2011

High-dimensional Objective-based Data Farming

Zeng Fanchao

James Decraene

Malcolm Low

Wentong Cai

Suiping Zhou

See next page for additional authors

10.1109/CISDA.2011.5945942

This article was originally published as: Fanchao, Z., Decraene, J., Low, M., Cai, W., Zhou, S., & Hingston, P. F. (2011). High-dimensional objective-based data farming. Paper presented at the IEEE Symposium on Computational Intelligence for Security and Defense Applications. IEEE. Paris, France. Original article available here

Authors
Zeng Fanchao, James Decraene, Malcolm Low, Wentong Cai, Suiping Zhou, and Philip F. Hingston

High-dimensional Objective-based Data Farming

Fanchao Zeng, James Decraene,
Malcolm Yoke Hean Low,
Cai Wentong
School of Computer Engineering
Nanyang Technological University, Singapore
Email: {fczeng,jdecraene,yhlow,aswtcai}
@ntu.edu.sg

Philip Hingston
School of Computer and Security Science
Edith Cowan University, Australia
Email: p.hingston@ecu.edu.au

Suiping Zhou
School of Computing
Teesside University, United Kingdom
Email: S.Zhou@tees.ac.uk

***Abstract*—In objective-based data farming, decision variables of the Red Team are evolved using evolutionary algorithms such that a series of rigorous Red Team strategies can be generated to assess the Blue Team's operational tactics. Typically, less than 10 decision variables (out of 1000+) are selected by subject matter experts (SMEs) based on their past experience and intuition. While this approach can significantly improve the computing efficiency of the data farming process, it limits the chance of discovering "surprises" and moreover, data farming may be used only to verify SMEs' assumptions. A straightforward solution is simply to evolve all Red Team parameters without any SME involvement. This modification significantly increases the search space and therefore we refer to it as high-dimensional objective-based data farming (HD-OBDF). The potential benefits of HD-OBDF include: possible better performance and information about more important decision variables. In this paper, several state-of-the-art multi-objective evolutionary algorithms are applied in HD-OBDF to assess their suitability in terms of convergence speed and Pareto efficiency. Following that, we propose two approaches to identify dominant/key evolvable parameters in HD-OBDF - decision variable coverage and diversity spread.**

I. Introduction

Data farming is an iterative experimental process which relies on the repeated execution of stochastic simulation models to expose major portions of the problem landscape [4]. One critical aspect of data farming is that it can generate a wide range of possible outcomes. For instance, a simulation model with only 5 parameters each of which taking on one of 100 values, can produce 10^{10} combinations. Typically, the data farming process starts with a trial and error selection of evolvable parameters in which subject matter experts (SMEs), modelers, analysts and decision-makers screen the parameters and identify important ones. Then, using High Throughput Computing (HTC) facilities, numerous simulation executions are conducted. Finally, the analyses are conducted by human experts which may be assisted by computer tools to gain insights into outliers, nonlinearities and intangibles. The inherent steps of this iterative process are repeated until sufficient insights to a problem are gained.

Objective-based data farming (OBDF) is a variant of data farming. In OBDF, decision variables are evolved using evolutionary algorithms (EAs) such that a series of rigorous Red Team strategies can be generated to assess the Blue Team's operational tactics. The strategies that perform exceedingly well against Blue's operational tactics are retained. These retained strategies can provide SMEs with alternative views regarding the various vulnerabilities in the Blue's operational tactics (Fig. 1). Complex Agent System Evolver (CASE), inspired by Automated Red Teaming (ART) [5] is an evolutionary and modular framework to automate the process of OBDF. It was constructed in a modular manner to accommodate with ease the user's specific requirements (e.g., use of different simulation engines or evolutionary algorithms). In CASE, the modelling and analysis steps of data farming can be carried out dynamically such that the manually intensive involvement of SMEs can be relieved.

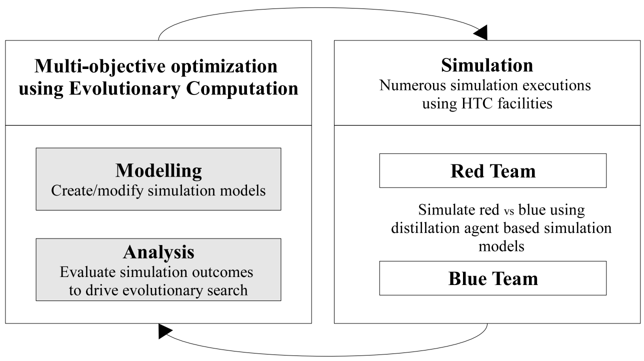

Fig. 1. Schematic representation of computational red teaming.

However, CASE also relies on the domain knowledge of SMEs to select evolvable parameters based on their past experience and intuitions. These parameters often focus on certain aspects of the scenarios, with less than 10 parameters selected (whereas a simple Map Aware Non-Uniform Automata (MANA) model [10] may contain 1000+ parameters). This limits the chance of discovering "surprises" and moreover, data farming may be used only to verify SME assumptions. A straightforward solution to this issue is to simply evolve all Red Team parameters using EAs without any SME involvement. This modification significantly increases the search space and therefore we refer to it as high-dimension objective based data farming (HD-OBDF). The potential benefits of HD-OBDF include: 1. Possible better performance: By evolving a larger set of decision variables, more sophisticated Red Team attacking strategies can be explored. Decision variables which appear to be uncorrelated may derive surprisingly effective tactics and even obtain better performance. 2. Information about more important decision variables: By investigating the spread of the decision variables, more important decision variables can be identified and further experiments can be carried out to exploit the effects of these key factors.

In this paper, to assess the suitability of different multi-objective evolutionary algorithms (MOEAs), several state-of-the-art MOEAs are applied in HD-OBDF using an anchorage protection scenario simulation model. These MOEAs include Strength Pareto Evolutionary Algorithm 2 (SPEA2) [17], Non-dominated Sorting Genetic Algorithm II (NSGAII) [8], Hypervolume Estimation Algorithm for MO Optimization (HYPE) [2] and Multi-objective Differential Evolution (MODE). In addition, the strategies, obtained using normal OBDF and HD-OBDF, are compared and the issue of local optima and early convergence in HD-OBDF are discussed as well. Following that, two parameter filter approaches are discussed to identify dominant evolvable parameters in HD-OBDF.

The remainder of the paper is structured as follows: An overview of related work is first provided in Section II. Then, Section III describes the CASE framework, followed by a brief introduction on multi-objective evolutionary algorithms. The experiments comparing MOEAs in HD-OBDF and discussion on different parameter filter approaches are documented in Section IV. Finally, Section V concludes with a summary of the paper.

II. Related Work

In the domain of objective-based data farming, example systems include: Irreducible Semi-Autonomous Adaptive Combat (ISAAC) [9], Automated Red Teaming [13], Warfare Intelligent System for Dynamics Optimization of Mission (WISDOM) [15] and Automated Red Teaming developed by DSO National Laboratories, Singapore (DSO-ART) [5].

1) ISAAC: Ilachinski [9] adopted a simple genetic algorithm using a single point crossover, mutation, elitist and truncation selection operator to identify and evolve the Red Team's behavioral parameters.

2) WISDOM: Yang et al. [15] first utilize a (1+1) Evolution Strategy (ES) and rely on the use of the linear combination of objectives approach to tackle the multi-objective optimization problems. Later, WISDOM was extended with NSGAII to improve the evolutionary dynamics as well as the range of best solutions. Their studies showed that objective-based data farming could provide better understanding of warfare.

3) DSO-ART: Automated Red Teaming (ART) is an automated process that augments Manual Red Teaming (MRT), which is a technique frequently used by the Military Operational Analysis community to uncover vulnerabilities in operational tactics. ART makes use of multi-objective evolutionary algorithms such as SPEA2 and NSGAII to effectively find a set of non-dominated solutions from a large search space. ART has been applied on several military based scenarios. Choo et al. [5] demonstrated the capability of ART using an urban operations scenario which involves the defense of an urban area controlled by the Red Team. Their work showed that ART was able to discover solutions which were useful for analysts 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. [12] 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. [14] extended the work by Sim et al. 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.

However, as mentioned earlier, these experiments focused on certain aspects of the scenarios with less than 10 evolvable parameters selected. For instance, in Yang et al. [15], ten different decision variables representing the characteristics of personalities are evolved for the Red Team in six different scenarios. In Choo et al. [5], the study focuses on how intangibles could lead Red to break Blue with 8 evolvable parameters (e.g., Red Squad Aggressiveness and Red Squad Cohesion). Sim et al. [12] evolved 5 decision variables to exploit the Red Team's behaviour (e.g., Aggressiveness and Cohesiveness). This evolvable parameter selection approach can significantly improve the computing efficiency of the OBDF process. But it limits the chance of discovering "surprises" and moreover, data farming may be used only to verify SMEs' assumptions. In this paper, HD-OBDF is explored using the CASE framework. A flowchart and the features of the CASE framework are presented in the next section.

III. The CASE framework

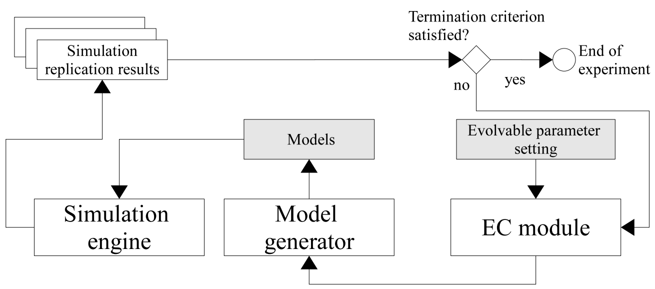

Fig. 2. Flowchart of CASE Framework.

To automate the process of OBDF, Complex Adaptive Systems Evolver (CASE) was constructed. CASE was inspired by the Automated Red Teaming (ART) [5] framework. In CASE, the modeling and analysis steps of data farming can be carried out dynamically based on EAs such that the manually intensive involvement of SMEs can be relieved. The three main components of CASE are distinguished as follows:

TABLE I
LIST OF EVALUATED MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS.

Algorithm	Ref.
Non-dominated Sorting Genetic Algorithm II (NSGAII)	[8]
Strength Pareto Evolutionary Algorithm 2 (SPEA2)	[17]
Hypervolume Estimation Algo. for MO Optimization (HYPE)	[2]
Multi-objective Differential Evolution (MODE)	*[1]

TABLE II
MANA SETTINGS FOR RED AND BLUE FORCES IN MARITIME ANCHORAGE PROTECTION SCENARIO.

Unit	Qty.	Speed	Detection range	Weapon range	Weapon hit probability
Inner-Blue patrol	3	16 knots	6 nm	2 nm	80%
Outer-Blue patrol	4	16 knots	6 nm	2 nm	50%
Green vessels	20	N/A	N/A	N/A	N/A
Red forces	5	16 knots	2 nm	2 nm (Blue) 1 nm (Green)	5% (Blue) 100% (Green)

A. The model generator

This component takes as inputs a base simulation model specified in the eXtended Markup Language (XML) and a set of model specification text files. According to these inputs, novel XML simulation models are generated and sent to the simulation engine for evaluation.

B. The simulation engine

The set of XML simulation models is received and executed by the stochastic simulation engine. Each simulation model is replicated a number of times to account for statistical fluctuations. A set of result files detailing the outcomes of the simulations (in the form of numerical values for instance) are generated. These measurements are used to evaluate the generated models, i.e., these figures are the fitness (or cost) values utilized by the EA.

C. The multi-objective evolutionary algorithm

EAs are stochastic population-based search techniques inspired by real phenomena occurring in nature. EAs simulate natural evolution through the variation (i.e., chromosomal recombinations and gene mutations) of genetic material and selection of fittest (from a phenotypic viewpoint) candidate solutions. A wide variety of EAs has been developed and they differ from each other on the specification and implementation of common properties: problem representation, variation and selection of candidate solutions. In contrast with single objective EAs (using linear combination techniques such as the weighted sum of objectives), Pareto-based multi-objective EAs (MOEAs) address explicitly multiple (and potentially conflicting) objectives. Table I lists the MOEAs (which are representative of the state of the art in the area) evaluated in this comparative study.

The key algorithmic differences between these algorithms depend in the specification of the selection schemes which determine the most promising candidate solutions to be conserved/evolved during the search. Several computational techniques exist to select the most "promising" candidate solutions whilst considering the above conflicting objectives: NSGAII and MODE utilize the "crowding distance" (i.e., an estimation of the density) of the solution points. HYPE employs the hypervolume of the solution space dominated by the Pareto set approximation (these algorithms employ differing implementations to approximate the hypervolume indicator value). The variation (e.g., recombination and mutation) of solutions for NSGAII, SPEA2 and HYPE are conducted using the simulated binary crossover (SBX) operator [6] whereas MODE utilizes weighted difference vectors. The PISA [3] implementations of the above algorithms are utilized to assist this research except for MODE[1], which was implemented by the authors).

In CASE, the set of simulation results and associated model specification files are received by the MOEAs, which, in turn, process the results and produce a new "generation" of model specification files. The generation of these new model specifications is driven by the user-specified (multi)objectives (e.g., maximize Blue casualties and minimize Red casualties). The algorithm iteratively generates models which would incrementally, through the evolutionary search, best exhibit the desired outcome behavior. The model specification files are sent back to the model generator; this completes the search iteration. The above components are depicted in Figure 2 which presents the flowchart of a CASE experiment.

IV. EXPERIMENTS

In this section, a maritime anchorage protection scenario is examined. Previous OBDF using this scenario conducted in [12], [14] evolved less than 10 decision parameters with parameter space pre-defined by SMEs. In this study, the MOEAs evolve a much wider range of the Red Team parameters. In the first case study, CASE evolves waypoint positions (33 parameters) and personality weightings for the Red Team (27 parameters). Hence, a total of 60 evolvable parameters are chosen. In the second case study, for each Red vessel, the number of intermediate waypoints (up to five) is evolved. This results in a total of 80 evolvable parameters with an even larger search space. Firstly, four MOEAs (NSGAII, SPEA2, HYPE and MODE) are applied in these two case studies to assess their performance in HD-OBDF. Secondly, two parameter filter approaches (decision variables coverage and diversity spread) are discussed to identify dominant evolvable parameters in HD-OBDF.

A. Maritime Anchorage Protection Scenario

In this scenario, a Blue Team (composed of 7 vessels) conducts patrols to protect an anchorage (in which 20 Green

[1]MODE is partially based on a DE variant proposed in [1]. MODE introduces an external archive to promote the effect of elitism. Moreover, unlike other multi-objective differential evolution algorithms where the individual solution sets are selected to generate offspring solution sets from the current population, MODE selects them from the archive (containing the best candidate solutions found so far).

TABLE III
EVOLVABLE RED PARAMETERS IN MARITIME ANCHORAGE PROTECTION SCENARIO.

Red property	Abbreviation	Min	Max
Way Point Position			
Team 1 initial position (x,y)	RedHX, RedHY	(0,0)	(399,39)
Team 2 initial position (x,y)	RedHX, RedHY	(0,160)	(399,199)
Intermediate waypoints (x,y)	RedMX, RedMY	(0,40)	(399,159)
Team 1 final position (x,y)	RedFX, RedFY	(0,160)	(399,199)
Team 2 final position (x,y)	RedFX, RedFY	(0,0)	(399,39)
Personality Weightings	RedAggression	-100	100
	RedCohesiveness	-100	100
	RedDetermination	-100	100
	AliveNeutrals	-100	100
	AliveEnemy	-100	100
	EasyTerrain	-100	100
	Centre	-100	100
	etc	-100	100

commercial vessels are anchored) against threats. Red forces (5 vessels) attempt to break Blue's defense strategy and inflict damage to anchored vessels. The aim of the study is to discover Red's strategies that are able to breach Blue's defensive tactic. Figure 3 depicts the scenario which was modeled using the agent based simulation platform MANA. The Blue patrolling strategy is composed of two layers: an outer (with respect to the anchorage area, 30 by 10 nm) and inner patrol. The outer patrol consists of four smaller but faster boats. They provide the first layer of defense whereas the larger and heavily armored ships inside the anchorage are the second defensive layer. Table II summarizes the model properties.

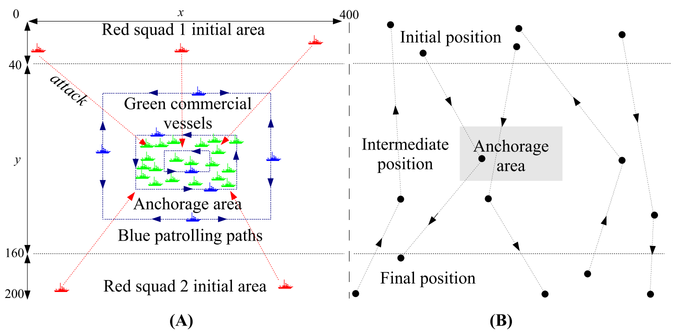

Fig. 3. MANA model of the maritime anchorage protection scenario. (A): Three of the Red vessels (squad 1) are set up to initiate their attack from the north while the remaining two attack (squad 2) from the south. The initial positions of Blue vessels are fixed. In contrast, the 20 Green commercial vessels' initial positions are randomly generated within the anchorage area at each MANA execution. (B): Example Red trajectories.

In CASE, each candidate solution is represented by a vector of real values defining the different evolvable Red behavioral parameters. The home and final positions together with the intermediate waypoint define the trajectory of each distinct Red vessel. Three of the Red craft (Team 1) were set up to initiate their attack from the north while the remaining two attack (Team 2) from the south. This allows Red to perform multi-directional attack on the anchorage. In addition, the final positions of the Red craft are constrained to the opposite region (with respect to the initial area) to simulate

TABLE IV
SUMMARY OF SIMULATION PROPERTIES.

Case study	Number of dimensions	Search space size
1	60	8.62×10^{133}
2	80	2.07×10^{187}

escapes from the anchorage following successful attacks. Personality weightings contain several personality properties (e.g., attraction or repulsion to enemies, enemy threat, ideal enemy, friends, neutrals, concealment) which can be varied between -100 and 100. The weighting value corresponds to the degree of attraction or repulsion. Previous studies [5], [12], [14] focus on the Red Team's attacking behaviors (e.g., Aggressiveness, Cohesiveness, Attrition and Determination) and set other behavior parameters (e.g., Easy Going, Cover, Alternative Waypoint, Neutrals and Unknowns) to be a neutral value of zero. In the maritime anchorage protection scenario, there are 27 personality weighting parameters available for the Red team and only 7 are listed in Table III. Two case studies are devised based on this model. The first case study tries to exploit the Red team's personality properties by evolving all Red team personality weighting parameters. In the second case study, more waypoints are added to evolve more complex attacking trajectories. The number of dimensions (evolvable parameters) and search space size for each case study are summarized in Table IV.

1) Case study one: All Red Team personality weightings are subject to evolution without any domain knowledge. This results in a total of 60 evolvable parameters selected for an extremely large search space.

2) Case study two: For each Red vessel, the number of intermediate waypoints (up to five) is evolved. This enables the evaluation of more complex trajectories. To promote the emergence of more advanced Red trajectories (involving more than a single intermediate waypoint), the anchorage area is expanded (doubled) and simulation time limit increased (from 250 to 1200 discrete time steps). This results in a total of 80 evolvable parameters with an even larger search space.

B. MOEAs evaluation in HD-OBDF

Four MOEAs (NSGAII, SPEA2, HYPE and MODE) are applied in these two case studies to assess their performance in HD-OBDF. The default and most commonly-used parameter settings (as reported in the literature [11], [5]) for these algorithms are employed in the experiments as shown in Table V. Two Measures of Effectiveness (MOEs) were used to evaluate a given solution. The two MOEs are:

- Mean Red Casualty (Minimize)
- Mean Green Alive (Minimize)

Each algorithm was configured to perform a maximum of 10,000 evaluations on the scenario. The MOEs obtained for each solution is the mean value computed from the end state of 30 replications of the simulation in MANA.

TABLE V
THE SETTINGS FOR THE MOEAS.

Settings	NSGAII	SPEA2	HYPE	MODE
Population Size	100	100	100	100
Number of Generations	100	100	100	100
Crossover rate	0.9	0.9	0.9	N/A
Crossover index	20	20	20	N/A
Mutation rate	0.1	0.1	0.1	N/A
Mutation index	20	20	20	N/A

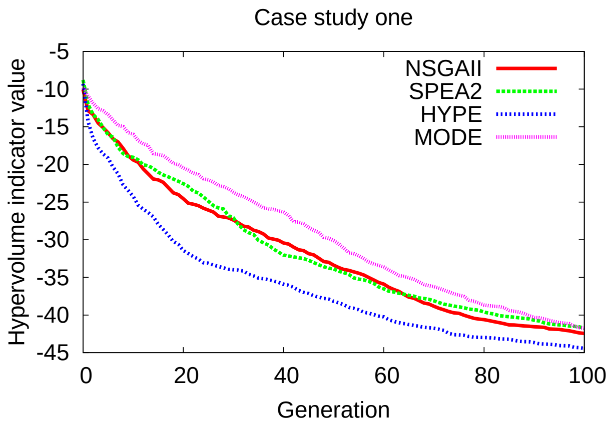

Fig. 4. The MOEAs' hypervolume dynamics over 100 generations in case study one.

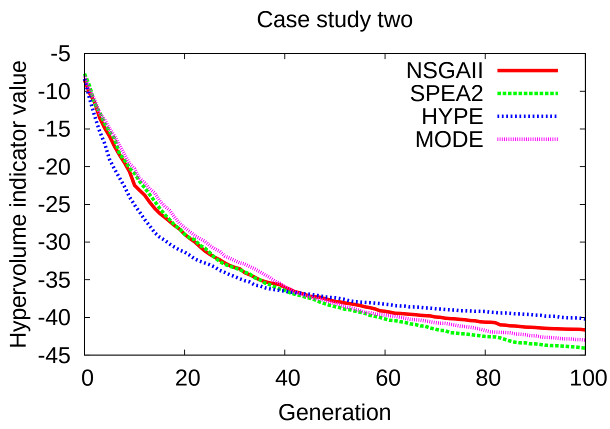

Fig. 5. The MOEAs' hypervolume dynamics over 100 generations in case study two.

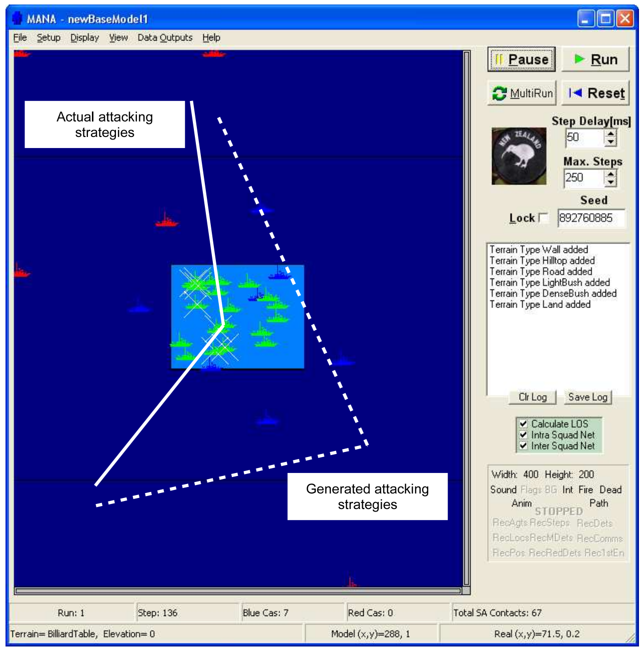

Fig. 6. Surprising attacking strategies obtained in HD-OBDF in which the Red team does not follow attacking trajectory.

There are two goals in multi-objective optimization (MOP), convergence and diversity. The hypervolume indicator considers both by measuring the volume of the dominated portion of the objective space [18]. As it possesses the highly desirable feature of strict Pareto compliance, hypervolume has been of exceptional interest in recent MOP studies. For the hypervolume indicator, if the Pareto set A dominates the Pareto set B, the hypervolume of A should be higher than that of B. In our study, the Weighted Hypervolume Indicator (WHI) package developed by Zitzler et al. [16] is utilized to compare the MOEs generated by MOEAs over 100 generations. To make the output consistent with the other indicator tools, WHI outputs negative hypervolume so a lower indicator value corresponds to a better approximation set. The MOEAs' hypervolume dynamics over 100 generations are presented in Figures 4 and 5. In terms of the final generation Pareto performance, we can observe that for case study one, HYPE achieves the best performance (NSGAII: -42.44505, SPEA2: -41.76637, HYPE: -44.42332, MODE: -41.72772) whereas in case study two, SPEA2 (-44.07501) outperforms all other MOEAs (NSGAII: -41.66205, HYPE: -40.20002, MODE: -43.01387). In terms of convergence speed, in both case studies one and two, HYPE converges much faster than the other MOEAs. As we can observe, before generation 40, there is a clear advantage of HYPE and by using the decision space diversity running performance metric [7], we find that the decision variables of HYPE converge and stabilise at around generation 40 whereas other MOEAs reach convergence much later.

Furthermore, for HD-OBDF, many surprising attacking strategies and behaviours are observed in the final generation population. Previous OBDF experiments using the anchorage model always derive high positive waypoint personality value which effectively enforces the Red team to follow the generated attacking trajectories. In HD-OBDF, negative waypoint personality value is observed which simply means that the Red team avoids the waypoint. With the combination of high attraction to the central area and other personality settings, a surprising behaviour is obtained as demonstrated in Figure 6. And unexpectedly, the Red Team manages to kill half of the green vessels without a single red causality. This type of strange yet efficient attacking behaviour has never been observed in normal OBDF.

C. Parameter filter approaches

As mentioned earlier, selecting more decision variables to evolve cannot guarantee that better performance in HD-OBDF can be achieved. In this section, we investigate two parameter filter approaches to identify the key factors to evolve in OBDF for case study one. Since HYPE can always converge relatively faster than other MOEAs, we examine the solution set generated by HYPE at generation 20 and extract dominant/key decision variables based on their spread and these further selected decision variables are evolved from the beginning again using NSGAII which is the most commonly used benchmark MOEA.

1) Decision variables coverage: In this approach, we investigate the coverage of the decision variables. The coverage is basically the spread of decision variables over the valid range. It is calculated through dividing the range of the decision variable in the population by the valid range of that particular decision variable, (max-min)/range. In our study, 15 decision variables with high coverage are selected to run the experiment again. To compare the effect of converged decision variables, we also chose 15 decision variables from the bottom which are converged in the early generation to repeat the experiment of case study one. All other non-evolvable decision variables are set to a neutral value of zero. These decision variables are shown in Table VI and their hypervolume performance is displayed in Figure 7. Intuitively, decision variables which are irrelevant to the objective values should behave randomly throughout the evolutionary optimization process and hence, achieve a wider spread. Yet, the simulation results are exactly opposite as shown in Figure 7. The decision variables which converge relatively early are not key/dominant factors. The two sets of decision variables have a huge difference in terms of hypervolume (wide spread: -29.35675, narrow spread: -22.42101). Finally, both wide and narrow spread decision variables (30 decision variables) are subject to evolution using NSGAII again and a hypervolume value of -40.7704 is obtained. This Pareto-performance is slightly worse than the hypervolume (NSGAII: -42.44505) achieved by evolving 60 decision variables.

2) Diversity Approach: In the diversity approach, instead of simply looking at the coverage, we try to explore the decision variables' diversity spread. The diversity performance is derived as follows:

Given the minimal and maximal boundary values, the hyperplane is thus divided into a number of grid cells (population size divided by the number of objectives). The diversity performance metric is based on whether each cell contains a solution point or not. The best diversity performance is achieved if all cells contain at least a solution point. The steps to calculate the diversity are as follows.

- Step 1: Calculate diversity array.
  The number of integer variables in the diversity array is equal to the number of cells in the hyperplane. Each variable in the diversity array corresponds to one particular cell i. The value $h(i)$ of the i^{th} elements is derived using Equation 1.

$$h(i) = \begin{cases} 1 & \text{, if cell i contains a representative point} \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

- Step 2: Assign a value, m() to each cell i depending on its neighboring cells' $h()$ values in the diversity array. The value of the i^{th} cell is calculated as shown in Table VII.
  For example let us consider the grid patterns p_1=010 (i.e., $h(i-1)=0$, $h(i)=1$ and $h(i+1)=0$ and p_2=101. According to Table VII, we obtain m(p_1) = m(p_2) = 0.75 which represents a good periodic spread pattern. Whereas if we consider p_3=110, we obtain m(p_3)= 0.67 meaning that p_3 covers a smaller spread.
- Step 3: For each dimension in the decision and objective space, calculate the diversity measure d_m by averaging the $m()$ values.

$$d_m = \frac{\sum_i^{\text{number of grids}} m(h(i-1), h(i), h(i+1))}{\text{Number of Grids}} \tag{2}$$

TABLE VI
WIDE AND NARROW SPREAD DECISION VARIABLES.

Wide spread variables			Narrow spread variables		
Variables	min	max	Variables	min	max
Red5HX	0	399	Red Determination	20	100
Red3MX1	0	399	inorgfriends	-100	100
Red1MY1	40	159	inorgunknowns	-100	100
inorgthreat3	-100	100	Red4FY	0	39
enideal	-100	100	orgotherfriend	-100	100
Red1FX	0	399	nextwaypoint	-100	100
aliveEnemy	-100	100	Red5MY1	40	159
Red4HX	0	399	easyterrain	-100	100
injuredfriends	-100	100	Red2HX	0	399
orgsquadfriend	-100	100	Red4MY1	40	159
enthreat1high	-100	100	Red3MY1	40	159
Red2HY	0	39	Red2MX1	0	399
centre	-100	100	Red3FX	0	399
Red5FX	0	399	Red1HY	0	39
Red4HY	160	199	aliveFriends	-100	100

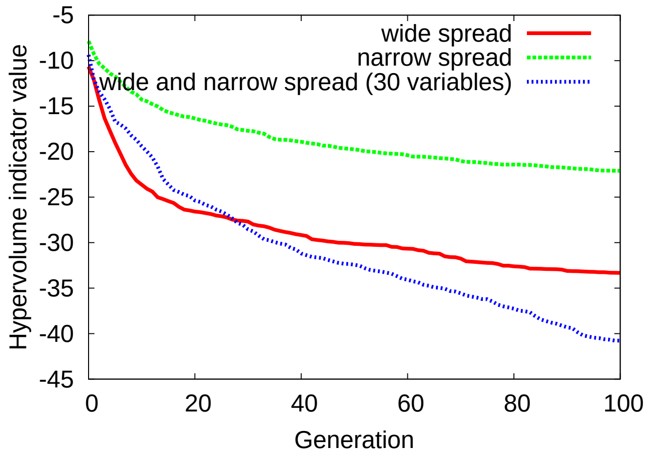

Fig. 7. Hypervolume performance of case study one using wide and narrow spread decision variables.

TABLE VII
MAPPING TABLE TO ASSIGN A VALUE TO $m()$. (ADAPTED FROM [7])

$h(i-1)$	$h(i)$	$h(i+1)$	$m(h(i-1), h(i), h(i+1))$
0	0	0	0.00
0	0	1	0.50
1	0	0	0.50
0	1	1	0.67
1	1	0	0.67
0	1	0	0.75
1	0	1	0.75
1	1	1	1.00

To illustrate the procedure to calculate the diversity measure, an example is presented in Figure 8.

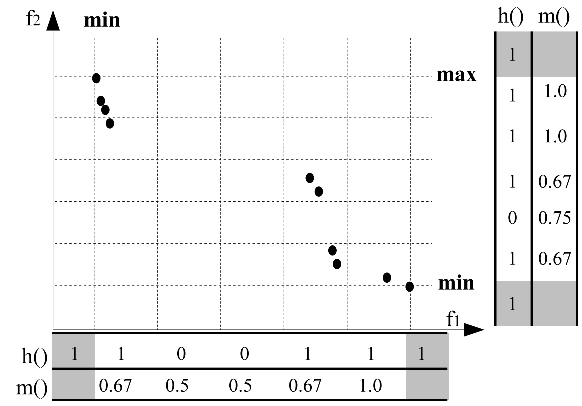

Fig. 8. Example of computing the diversity metric

In this example, a two-dimension (f_1 and f_2) diversity measure is examined. The solution points are marked as points. Suppose the population size is 10, we divide the range of f_1 and f_2 values into 10/2 = 5 grids. Then, for each grid, the value of $h()$ is calculated based on whether the grid contains a representative solution point or not. Then, the value of $m()$ and the diversity measures are calculated based on a sliding window containing three consecutive grids. The $h()$ values of the imaginary boundary grids are always 1 as shown in the shaded grids. Firstly, the f_2 = 0 plane is used as the reference plane.

$$d_m(f_1) = \frac{0.67 + 0.50 + 0.50 + 0.67 + 1}{5} = 0.668$$

Then, the f_1 = 0 plane is selected as the reference plane.

$$d_m(f_2) = \frac{1 + 1 + 0.67 + 0.75 + 0.67}{5} = 0.818$$

So clearly decision variable f_2 has a better diversity spread than decision variable f_1.

Based on the diversity metric described above, the decision variables at generation 20 are ranked. The top and bottom 15 decision variables are chosen to run further experiments using case study one as listed in Table VIII. All other non-evolvable decision variables are set to a neutral value of zero. Their hypervolume dynamics over 100 generation are presented in Figure 9. As shown, the evolution using low diversity decision variables has a much better initial performance than the one with high diversity; however, as the evolution progresses, the evolutionary process using high diversity decision variables produces better hypervolume performance (low diversity: -33.56768, high diversity: -36.13803). But this time, the difference between the two is not as obvious as the one in the decision coverage approach. Then, we combine both high and low diversity decision variables and form a new set of 30 decision variables. By evolving these 30 decision variables, the experiment is repeated. As demonstrated in Figure 10, the set of experiments evolve fewer (30) decision variables, yet achieves better results (Diversity approach (30 variables): -42.81468, NSGAII (60 variables):-42.44505, Variable coverage (30 variables): -40.77040).

TABLE VIII
HIGH AND LOW DIVERSITY DECISION VARIABLES.

High diversity variables			Low diversity variables		
Variables	min	max	Variables	min	max
Red4MX1	0	399	cover	-100	100
orgknowns	-100	100	Red2HY	0	39
injuredfriends	-100	100	Red5MY1	40	159
Red1MX1	0	399	inorgfriends	-100	100
Red1HY	0	39	Red3FX	0	399
Red4HX	0	399	Red5MX1	0	399
concealment	-100	100	Red1HX	0	399
Red5FY	0	39	nextwaypoint	-100	100
Red2MX1	0	399	enideal	-100	100
Red4MY1	40	159	Red2HX	0	399
inorgthreat2	-100	100	Red5HY	160	199
Red Cohesiveness	-100	100	aliveneutrals	-100	100
Red3MX1	0	399	Red4FX	0	399
Red1FY	160	199	orgneutrals	-100	100
Red4HY	160	199	aliveEnemy	-100	100

Hence, it seems that the diversity of decision variables is a more promising approach to identify dominant decision variables. The high diversity decision variables can facilitate exploration of the search space whereas the low diversity decision variables can exploit the Pareto front and further improve the quality of the solution sets. In addition, our intuitive guess that irrelevant decision variables should behave randomly whereas key/dominant one should converge early is disproved in HD-OBDF.

V. CONCLUSION

In this paper, the first preliminary work on high-dimensional data farming is explored. Firstly, to assess the suitability of different MOEAs, several state of the art MOEAs (NSGAII, SPEA2, HYPE and MODE) are applied in HD-OBDF using an anchorage protection scenario simulation model. In terms of final generation Pareto performance, we can observe that HYPE performs the best in case study one whereas in case study two, SPEA2 outperforms all other MOEAs. In terms of convergence speed, in both case studies one and two, HYPE converges much faster than other MOEAs. This feature is significantly desirable for very complex stochastic models which require long time to run and multiple replications to consolidate the data. By evolving a larger set of decision variables, more complex attacking strategies and behaviour can be explored

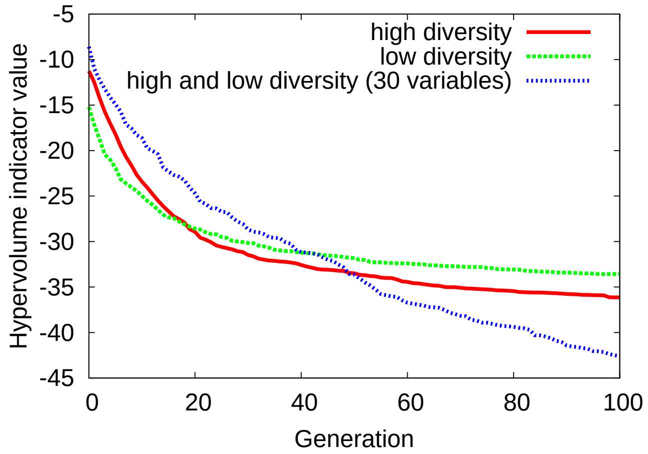

Fig. 9. Hypervolume performance of case study one using high and low diversity spread decision variables.

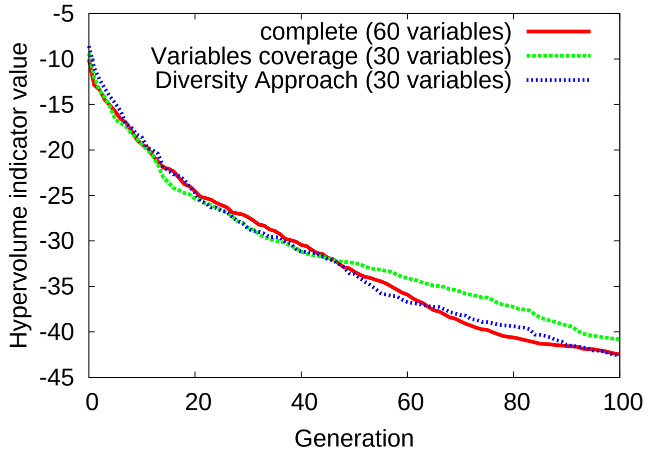

Fig. 10. Hypervolume performance of case study one using complete and selected decision variables.

and decision variables which appear to be uncorrelated may derive surprisingly effective tactics and even obtain better performance. Secondly, by investigating the spread of the decision variables, more important decision variables can be identified. Intuitively, decision variables which are irrelevant to the objective values should behave randomly throughout the evolutionary optimization process and hence, achieve a wider spread. Yet, the simulation results show the exact opposite. Both evolutions of wide spread and high diversity decision variables obtain much better performance than the experiments evolving the early converged decision variables. Two parameter filter approaches are presented in this work. Based on our preliminary experiments, the diversity spread approach is quite promising. The high diversity decision variables can facilitate exploration of the search space whereas the low diversity decision variables can exploit the Pareto front and further improve the quality of the solution sets. By selecting both high diversity and low diversity decision variables, it manages to achieve better hypervolume performance than evolving a complete set of decision variables.

ACKNOWLEDGMENT

We thank the Defence Research and Technology Office, Ministry of Defence, Singapore, for sponsoring the Evolutionary Computing Based Methodologies for Modeling, Simulation and Analysis project which is part of the Defense Innovative Research Programme FY08.

REFERENCES

[1] Hussein A. Abbass, Ruhul Sarker, and Charles Newton. PDE: A Pareto Frontier Differential Evolution Approach for Multiobjective Optimization Problems. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 971–978, 2001.

[2] J. Bader and E. Zitzler. HypE: Fast Hypervolume-based Multiobjective Search using Monte Carlo Sampling. *TIK Report Number 286*, 286.

[3] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA: A Platform and Programming Language Independent Interface for Search Algorithms. In *Evolutionary Multi-criterion Optimization*, pages 494–508. Springer, 2003.

[4] A. Brandstein and G. Horne. Data Farming: A Meta-technique for Research in the 21st Century. *Maneuver Warfare Science*, 1998.

[5] C.S. Choo, C.L. Chua, and S.H.V. Tay. Automated Red Teaming: A Proposed Framework for Military Application. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, page 1942. ACM, 2007.

[6] K. Deb and R.B. Agrawal. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9(2):115–148, 1995.

[7] K. Deb and S. Jain. Running Performance Metrics for Evolutionary Multi-objective Optimization. *Kanpur Genetic Algorithms Laboratory Report*, 2002004, 2002.

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

[9] A. Ilachinski. Irreducible Semi-autonomous Adaptive Combat (isaac): An Artificial-life Approach to Land Combat. *Military Operations Research*, 5(3):29–46, 2000.

[10] M. Lauren and R. Stephen. Map-aware Non-uniform Automata (MANA)-A New Zealand Approach to Scenario Modelling. *Journal of Battlefield Technology*, 5:27–31, 2002.

[11] K.H. Liang and K.M. Wang. Using Simulation and Evolutionary Algorithms to Evaluate the Design of Mix Strategies of Decoy and Jammers in Anti-torpedo Tactics. In *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pages 1299–1306, 2006.

[12] WC Sim, CS Choo, EC Ng, F. Martinez-Tiburcio, and ER Toledo-Ramirez. K. Lin. 2007. Applying Automated Red Teaming in a Maritime Scenario.

[13] S.C. Upton, S.K. Johnson, and M. McDonald. Breaking Blue: Automated Red Teaming Using Evolvable Simulations. In *Proceedings of Genetic and Evolutionary Computation Conference 2004*, 2004.

[14] CH Wong, WC Sim, CL Chua, YK Lim, SC Kang, C.L.J. Teo, T. Lampe, P. Hingston, and B. Abbott. Applying Automated Red Teaming in a Maritime Scenario. *Scythe Issue*, 1(3):3–5, 2007.

[15] A. Yang, H.A. Abbass, and R. Sarker. Characterizing Warfare in Red Teaming. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(2):268–285, 2006.

[16] E. Zitzler, D. Brockhoff, and L. Thiele. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators via Weighted Integration. In *Evolutionary Multi-Criterion Optimization*, pages 862–876. Springer, 2007.

[17] E. Zitzler, M. Laumanns, L. Thiele, et al. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In *EUROGEN 2001*, pages 95–100. International Center for Numerical Methods in Engineering, 2001.

[18] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257, 1999.