony Optimization (MOBCO) Algorithm

The MOBCO algorithm is an extension of the original bee colony optimization algorithm to find a set of optimal solutions for multi-objective optimization problems. Figure 1 shows the pseudo code of the MOBCO algorithm. This algorithm requires two parameters to be set in advance, namely: population size (n bees) and maximum number of generations (i_{max}).

```
// initialization, see section 3.1.1
Initialize population B with a set of n random solutions, set generation i = 0
Initialize waggle dance B_waggle={}. Size of waggle dance set w=0.1n
While i<i_max
    // waggle dance selection, see section 3.1.2
    Select the set of non-dominated solution B_ND^i Є B
    Rank B_ND based on crowding distance, B_ND^i→B_NDC^i
    Select the set of dancing bees B_waggle^i = top w solutions B_NDC^i
    // waggle dance observation, see section 3.1.3
    Each bee in B^i follows a random dance by a bee in B_waggle^i
    // foraging, see section 3.1.4
    Each bee in B^i carries out foraging and updates its solution
    i = i + 1
Endwhile
```

Figure 1: Pseudo Code of MOBCO Algorithm

3.2 Population Initialization

The bee population B is initialized with n bees at the beginning of the evaluation (i.e. at generation zero).

$$B = \{b_1, b_2, b_3, \ldots, b_{n-1}, b_n\}$$

Each bee b_i has an associated flower patch p that represents a solution which consists of k normalized objectives values, $p = \{p_1, p_2, \ldots, p_{k-1}, p_k\}$, $p_j \in \{0,1\}$ for $1 \leq j \leq k$. Each solution $b_i.p$ is randomly initialized in generation zero.

The set of bees performing waggle dance is initially empty. A set of w bees will be selected to perform waggle dance in each generation. In this paper, w is set to 10% of the colony size, i.e. w=0.1n.

3.2.1 Waggle Dance Selection based on Crowding Distance

Only small number of bees are allowed to dance in the hive. The selection of bees for waggle dance is based on the ranking of their solution using non-dominated sorting and crowding distance. In our algorithm, the solutions by all bees are first sorted based on their pareto strength. The set of solutions with strength zero is selected to form the non-dominated set B_{ND}. If the size of B_{ND} is less that w, then solutions with increasing strength (e.g. strength 1 and then strength 2) are selected to top up the set to size w. The set B_{ND} is then sorted in an increasing crowding distance order. The top w solutions in the set B_{ND} that have the highest crowding distance (least crowded) are selected to form the set of bees B_{waggle} for waggle dance.

Figure 2 shows a two-objective example of the crowding distance technique. Each point in the diagram represents a solution. Based on the diagram, the crowding distance for the point i can be computed by adding its distance from i-1 and i+1. This technique calculates the average distance of two points on either side of a selected point along each of the objectives. Boundary solutions which have the highest and lowest objective values are given the maximum distance so that they are always retained. Lastly, the final crowding distance is computed by adding the crowding distance obtained in each objective. Crowding distance estimator has been used in other algorithms (Li et al. 2003, Raquel and Naval 2005, Sierra and Coello 2005). The pseudo code for computing crowding distance can be found in (Deb et al. 2002).

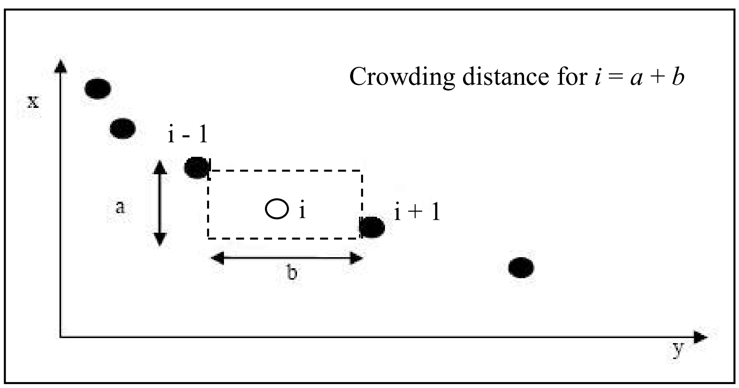

Figure 2: Crowding Distance

3.2.2 Waggle Dance Observation and Adoption

The majority of the bees in the hive will follow the bees that are selected to perform waggle dances. In our algorithm, 70% of the bees will follow a dance selected from the top 50% (in terms of crowding distance) of the bees in B_{waggle}, 20% of the bees follow the next 30% of the dance in B_{waggle}, and the last 10% of the bees follow the balance 20% of the dance in B_{waggle}.

3.2.3 Foraging and Updating of Patches

Each bee b_i will carry out foraging based on the dance it observed. However, it does not go directly to the flower patch indicated by the dance it observed. Instead, the flower patch it goes to is updated based on the following:

$$b_i.p = b_i.\mathrm{p} + r(b_j.\mathrm{p} - b_i.\mathrm{p}) \quad \text{and} \quad -1 \leq r \leq 1$$

In the above equation, the update is carried out on each objective value in $b_i.p$ respectively. For each corresponding objective value p_{a1} in $b_i.p$ and p_{a2} in $b_j.p$, if $p_{a1} >= p_{a2}$, then no update will be carried out for the variable p_{a1}. If $p_{a1} < p_{a2}$, then p_{a1} will be updated by the difference between p_{a1} and p_{a2} multiplied by a random number r ($-1 \leq r \leq 1$).

4 AUTOMATED RED TEAMING

The architectural design of the ART framework consists of several components as shown in Figure 3. It is designed to be modular and flexible to permit future extensions to the framework. We provide a brief description about the architecture in this paper. Detailed explanations about the architecture are given in Choo et al. (2007).

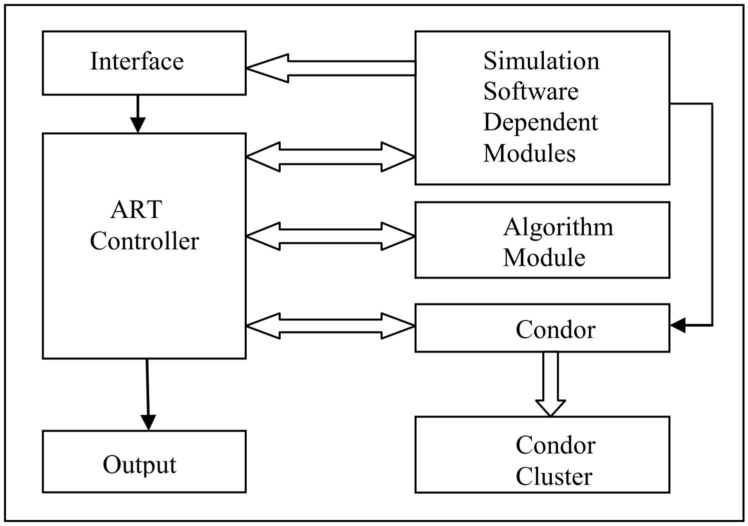

Figure 3: The ART Framework

The interface module controls the display and allows users to set parameters to be considered for ART. It allows for the viewing, creation and editing of the study profile. The role of the ART controller is to facilitate communications between the different modules. Additionally, it also ensures proper loading and execution of the different modules. The output module is responsible for the formatting and creation of the results into a CSV file.

Support for simulation software is provided by creating software dependent modules for each of the simulation software. These modules act as wrapper for the ART to access and amend the input to the simulation model. MANA, an agent-based simulation program, is currently supported in the ART framework. MANA (Galligan et al. 2005) is created by the New Zealand's Defence Technology Agency specifically for combat or other military related modelling. It has the capability to model complex relationships and interactions between agents as well as different environmental conditions.

The Condor module is responsible for the submission of jobs to the Condor clusters. Lastly, the algorithm module was designed to work as an external library which will be loaded during runtime. The MOBCO algorithm described in this paper is integrated into ART algorithm module as an external library.

5 EXPERIMENT AND RESULTS

5.1 Benchmark Problem

In this section we present the performance comparison using a set of standard mathematical benchmark function from the ZDT-series. The ZDT series of benchmark problems are created by Zitzler et al. (2000) for comparisons of multi-objective algorithms. Each problem in this series involves a particular feature that is known to cause difficulty in the optimization process. In this paper we consider the problems ZDT 1, 2, 3, 4 and 6. The performance of MOBCO is compared against another well known multi-objective optimization algorithm NSGAII (Deb et al. 2002). The true Pareto front used in this experiment is from the jMetal website (http://jmetal.sourceforge.net/).

5.1.1 Results

Both algorithms were allowed to run up to 100,000 fitness evaluations for each ZDT function. This allows the algorithms to converge towards the Pareto-optimal front. Each set of experiments is repeated ten times. Tables 1 and 2 summarize the experiment results obtained from running the MOBCO and the NSGAII algorithms on the ZDT problems.

Table 1 shows the mean and standard deviation for the convergence metric Inverted Generational Distance (IGD) (Sierra and Coello 2005). IGD uses the true Pareto front as a reference and compares each of its elements with respect to the front produced by an algorithm. It is defined as:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}$$

where n is the number of elements in the true Pareto front, and d_i is the Euclidean distance between each of elements and the nearest member from the set of non-dominated solutions found by the algorithm. IGD=0 indicates that all non-dominated solutions found by the algorithm are in the true Pareto front of the problem. Likewise a high value for IGD indicates that the obtained front is far from the true Pareto Front.

Table 2 shows the mean and standard deviation for the SPREAD (Deb et al. 2002) diversity metric. The spread indicator measures the extent of spread achieved among the obtained solutions using the equation below.

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} \left| d_i - \bar{d} \right|}{d_f + d_l + (N-1)\bar{d}}$$

d_i is the Euclidean distance between consecutive solutions in the obtained non-dominated set of solutions. d_f and d_l are the Euclidean distance between the boundary solutions of the obtained non-dominated set. In the event where the obtained solutions are well-spread Δ will be equal to zero.

Table 1: Results for the Inverted Generational Distance (IGD) Metric

	MOBCO		NSGAII	
	Mean	Standard Deviation	Mean	Standard Deviation
ZDT 1	1.73×10^{-4}	6.50×10^{-6}	1.91×10^{-4}	1.08×10^{-5}
ZDT 2	1.87×10^{-4}	1.73×10^{-5}	1.88×10^{-4}	8.36×10^{-6}
ZDT 3	2.33×10^{-4}	1.86×10^{-5}	2.59×10^{-4}	1.16×10^{-5}
ZDT 4	1.70×10^{-4}	1.92×10^{-5}	1.84×10^{-4}	9.86×10^{-6}
ZDT 6	1.33×10^{-4}	5.66×10^{-6}	1.59×10^{-4}	1.24×10^{-5}

Table 2: Results for the SPREAD Diversity Metric

	MOBCO		NSGAII	
	Mean	Standard Deviation	Mean	Standard Deviation
ZDT 1	1.65×10^{-1}	1.08×10^{-2}	3.83×10^{-1}	3.14×10^{-2}
ZDT 2	1.88×10^{-1}	8.90×10^{-3}	3.52×10^{-1}	7.25×10^{-2}
ZDT 3	7.34×10^{-1}	4.06×10^{-3}	7.49×10^{-1}	1.49×10^{-2}
ZDT 4	1.89×10^{-1}	1.31×10^{-2}	3.96×10^{-1}	2.94×10^{-2}
ZDT 6	1.51×10^{0}	1.73×10^{-1}	4.80×10^{-1}	4.49×10^{-2}

The results shows that for the Inverted Generational Distance (IGD) metric, MOBCO is able to achieve a good convergence compared to NSGAII. For ZDT 2 the difference between the two algorithms is very small. For the other four test functions, MOBCO clearly out-performed NSGAII. For the SPREAD diversity metric MOBCO has a good diversity for ZDT 1, 2, 3 and 4 while NSGAII has a good diversity for ZDT 6. The mean value of the MOBCO diversity is almost half of the NSGAII value for ZDT 1, 2 and 4. Comparing the results of both algorithms for the IGD and SPREAD performance metrics, we can conclude that MOBCO is able to achieve better performance compared to NSGAII on the ZDT benchmark.

5.2 Maritime Case Study

Maritime protection has become very important as commercial ships in the anchorage face many threats from pirates and terrorist attacks. The anchorage defense scenario was first hypothesized by Wong et.al (2007) in their paper for analyzing the variability of ART using the SPEA2 (Zitzler et al. 2001) algorithm. In this scenario, the responsibility of the Blue Team was to conduct patrols to guard against threats on an anchorage. A maximum of ten ships are allowed to be anchored in any location within the anchorage. Red forces would attempt to penetrate the Blue's defence and inflict damages on the anchored vessels. Any damages done to the commercial ships will have severe economic repercussions and hence should be prevented. The anchorage covers an area of 30nm by 10nm (1 nm = 1.852km). The Area of Operations (AO) was designed to be 100nm by 50nm so as to allow the Red Team more depth in their movement. The total number of Red and Blue crafts considered in the study are 5 and 7 respectively.

5.2.1 Patrolling Strategy

Figure 4 shows the patrolling strategy employed by the Blue force. The outer rectangle represents the AO, while the inner shaded rectangle represents the anchorage. The patrolling strategy is made up of two layers of patrolling: an outer patrol and an inner patrol. The lines outside the anchorage depict the movement of the outer patrols while the lines inside the anchorage represent the movement of the inner patrols. This strategy was designed to minimize the exploitable gaps in the patrolling. The outer patrol consists of four smaller but faster boats. They provide the first layer of defence against any attacks while the larger and heavily armoured ships inside the anchorage take charge of the second layer of defence. The remaining three crafts are the bigger vessels that conduct patrols within the anchorage.

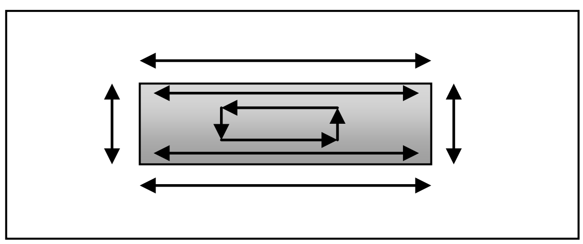

Figure 4: Patrolling Strategy of Blue PVs

5.2.2 Blue and Red Force Characteristics

Table 3 lists the settings for the respective forces in the scenario. For the Blue force, the inner and outer patrols was given different hit probability for their weapons. Also, the manoeuvring speed of the outer patrols is faster than that of the inner patrols. The outer and inner patrols require hits-to-kill of 2 and 5 respectively to differentiate them as the smaller and bigger crafts. The Red force is given the ability to damage or kill the Blue force with low hit probability. Each Red craft was allowed to kill multiple CVs and Blue crafts. Both forces are not given any communication capabilities.

The aim of the study is to discover Red's strategies that are able to breach the Blue's defence. The efficiency of the algorithm is measured by the number of Blue casualties with respect to the number of Red casualties.

5.2.3 Decision Variables

In ART, each solution is represented by a vector of real-valued numbers which in turn represents different components related to the Red force's behaviour. As the number of decision variables and Red crafts increases, the search space for the problems becomes larger as there are more potential solutions available.

Table 3: Settings for Red and Blue Forces

Units	Quantity	Maneuver Speed	Detection Range	Range	Probability
Red Craft (RC)	5 (3 top, 2 bottom)	16 knots	2nm	2nm (PV) 2nm (OP) 1nm (CV)	5% (PV) 5 %(OP) 100% (CV)
Inner Patrol Vessel (PV)	3	8 knots (Patrol) 16knots (Chase)	6nm	2nm	80%
Outer Patrol (OP)	4	16 knots (Patrol)	6nm	2nm	50%
Commercial Anchored Vessels (CV)	10	Stationary	Nil	Nil	Nil

Decision Variable	Minimum	Maximum	Locations on AO
Home Position (Top Red crafts)	[0,0]	[399,39]	
Home Position (Bottom Red Crafts)	[0, 160]	[399,199]	
Intermediate Waypoints	[0, 40]	[399,159]	
Final Position (Top Red Crafts)	[0, 160]	[399,199]	
Final Position (Bottom Red Crafts)	[0,0]	[399,39]	
Aggressiveness to Blue PV	-100	100	-
Cohesiveness	-100	100	-
Determination	20	100	-

Figure 5: Decision Variables for a Single Red Craft

Figure 5 shows the decision variables for a single red craft. The home and final position together with the intermediate waypoint represents the movement path for a Red craft. Psychological elements were also included in the decision variables to understand how these elements affect the Red force. The aggressiveness determines the reaction of individual Red craft when it detects a patrol. Cohesiveness affects whether the Red crafts attack as a group or individually whereas determination affects the Red crafts willingness to attack.

Setting a boundary for respective decision variables prevent exploration in regions outside the area of concern. In the case of the Anchorage Defence scenario, the problem scope was narrowed down by enforcing that the Red crafts start at a specific region in the AO. Figure 5 shows the range and their respective locations (yellow) on the AO. Three of the Red crafts were configured to attack the anchorage from the north while the remaining two attack from the south respectively. This enforces the Red force to perform multi-directional attack at the anchorage. However, it does not prevent the Red crafts from gathering at a location near the anchorage prior to an attack. Additionally, the final positions of the Red crafts were fixed to a region outside the anchorage to simulate escapes from the anchorage after successful attacks. The Red crafts' aggressiveness against the Blue force was varied from unaggressive (-100) to very aggressive (100). Likewise, the cohesiveness of the Red crafts was varied from independent (-100) to very cohesive (100). Lastly, a minimum value of 20 was fixed for the determination parameter force the Red crafts to move towards their waypoints and prevent inaction.

5.2.4 Results

Two Measure of Effectiveness (MOEs) were used for computing a given solution. The two MOEs are:

- Mean Red Casualty (Minimise)
- Mean Commercial Vessels Casualty (CV Casualty) (Maximise)

Each algorithm was configured to perform a maximum of 5000 evaluations on the scenario. The MOE obtained for each solution is the mean value computed from the end state of 20 replications of the simulation in MANA. The settings for the two algorithms are shown in Table 4.

Table 4: MOBCO and NSGAII Settings

	MOBCO	NSGA2
Population Size	100	100
Number of Generations	50 (5k evaluations)	50 (5k evaluations)
Crossover rate	N/A	0.8
Mutation rate	N/A	0.03

Figure 6 shows the relationship between the Red and Blue casualties for both algorithms, MOBCO and NSGAII. Our experimental results show that blue casualties increase with red casualties with both algorithms. MOBCO generates the highest Blue casualties with lesser Red casualties (or higher Red survival rate) compared to NSGAII. From Figure 6, we can see that results obtained from MOBCO has the maximum number of Blue casualties of 5.45 with 2.2 Red casualties whereas NSGAII obtained the same amount of Blue casualties with 2.65 Red casualties. In the graph, MOBCO has 6 non-dominated solutions (compared to the solutions obtained by NSGAII) out of a total of 8 solutions found. Comparatively, NSGAII has only 5 non-dominated solutions (compared to the solutions obtained by MOBCO) out of the 12 solutions found. Comparing the results obtained by both algorithms, we can conclude that MOBCO performs similar or better against NSGAII, thus making MOBCO algorithm a better alternative over the currently available evolutionary algorithms that work with the ART framework.

6 CONCLUSION

In this paper, the application of the MOBCO optimization algorithm to Automated Red Teaming is analyzed. The algorithm is based on the foraging and the self organizing behavior of bees. The performance of MOBCO is compared against the NSGAII algorithm first using a set of mathematical benchmark functions. Experimental results on the ZDT series of benchmark problems show that MOBCO achieved a better performance over NSGAII. Following that, the MOBCO is tested in the ART framework using an anchorage protection scenario. Experimental results revealed that MOBCO can deliver a comparable performance to NSGAII in ART. For future work, we will refine the MOBCO algorithm further to improve its accuracy and spread. We will also carry out benchmarking on the different algorithms in ARTS to create guidelines on which algorithms to use under different circumstances.

ACKNOWLEDGEMENTS

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

- Directorate of Research & Development in the Defence Science & Technology Agency (DRDDSTA), Singapore, for funding this research.
- Defence Technology Agency, New Zealand Defence Force, for sharing the Agent Based Model, MANA.

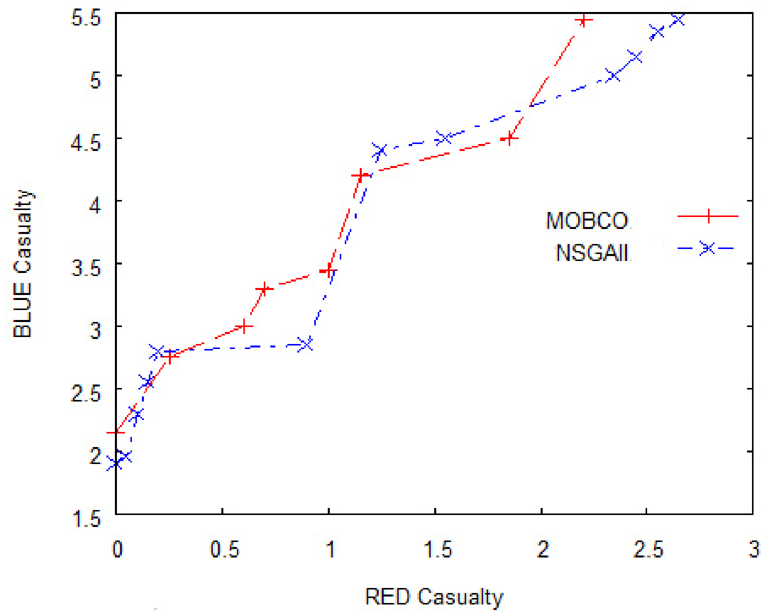

Figure 6: Relationship between Red Casualty and Blue Casualty

REFERANCES

Adams, J., E. Balas and D. Zawack. 1988. The Shifting Bottleneck Procedure for Job Shop Scheduling. *Management Science*, 34(1): 391-401.

Biesmeijer, J.C. and T. D. Seeley. 2005. The Use of Waggle Dance Information by Honey Bees Throughout Their Foraging Careers. *Behavioral Ecology and Sociobiology*, 59(1): 133-142.

Chong, C. S., M. Y. H. Low, A. I. Sivakuma and K. L. Gay. 2005. Using Simulation Based Approach to Improve on the Mean Cycle Time Performance of Dispatching Rules. In *Proceedings of the 2005 Winter Simulation Conference*. eds. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 2194-2202. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Choo, C. S., C. L. Chua, and S.-H. V. Tay. 2007. Automated Red Teaming: A Proposed Framework for Military Application. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, 1936–1942, New York, NY, USA. ACM.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 6(2): 182-197.

De la Cruz, J.M., E. Besada-Portas, L. Torre-Cubillo, B. Andres-Toro, and J. A. Lopez-Orozco. 2008. Evolutionary Path Planner for UAVs in Realistic Environments. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. 1477-1484.

Dyer, F.C. 2002. The Biology of the Dance Language. *Annual Review of Entomology*. 47: 917-949.

Galligan, D.P., M. A. Anderson, and M. K. Lauren. 2005. Map Aware Non-uniform Automata Version 3.0. Operations Analysis Section, Defence Technology Agency, New Zealand.

Liang, K.H. and K.M. Wang. 2006. Using Simulation and Evolutionary Algorithms to Evaluate the Design of Mix Strategies of Decoy and Jammers in Anti-torpedo Tactics. In *Proceedings of the 2006 Winter Simulation Conference*, eds. L. R. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1299-1306. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Li, X. 2003. A Nondominated Sorting Particle Swarm Optimizer for Multiobjective Optimization. In *Proceedings of 5th Annual Conference on Genetic and Evolutionary Computation*. 37-48.

Mittal, S. and K. Deb. 2007. Three-dimensional Offline Path Planning for UAVs using Multiobjective Evolutionary Algorithms. In *Proceedings of IEEE Congress on Evolutionary Computation*. 3195-3202.

Nakrani, S. and C. Tovey. 2004. On Honey Bees and Dynamic Allocation in an Internet Server Colony. Adaptive Behavior, 12(3-4): 223-240.

Nowicki, E. and C. Smutnicki. 1996. A Fast Taboo Search Algorithm for the Job Shop Problem. *Management Science*, 42(6) : 797-813.

Raquel, C. R and P. C. Naval. 2005. An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*. 257-264, New York, NY, USA. ACM.

Ridder, J. P. and J. C. HandUber. 2005. Mission Planning for Joint Suppression of Enemy Air Defenses using a Genetic Algorithm. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*. 1929-1936, New York, NY, USA. ACM.

Rosenberg, B., M. Richards, J. T. Langton, S. Tenenbaum, and D. W. Stouch. 2008. Applications of Multi-Objective Evolutionary Algorithms to Air Operations Mission Planning. In *Proceedings of the 2008 GECCO Conference Companion on Genetic and Evolutionary Computation*. 1879-1886.

Sierra, M and C. Coello. 2005. Improving PSO-based Multi-objective Optimization using Crowding, Mutation and epislon-Dominance. In *Third International Conference on Evolutionary Multi-Criterion Optimization*. 505-519.

Sim, W.C., C. S. Choo, F. M-Tiburcio, K. Lin, and M. Shee. 2006. Applying Automated Red Teaming in a Maritime Scenario. In *Scythe 2: Proceedings and Bulletin of the International Data Farming Community*. 26-29, Montery, CA, USA. Naval Postgraduate School.

Von Frisch, K. 1974. Decoding the Language of the Bee. *Science*. 185(4152): 663-668.

Wong, A.C.H., C. L. Chua, Y. K. Lim, S. C. Kang, C. L. J. Teo, T. Lampe, P. Hingston, and B. Abbott. 2007. Applying Automated Red Teaming in a Maritime Scenario. In *Scythe 3: Proceedings and Bulletin of the International Data Farming Community*. 3–5, Montery, CA, USA. Naval Postgraduate School.

Wong, L.P., C. Y. Puan, M. Y. H. Low and C. S. Chong. 2008. Bee Colony Optimization Algorithm with Big Valley Landscape Exploitation for Job Shop Scheduling Problems. In *Proceedings of the 2008 Winter Simulation Conference*, eds. S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, 2050-2058. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Wong, L.P., M. Y. H. Low, and C. S. Chong. 2008. A Bee Colony Optimization Algorithm for Traveling Salesman Problem. In *Proceedings of the 2nd Asia Modelling Symposium (AMS 2008)*. 818-823, Kuala Lumpur, Malaysia.

Zitzler, E., K. Deb and L. Thiele. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*. 8(2): 173-195.

Zitzler, E., M. Laumanns and L. Thiele. 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In *Proceedings of EUROGEN 2001*, Athens, Greece, September 2001.

AUTHOR BIOGRAPHIES

MALCOLM YOKE HEAN LOW is currently an Assistant Professor in the School of Computer Engineering at the Nanyang Technological University (NTU), Singapore. Prior to this, he was with the Singapore Institute of Manufacturing Technology, Singapore (SIMTech). He received his Bachelor and Master of Applied Science in Computer Engineering from NTU in 1997 and 1999 respectively. He was awarded a Gintic (now SIMTech) Postgraduate Scholarship in 1999. In 2002, he received his D.Phil. degree in Computer Science from Oxford University. His current research interest is in the application of parallel and distributed computing for the modeling, simulation, analysis and optimization of complex systems. His email address is <yhlow@ntu.edu.sg>.

MAHINTHAN CHANDRAMOHAN is currently a second year undergraduate student in the School of Computer Engineering at the Nanyang Technological University of Singapore. His areas of interests include evolutionary computation and simulation modeling and analysis. His e-mail address is <chan0415@ntu.edu.sg>.

CHWEE SENG CHOO is currently a Principal Member of Technical Staff in Operations Research Laboratory (ORL), DSO National Laboratories, Singapore. He received his B.S in Physics from the National University of Singapore (NUS) in 1992, and his M.S. in Operations Research from Stanford University in 1997. His areas of interests include Combat Modelling, Simulation and Analysis, Experimental Designs, Evolutionary Computation and Data Farming. His e-mail address is <cchweese@dso.org.sg>.