Evolving Agent-Based Model Structures using Variable-Length Genomes

James Decraene, Mahinthan Chandramohan, Fanchao Zeng, Malcolm Yoke Hean Low, and Wentong Cai

School of Computer Engineering,
Nanyang Technological University, Singapore 609479.
{jdecraene, chan0415, fczeng, yhlow, aswtcai}@ntu.edu.sg,
WWW home page: http://pdcc.ntu.edu.sg/EVOSIM/

Abstract. We present a novel evolutionary computation approach to optimize agent based models using a variable-length genome representation. This evolutionary optimization technique is applied to Computational Red Teaming (CRT). CRT is a vulnerability assessment tool which was originally proposed by the military operations research community to automatically uncover critical weaknesses of operational plans. Using this agent-based simulation approach, defence analysts may subsequently examine and resolve the identified loopholes. In CRT experiments, agent-based models of simplified military scenarios are repeatedly and automatically generated, varied and executed. To date, CRT studies have used fixed-length genome representation where only a fixed set of agent behavioural parameters was evolved. This may prevent the generation of potentially more optimized/interesting solutions. To address this issue, we introduce the hybrid variable-length crossover to evolve the structure of agent-based models. A maritime anchorage protection scenario is examined in which the number of waypoints composing the vessel's route is subjected to evolution. The experimental results demonstrate the effectiveness of our proposed method and suggest promising research avenues in complex agent-based model optimization.

Keywords: Agent-based simulation, multi-objective optimization, variable-length genome

1 Introduction

Computational Red Teaming is an agent-based simulation method which aims at identifying the critical weaknesses of military operational plans [16, 4]. A bottom-up/agent-based approach is thus adopted to analyse the complex dynamics that may emerge in combat systems. In CRT experiments, many agent-based model variants are executed/evaluated where two teams (a defensive "Blue" and belligerent "Red") are opposed using different tactical plans (as defined in the agent-based model specifications). The modelling and analysis of these tactical plans are automated and are conducted using evolutionary algorithms. The objectives of the evolutionary algorithms are, for instance, to generate Red tactical plans to best defeat Blue.

Through the analysis of optimized agent-based simulation models, one may identify Red tactical plans which may pose serious threats. Following on from this, defence analysts may attempt to resolve the operational weaknesses exposed through CRT. To our knowledge most CRT studies (see Section 2.2 for a brief survey) have focused on the examination of the agents' behaviour (e.g., aggressiveness, cohesiveness, determination, etc.). Such properties were subjected to evolutionary optimization. In these studies, the set of "evolvable model parameters" was fixed and commonly included less than 20 behavioural parameters.

We argue that such studies are limited when considering the optimization of *particular* military operations. Indeed, one may be interested in examining/generating complex courses of actions which cannot be encoded/evolved using a fixed set of behavioural parameter values. For instance, we may desire to optimize the agents' operational route where the number of waypoints may vary. Another example is the optimization of squad composition where both the number of agents and associated profile (e.g. engineer, infantry, sniper, etc) could be varied.

When considering the traditional fixed-length genome approach, one has to pre-determine and fix the number of, e.g., waypoints or agents. If this parameter is set too low, then this would prevent the emergence of optimal operational plans as the evaluated solutions are not "complex" enough. On the contrary, if such parameters are set too high, then this unnecessarily increases the complexity (through augmenting the search space dimensionality) of the search process and may prevent, as well, the finding of optimal solutions. Our proposed method attempts to deal with the optimization of such operational plans where a fixed-length genome approach may lead to optimality issues.

Novel techniques are required where additional simulation model properties, including the simulation model structure, are to be dynamically varied (i.e. added/removed) and evaluated. To extend and potentially enhance the CRT methodology, we investigate the evolution of agent-based model structures using variable-length genomes (through introducing a novel evolutionary computation technique coined the hybrid variable length crossover), in which additional simulation model properties (e.g., the model or distinct agent's structure) can be subjected to evolution.

To assist this research, we utilize a modular evolutionary framework coined CASE (complex adaptive systems evolver). Multi-objective evolutionary computation techniques are utilized to optimize the agent-based models.

Background material on agent-based models for military applications and CRT are first presented. A survey on variable length genome techniques for evolutionary algorithms follows. The CASE framework is then detailed. Experiments, using CASE and the agent-based platform MANA, are then conducted to evaluate the application of variable length genomes for the evolution/optimization of agent-based model structures. The experiments consider a simplified CRT maritime anchorage protection scenario. Finally, we conclude the paper and outline future research directions which may merit investigations to develop this work. This paper is a direct follow-up of the preliminary study reported in [8] where agent-based model structures were evolved using a *fixed*-length genome representation.

2 Background

We first briefly describe some agent-based simulations that have been applied to military operations research. Then the Computational Red Teaming concept is presented.

2.1 Military Agent-Based Simulations

Agent Based Simulations have recently attracted significant attention to model the intricate and non-linear dynamics of warfare. Combat is thus here conceptually regarded as a complex adaptive system which components (i.e. the battlefield, soldiers, vehicles, etc.) are modelled using a bottom-up agent-based approach. The agents' computational methods may include stochastic processes resulting in a stochastic behaviour at the system level. Examples of ABS applied to Military Decision Making include: ISAAC/EINSTein [16], CROCADILE [12], WISDOM [26], MANA [18] and Pythagoras [1]. A review of ABS applied to various military applications is provided by Cioppa et al [6].

These systems have been specifically devised to simulate defence related scenarios in which the properties of the environment and the Red/Blue teams may be specified [20]. The level of representation/abstraction (e.g., number of spatial dimensions, range of agents' properties, type of vehicles, etc.) varies among these ABS systems. Although the level of accuracy in representing real world environments/individuals may not faithfully reflect reality, it is argued that such ABS models account for the key features (e.g., local interactions between agents) necessary to exhibit complex emerging phenomena/behaviour at the system level which are typical of real battlefields [16]. Thus, these "distillation" models can expose the emerging phenomena of interest without the burden of modelling and simulating unnecessary complex features (e.g., gravity, wind, detailed physics of distinct simulated agents/weapons/vehicles, etc.). Agent-based modelling is one of the key technologies supporting Computational Red Teaming which is described in the next section.

2.2 Computational Red Teaming

Computational Red Teaming (CRT) combines agent-based simulations and evolutionary computation (EC) techniques as follows. CRT exploits EC techniques to evolve simulation models to exhibit pre-specified/desirable output behaviors (i.e., when Red defeats Blue). To date, most CRT studies have only addressed the evolution of a fixed set of agent parameters (e.g., troop clustering/cohesion, response to injured teammates, aggressiveness, stealthiness, etc.), defining the behaviour or personality of the Red team. These parameters are evolved to optimize the Red agents collective efficiency (e.g., maximize damage to target facilities) against the Blue team. Example studies include: [16, 26, 4, 19]. These studies demonstrated the promising potential of CRT systems to automatically identify the Blue team's weaknesses.

Further CRT investigations adopted a co-evolutionary approach where the set of behavioural parameter values of both teams are coevolved. This arms race approach complements the previous one by automating the analysis required to improve the Blue team's defence operational plan against the adaptive Red team. Examples of co-evolutionary CRT studies can be found in [17, 22, 5]. This approach enables one to generate operational tactics that are more efficient and robust against a larger range of scenarios. Nevertheless a trade-off exists in terms of robustness over efficiency according to the range of confronted Red tactics (i.e., the evolved tactics only yield average performances against multiple Red tactics).

The extension of one-sided to co-evolutionary CRT significantly increases the search spaces allowing for the exploration of more diverse simulation models. As the diversity of evaluated simulation models is increased, a wider range of potentially critical scenarios may be identified. Exploring more diverse scenarios enables one to devise more robust and effective defensive strategies against potential threats and adaptive adversaries. Nevertheless, the expansion of this search space is associated with a *dramatic* increase in computational cost. Also, due to this high effect on computational complexity, no Pareto-based multi-objective co-evolutionary approaches to CRT have been proposed to date. In this paper, we focus on the multi-objective structural evolution of agent-based models where only Red is evolved against Blue.

Finally, none of the above studies has attempted to evolve agent-based model structures. We here propose a novel method to dynamically vary the range of evolvable parameters through varying the candidate solutions' genome string length. In the next section, we survey some related evolutionary computation approaches which focused on variable-length genome techniques.

3 Survey of Variable Length Genome Techniques

Several studies have investigated variable-length genomes in the context of genetic algorithms. None of these schemes has been applied, to the authors' knowledge, to vary the structure of genomes which encode for agent-based model specifications.

3.1 Messy Genetic Algorithm

An early attempt addressing variable-length genomes was proposed by the Messy Genetic Algorithm (m-GA) [13]. In m-GA, the classical one-point crossover operator is replaced by the "cut" and "splice" operators. The cut operator is first applied upon each parent genome string where a locus point is selected at random on each genome, cutting each string into two strings. The splice operator is employed to rejoin the resulting four strings in a random order. The cut and splice operators were applied upon bit strings and is thus not directly applicable to the real-valued genomes used in Computational Red Teaming experiments.

This seminal work inspired the crossover techniques for variable-length genomes presented in the next sections.

3.2 The Speciation Adaptation Genetic Algorithm

The Speciation Adaptation Genetic Algorithm (SAGA) was introduced by Harvey [14]. In m-GA, strings were recombined regardless of the strings' contents. In contrast, SAGA was proposed to maximize the similarity between strings that are recombined to diminish undesirable disruptive effects that may occur when using a "blind" cut and splice method.

The similarity of the two parent genome strings is computed using the Longest Common Subsequence (LCSS) metric. The LCSS is the longest uninterrupted matching substring of gene values (alleles) found between two strings of arbitrary length. In SAGA, a random crossover point is chosen on the first parent string, then the algorithm tests every possible crossover point on the second string. For each potential crossover point, the algorithm calculates the LCSS sum on both the left and right regions of the genomes. The second parent string is cut at the crossover point with the highest LCSS score. If multiple crossover points are eligible, then one is selected at random.

3.3 Virtual Virus

Similarly to SAGA, the Virtual Virus (VIV) crossover [2] is based on the similarity between parent genome strings. In contrast with SAGA, VIV can only be applied upon similar sequences.

In VIV, the probability of crossover is governed by the level of similarity between the parent genome strings. This level of similarity is determined by the number of matched alleles between parent strings within a pre-specified fixed size window. As in SAGA, a random crossover locus point is selected on one of the parent strings. VIV then compares the sequence of alleles (limited by the window size) from this selected point with all possible substrings of the same size on the other parent string. The substring position that includes the greatest number of matched alleles is then recorded. The strings are then cut within the matched substring given a similarity-based probability.

3.4 Synapsing Variable-Length Crossover

In both SAGA and VIV crossover operators, the crossover locus point was first selected in *one* of the parent strings, then a relatively similar substring was searched for in the second parent string. Thus the first selected string was utilized as a template. In contrast, the Synapsing Variable-Length Crossover (SVLC) [15] employs both parent strings as a template. The motivation is to preserve any common sequences between the parent strings, where only differences are to be exchanged during recombination.

In SVLC, the level of similarity between parent strings is computed using a variant version of the LCSS (used in SAGA). A major difference with the previous crossover techniques is that SVLC also includes mutation operators (which are individually applied on children genome strings) which may affect the genome length. These length varying mutation operators were implemented in addition to the traditional point mutation operators. Four length varying mutation operators are distinguished as follows: 1) A random sequence of alleles is inserted at a random locus point on the genome string, 2) a genome substring is selected/removed at random, 3) a substring is selected at random and duplicated at a random locus point and 4) a substring is selected at random and duplicated at the beginning or end of the genome string. Various probabilities were pre-defined for each mutation operator (most disruptive operators, such as the substring insertion, were assigned a significantly lower probability of occurring).

3.5 NeuroEvolution of Augmenting Topologies

In NeuroEvolution of Augmenting Topologies (NEAT) [23], the structural evolution of artificial neural networks was investigated. NEAT included an evolutionary scheme which accounted for a variable-length genome representation. The key idea of NEAT is evolutionary complexification where (initially simple) structures/genome strings would *incrementally* complexify (as determined by the number of network nodes/interactions) through evolution.

A benefit of NEAT is to minimize the dimensionality (number of genes) through complexification. Indeed, the evolutionary process would evaluate genome strings of higher dimensionality only if these structures yield higher fitness values. This enables NEAT to search through a minimal number of genes, significantly reducing the number of generations necessary to find competitive solutions, and ensuring that genome strings are not more complex than necessary.

A historical marking technique was implemented to identify the similarities between genome strings. This marking was also used to perform the recombinations. The mutation operators included a gene duplication method (a similar length varying mutation operator was implemented in SAGA). In contrast with the bitstring representation of genomes in the previous approaches, NEAT relies on a real-valued genome representation. Alterations of the gene values were conducted using the mutation operators and not through recombinations.

3.6 Summary

This section summarises the above techniques and attempts to identify the most promising computational techniques. The latter will be then considered and evaluated in our study on the evolution of agent-based models using variables length genomes.

m-GA uses a simple cut and splice implementation which ignored any similarities between parent genome strings. SAGA and VIV accounted for similarities between the parent genome strings, however, recombinations were heavily based on the first selected parent genome string (i.e. the template) where no appropriate crossover points could be found in the second parent string. This may result in disruptive outcomes (i.e. loss of information). SVLC resolved these issues through considering both parent genome strings as templates. Moreover, SVLC introduced length-varying mutation operators. NEAT is, to some extent, similar to SVLC but took an evolutionary complexification approach where genome strings progressively increase in complexity/length through evolution.

When evolving agent-based models, the model specifications are encoded as real-valued genome strings (where each value encodes for a specific agent behavioural parameter). This conflicts with the bitstring encoding representation of m-GA/VIV/SAGA and SVLC. NEAT used a real-valued genome representation but the crossover operator does not directly modify the gene values through recombinations. In real-valued fixed-length genome evolutionary algorithms, the Simulated Binary Crossover [7] has long been established as an efficient method to recombine such genome strings encoded in continuous space. SBX will be considered into the novel hybrid method proposed in Section 5.

Moreover, the varying length mutation operators proposed in SVLC and NEAT presented promising outcomes to dynamically evolve the structure of genome strings. Such operators will be examined in our hybrid crossover method. Finally the evolutionary complexification approach of NEAT may yield potential benefits as it would avoid the exploration/evaluation of unnecessarily complex genome strings (i.e. reducing computational efforts). Nevertheless evolutionary complexification is not explored here but will be investigated in future work.

4 The Evolutionary Framework

A detailed description of the evolutionary framework, coined CASE (complex adaptive systems evolver), is provided in this section. This framework was also described and evaluated (against additional system features such as optimization under constraint, multi-objective optimization and cloud computing) in [11, 10, 9].

The CASE framework was inspired by the Automated Red Teaming framework [4] which was developed by the DSO National Laboratories of Singapore. In contrast with DSO's system (which was dedicated to examining military simulation models), we aim at providing a relatively more flexible and platform-independent system capable of evolving simulation models for a wider variety of application domains.

CASE is composed of three main components which are distinguished as follows:

1. *The model generator*: This component takes as inputs a base simulation model specified in the eXtended Markup Language and a set of model specification text files. According to these inputs, new XML simulation models are generated and sent to the simulation engine for evaluation. Thus, as currently devised, only simulation models specified in XML are supported. Moreover, the model generator may consider constraints over the evolvable

parameters (this feature is *optional*). These constraints are specified in a text file by the user. These constraints (due for instance to interactions between evolvable simulation parameters) aim at increasing the plausibility of generated simulation models (e.g., through introducing cost trade-off for specific parameter values).
2. *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 evolutionary algorithm (EA) to direct the search.
3. *The evolutionary algorithm*: The set of simulation results and associated model specification files are received by the evolutionary algorithm, which in turns, processes 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/minimize some quantitative values capturing the target system behaviour). The algorithm iteratively generates models which would incrementally, through the evolutionary search, best exhibit the desired outcome behaviour. The model specification files are sent back to the model generator; this completes the search iteration. This component is the key module responsible for the automated analysis and modelling of simulations.

The above components are depicted in Figure 1 which presents the flowchart of an example experiment. Further details about the input files settings can be found in [9]. Finally, a demonstration video of CASE can be visualized at `http://www.youtube.com/watch?v=d2Day_MEruc`.

5 Hybrid Variable Length Crossover

As discussed earlier (Section 3.6), we incorporate a number of existing evolutionary computation techniques to implement our crossover technique for variable-length genomes. We propose the hybrid variable length crossover (HVLC), which is a combination of both SBX and one point crossover (where similarities between parent genome strings are considered).

In HVLC, two distinct regions within a genome string (Fig. 2) are distinguished: A static sequence of genes (which is located at the beginning of the genome string) and dynamic sequence of genes which may vary in length.

The SBX crossover [7] was designed to recombine fixed-length genomes, thus it cannot be directly used for variable-length genomes. SBX is here utilized to recombine common substrings (which includes the static genome string region and any other sequences, with equal sizes, of "matched" genes encoding for identical model properties).

During the crossover operation, a crossover point is randomly selected (at a valid locus point, so that no structures are broken, see Fig. 3) upon the common

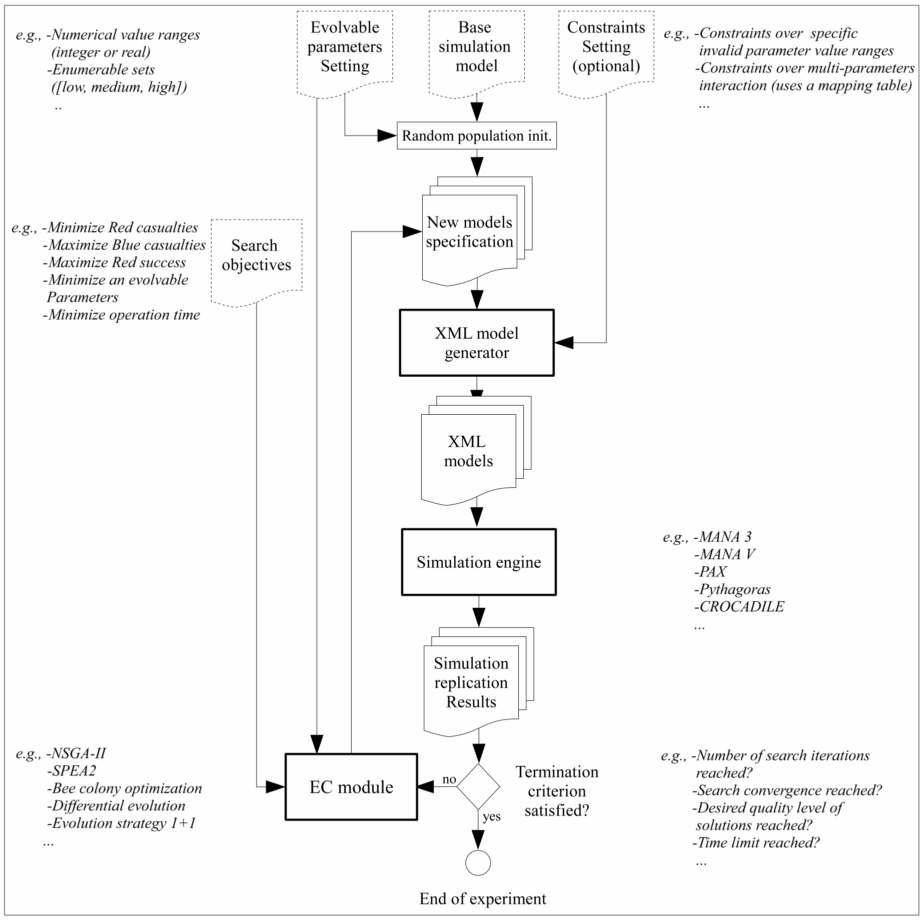
e.g., -Numerical value ranges (integer or real)
-Enumerable sets ([low, medium, high])
..
Evolvable parameters Setting
Base simulation model
Constraints Setting (optional)
e.g., -Constraints over specific invalid parameter value ranges
-Constraints over multi-parameters interaction (uses a mapping table)
...
Random population init.
e.g., -Minimize Red casualties
-Maximize Blue casualties
-Maximize Red success
-Minimize an evolvable Parameters
-Minimize operation time
Search objectives
New models specification
XML model generator
XML models
Simulation engine
e.g., -MANA 3
-MANA V
-PAX
-Pythagoras
-CROCADILE
...
Simulation replication Results
e.g., -NSGA-II
-SPEA2
-Bee colony optimization
-Differential evolution
-Evolution strategy 1+1
...
EC module
no
yes
Termination criterion satisfied?
e.g., -Number of search iterations reached?
-Search convergence reached?
-Desired quality level of solutions reached?
-Time limit reached?
...
End of experiment

Fig. 1. Flowchart of an example experiment. The dashed documents distinguish the user inputs. Using the base XML model, a population of randomly generated model variants is first created. The initial parameter values are randomly generated using a uniform distribution and are bounded by the evolvable parameters setting file provided by the user. Both the simulation engine and evolutionary computation module call external libraries and/or binaries. The XML model generator employs the Libxml library (`http://libxml.rubyforge.org`) to parse and generate XML models. The constraint setting file is utilized by the XML model generator to apply user-defined constraints over the evolvable parameters.

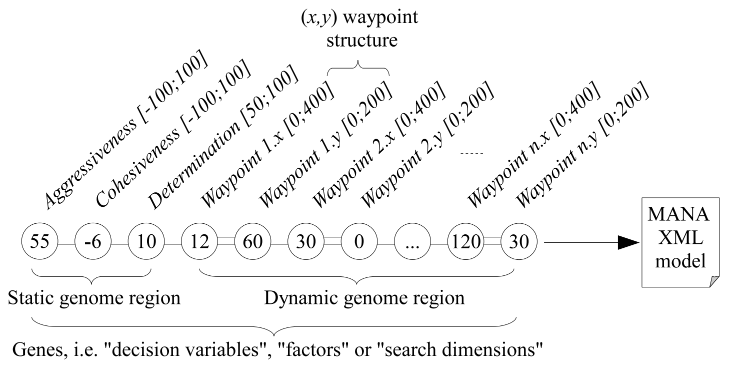

Fig. 2. Variable Genome String Representation. This genome string illustrates the encoding of simulation models utilized in the experiments reported in Section 6. In this example the waypoint structures (composed of a pair of genes encoding for spatial coordinates) are dynamically varied (removed/added) during the evolutionary search, reducing/expanding the length of the genome string. Double-linked genes indicate structures that cannot be broken through recombination.

genetic sequences of both parent genome strings. The genome strings are then cut and spliced as in m-GA. Then the SBX operator is applied over the common sequences of genes.

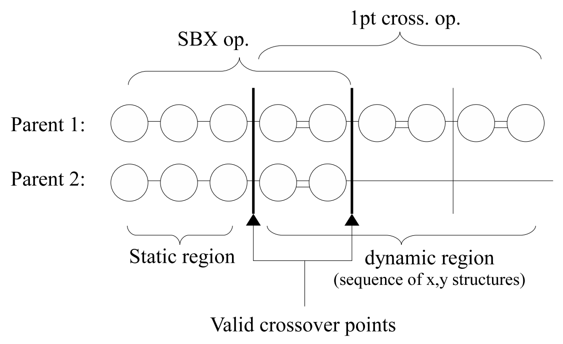

Fig. 3. Example HVLC crossover operation. Valid crossover points are distinguished to disable the recombination of genetic sequences encoding structures. In this example, waypoints are such "unbreakable" structures composed of two distinct genes encoding for x, y spatial coordinates.

To vary the genome length, we propose a length varying mutation operator in which two types of mutation can be distinguished (Fig. 4):

1. *Deletion*: A gene (or structure composed of multiple genes) is removed from the end of the genome string (reducing the genome string length).

2. *Duplication*: A gene (or structure composed of multiple genes) is selected and duplicated at the end of the genome string (increasing the genome string length).

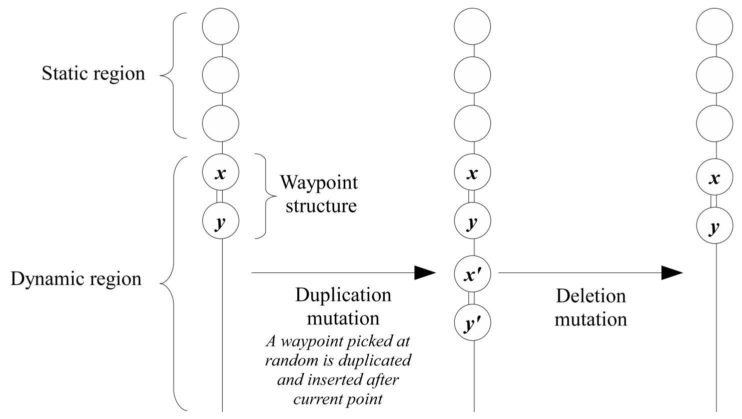

Fig. 4. Example HVLC mutation operations. In this example, the duplication and deletion mutations are applied upon waypoint structures. As a result, the entire genetic sequence representing these structures are dynamically duplicated/deleted within the genome string.

The probability of each of these operators being applied to any children genome string is 0.01. Finally, the polynomial mutation operator is also applied to each gene, introducing further variations upon the gene values.

6 Experiments

We report a series of experiments using the CASE framework and the agent-based simulation platform MANA [18]. A single case study is here examined in which the agents' structure or more specifically the number of waypoints and associated coordinates determining the agents' routes are subjected to the evolutionary process. Although examining a unique case study *considerably* limits the significance of the experimental results (the authors acknowledge that further case studies must be examined for a better appreciation of the results), we limit the current investigation to a single case study as this particular scenario was previously studied in multiple publications [24, 19, 25, 8]. This paper extends the work that has been conducted in this well-studied model scenario. Future work will include other case studies to complement our investigation and understanding on variable-length genomes for the structural evolution of simulation models.

6.1 The model

The maritime anchorage protection scenario was originally proposed by a team of defence analysts and academic researchers [24] and further developed in [19, 25, 8]. In this scenario, a Blue team (composed of 7 vessels) conducts patrols to protect an anchorage (in which 20 Green commercial vessels are anchored) against threats. Single Red vessel attempts to break Blues defence tactics and inflict damages to anchored vessels. The aim of the study is to discover Reds strategies that are able to breach through Blues defensive tactic. Fig. 5 depicts the scenario which was modelled using the ABS platform MANA.

In [8], a preliminary study on "evolvable simulation" (i.e. where the structure of the model is evolved) was examined. In this work, a fixed-length genome was employed, additional genes were introduced to control the number of waypoints to be "switched on". Thus, the maximum number of waypoints that compose the Red vessel route had to be pre-specified (this determines the genome string length). The current study extends this preliminary work through removing such "control" genes and effectively vary dynamically the genome string length.

6.2 Experimental Setting

In CASE, each candidate solution (a distinct simulation model) is represented by a vector of real values defining the different evolvable Red behavioural parameters (Table 1). As the number of decision variables increases, the search space becomes dramatically larger.

The selection scheme (based on the crowding distance and Pareto sorting) of the Non-dominated Sorting Algorithm II (NSGAII) is employed to assist the evolutionary search. This algorithm is executed using the following parameters: population size = 100, number of search iterations = 200, mutation probability = 0.1, mutation index = 20, crossover rate = 0.9 and crossover index = 20. Such parameter values for NSGAII are commonly used in the literature to solve two-objective optimization problems. The population size and number of search iterations indicate that 20,000 distinct MANA simulation models are generated and evaluated for each experimental run. Each individual simulation model is executed/replicated 30 times to account for statistical fluctuations (30 replications would approximately take 10 wallclock seconds to execute using an Intel Dual Core CPU @ 2.66GHZ).

The efficiency of the search is measured by the number of Green casualties with respect to the number of Red casualties. In other words, the search objectives are:

- To minimize the number of Green (commercial) vessels "alive".
- To minimize the number of Red casualties.

Considering the current scenario, these objectives are thus conflicting. Moreover, the true Pareto front is here unknown. In the next section we report the experimental results using the above model.

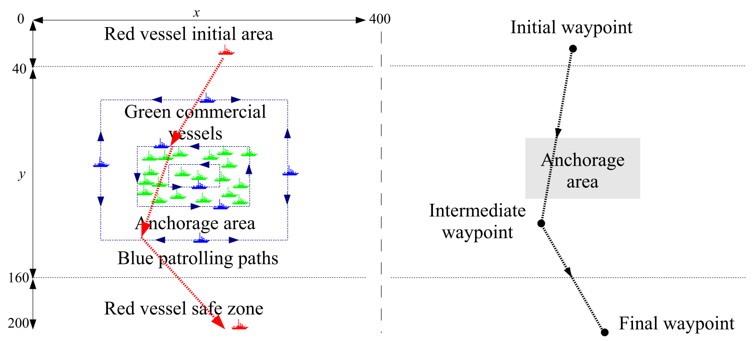

Fig. 5. Schematic overview of the case study . The map covers an area of 100 by 50 nautical miles (1 nm = 1.852km). 7 Blue vessels conduct patrols to protect an anchorage of 20 Green commercial vessels against a single Red vessel. The Red vessel intends to break the Blues defence, inflict damages to the anchored Green vessels and finally, to escape to the Red vessel safety area. **Left**: The dashed lines depict the patrolling paths of the different Blue vessels. 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 defence whereas the larger and heavily armoured ships inside the anchorage are the second defensive layer. The Red craft was set up to initiate its attack from the north. The initial positions of Blue vessels are fixed. In contrast, the Green commercial vessels' initial positions are randomly generated within the anchorage area at each MANA execution. **Right**: Example Red route. Home waypoint (Home WP) is constrained to the distinct agent's initial area. Similarly, the final waypoint is to be located in the opposite area. Intermediate waypoints occur in the remaining middle area. Note that in the below experiments, we dynamically evolve the number of intermediate waypoints. Whereas the coordinates of all waypoints, including the home and final ones, are subjected to evolution.

7 Results

To evaluate the quality of the (multi-objective) solutions through the evolutionary search, the hypervolume indicator [27] is utilized. This method is currently considered as the state of the art technique to evaluate Pareto fronts. This indicator measures the size of the objective space subset dominated by the Pareto front approximation.

In Fig. 6, the HVLC is compared with the fixed-length genome approach studied in [8] using NSGAII. Whereas the numerical values resulting from the evolutionary experiments are shown in Table 2.

In Fig. 6 and Table 2, it can be observed that HVLC consistently outperformed its fixed-length genome counterpart. When comparing the best Pareto set approximations (Fig. 7) resulting from both sets of evolutionary runs, comparable results were achieved. HVLC was nevertheless more consistent throughout the 10 distinct evolutionary runs (when considering the mean hypervolume indicator value and standard deviation) in achieving competitive results.

(a) Fixed Blue parameters

Parameter	Value
Detection range (nm)	24
# hits to be killed	2
Weapon hit prob.	0.8
# patrolling agents	7
Speed (unit)	100
Weapon range (nm)	8
Determination	50∨0
Aggressiveness	0∨100
Cohesiveness	0

(b) Fixed Red parameters

Parameter	Value
Detection range (nm)	8
# hits to be killed	1
Weapon hit prob.	0.8
# agents	5
Speed (unit)	100
Weapon range (nm)	5

(c) Evolvable Red paramaters

Parameter	Min	Max
Vessel home position(x,y)	(0,0)	(399,39)
Intermediate waypoint position (x,y)	(0,40)	(399,159)
Vessel final position (x,y)	(0,160)	(399,199)
Determination	20	100
Aggressiveness	-100	100
Cohesiveness	-100	100

Table 1. (a): Fixed Blue parameters. Value pairs are specified for the determination and aggressiveness properties. In this model, Blue changes its behaviour upon detecting Red, i.e., Blue "targets" Red, with aggressiveness being increased, when the latter is within Blue's detection range. (b): Fixed Red parameters. The behavioural parameters are not specified as these parameters are subjected to evolution. (c): Evolvable Red parameters: As mentioned earlier, the intermediate waypoint structures are dynamically inserted/removed within the agent-based model during the evolutionary search. The final positions of the Red craft is constrained to the opposite region (with respect to initial area) to simulate escapes from the anchorage following successful attacks. Behavioural or "psychological" elements are included in the evolvable decision variables. The aggressiveness determines the reaction of individual vessels upon detecting an adversary. Cohesiveness influences the propensity of vessels to maneuver as a group or not, whereas determination stands for the agent's willingness to follow the defined routes (go to next waypoint). The Red vessels' aggressiveness against the Blue patrolling force are varied from unaggressive (-100) to very aggressive (100). Likewise, the cohesiveness of the Red crafts are varied from independent (-100) to very cohesive (100). Finally, a minimum value of 20 is set for determination to prevent inaction from occurring.

Although the above preliminary results suggest a somewhat promising potential for the variable-length genome approach, only a single case study was here considered. As mentioned in the introduction, we do expect that this method may only benefit scenarios in which the structural evolution of simulation models is relevant (i.e. where the evolutionary experiment is not *constrained* to a set of evolvable parameters). Our future work will include a broader set of scenario

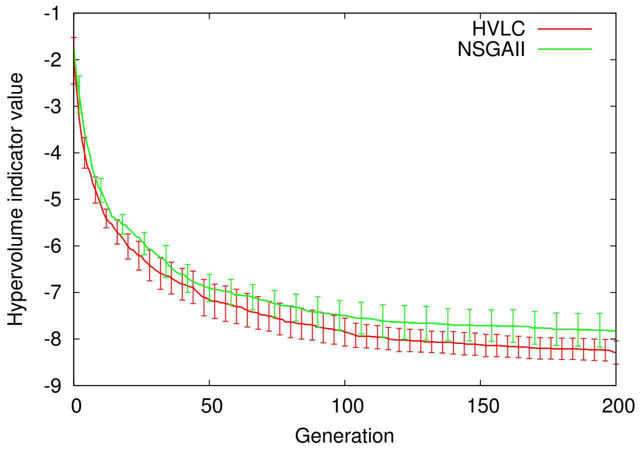

Fig. 6. Hypervolume volume dynamics. The lines identify the hypervolume indicator value averaged over 10 individual evolutionary runs using unique seeds. The error bars stands for the confidence interval (with $\alpha = 0.05$). The negative value of the hypervolume indicator is utilized for consistency with the *cost minimization* approach used in the experiments.

Table 2. Pareto optimality performance

Algo.	Best	Mean
NSGAII	-8.9249	-7.8211 $\pm$ 0.36
HVLC	-8.9995	**-8.2854** $\pm$ **0.25**

The bold values identify the best overall Pareto optimal approximation sets (when considering both the mean and a 95% confidence interval).

to better evaluate HVLC against existing evolutionary computation techniques such as the NSGAII.

In the remainder of this section, we discuss a potential explanation for the dynamics observed in the experiments using the fixed-length genome approach: A potential drawback of the fixed-length genome representation is the *epistasis* phenomenon. In biology, epistasis refers to non-linear interactions occurring between genes. It is currently hypothesized that epistasis may emphasize the "ruggedness" of the fitness landscape [3], leading to an increased level of difficulty for the evolutionary search. Note that epistasis may already occur *implicitly* between genes (here simulation model parameters) according to their specific values. The specification of control genes *explicitly* introduce epistatic interactions, which may harden the search difficulty level.

Indeed, a slight mutation in the value of "control" or epistatic genes, using the fixed-length genome approach, would result in large phenotype changes, where many waypoints may be turned off or on simultaneously. This clearly introduces non-linearities in the evolutionary search process. The level of epistatic

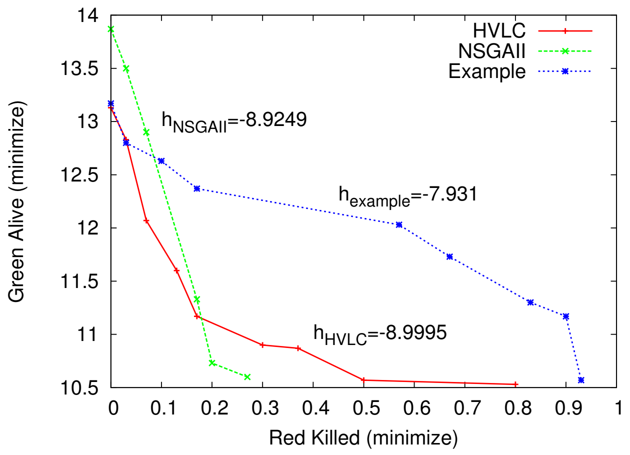

Fig. 7. Best Pareto set approximations resulting from the 10 distinct evolutionary runs conducted using NSGAII and HVLC. A third example Pareto front is shown to illustrate how a difference of 1 in the hypervolume indicator value may affect the Pareto front approximation quality.

interactions would moreover increase according to the pre-specified maximum number of waypoints. This ultimately limits the utilization of the fixed-length genome representation for such agent-based model optimization applications.

The results suggest that the variable-length genome representation may potentially alleviate this epistatic issue, leading to better Pareto optimality performances. Although these results are promising, further experiments remain required to investigate the potential benefit of the variable-length genome representation approach. Future work include the evolution of simulation models in which a large number of model structures is evolved. We hypothesize that the fixed-length genome approach would rapidly and significantly be outperformed by HVLC when tackling larger search problems including a relatively high level of explicit epistatic interaction. Methods which may quantify epistasis (as used in molecular biology research [21]) would also assist this future research.

Also the evolutionary complexification concept proposed by Stanley and Miikkulainen [23] will be investigated as it may reduce the number of search generations (i.e. optimizing the search convergence speed) through avoiding the evaluation of unnecessary complex simulation models.

8 Conclusions

The Computational Red Teaming methodology and related supporting technologies were first introduced. A survey on variable-genome length techniques for evolutionary computation was then presented. The evolutionary framework CASE was briefly described and utilized using a novel variable-length computational technique coined the hybrid variable length crossover. A series of experiments was conducted in which the structure of a simplified military agent-based model was evolved. HVLC was compared against a fixed-length genome approach. The experimental results suggested that our variable-length genome approach is a promising technique which, in overall, achieved better Pareto optimality performances than using fixed-length genomes. Nevertheless, this potential benefit must be further examined in future work where supplementary evolutionary experiments of differing complexity will be conducted.

Acknowledgements

We would like to 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 Defence Innovative Research Programme FY08.

References

1. Bitinas, E.J., Henscheid, Z.A., Truong, L.V.: Pythagoras: A New Agent-based Simulation System. Technology Review pp. 45–58 (2003)
2. Burke, D., De Jong, K., Grefenstette, J., Ramsey, C., Wu, A.: Putting More Genetics Into Genetic Algorithms. Evolutionary Computation 6(4), 387–410 (1998)
3. Choi, S., Jung, K., Moon, B.: Lower and Upper Bounds for Linkage Discovery. Evolutionary Computation, IEEE Transactions on 13(2009), 201–216 (2009)
4. Choo, C.S., Chua, C.L., Tay, S.H.V.: Automated Red Teaming: a Proposed Framework for Military Application. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation. pp. 1936–1942. ACM (2007)
5. Choo, C.S., Chua, C.L., Low, K.M.S., Ong, W.S.D: A Co-evolutionary Approach for Military Operational Analysis. In: GEC '09: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation. pp. 67–74. ACM (2009)
6. Cioppa, T.M., Lucas, T.W., Sanchez, S.M.: Military Applications of Agent-based Simulations. In: Proceedings of the 36th Winter Simulation Conference. pp. 171–180. ACM (2004)
7. Deb, K., Agrawal, R.: Simulated Binary Crossover for Continuous Search Space. Complex Systems 9(2), 115–148 (1995)
8. Decraene, J., Chandramohan, M., Low, M.Y.H., Choo, C.S.: Evolvable Simulations Applied to Automated Red Teaming: A Preliminary Study. In: Proceedings of the 42th Winter Simulation Conference. pp. 1444–1455. ACM (2010)
9. Decraene, J., Low, M.Y.H., Zeng, F., Zhou, S., Cai, W.: Automated Modeling and Analysis of Agent-based Simulations using the CASE Framework. In: In Proceedings of 11th International Conference on Control, Automation, Robotics and Vision (ICARCV). pp. 346–351. IEEE (2010)
10. Decraene, J., Yong, Y.C., Low, M.Y.H., Zhou, S., Cai, W., Choo, C.S.: Evolving Agent-based Simulations in the Clouds. In: Third International Workshop on Advanced Computational Intelligence (IWACI). pp. 244–249. IEEE (2010)

11. Decraene, J., Zeng, F., Low, M.Y.H., Zhou, S., Cai, W.: Research Advances in Automated Red Teaming. In: Proceedings of the 2010 Spring Simulation Multiconference (SpringSim). pp. 47:1–47:8. ACM (2010)
12. Easton, A., Barlow, M.: CROCADILE: An Agent-based Distillation System Incorporating Aspects of Constructive Simulation. In: Proceedings of the SimTecT 2002 Conference. pp. 233–238 (2002)
13. Goldberg, D., Korb, B., Deb, K., et al.: Messy Genetic Algorithms: Motivation, Analysis, and First Results. Complex systems 3(5), 493–530 (1989)
14. Harvey, I.: The SAGA Cross: The Mechanics of Recombination for Species with Variable Length Genotypes. In: Männer, R., Manderick, B. (eds.) In Proceeding of the Parallel Problem Solving from Nature 2. pp. 269–278. Elsevier (1992)
15. Hutt, B., Warwick, K.: Synapsing Variable-length Crossover: Meaningful Crossover for Variable-length Genomes. IEEE transactions on evolutionary computation 11(1), 118–131 (2007)
16. Ilachinski, A.: Artificial war: Multiagent-based Simulation of Combat. World Scientific Pub Co Inc (2004)
17. Kewley, R.H., Embrechts, M.J.: Computational Military Tactical Planning System. IEEE Transactions on Systems, Man, and Cybernetics, Part C 32(2), 161–171 (2002)
18. Lauren, M., Stephen, R.: Map-aware Non-uniform Automata (MANA)-A New Zealand Approach to Scenario Modelling. Journal of Battlefield Technology 5, 27–31 (2002)
19. Low, M.Y.H., Chandramohan, M., Choo, C.S.: Multi-Objective Bee Colony Optimization Algorithm to Automated Red Teaming. In: Proceedings of the 41th Winter Simulation Conference. pp. 1798–1808. ACM (2009)
20. Lucas, T.W., Sanchez, S.M., Martinez, F., Sickinger, L.R., Roginski, J.W.: Defense and Homeland Security Applications of Multi-agent Simulations. In: Proceedings of the 39th Winter Simulation Conference. pp. 138–149. IEEE (2007)
21. Matsuura, T., Kazuta, Y., Aita, T., Adachi, J., Yomo, T.: Quantifying Epistatic Interactions among the Components Constituting the Protein Translation System. Molecular Systems Biology 5(297) (2009)
22. McDonald, M.L., Upton, S.C.: Investigating the Dynamics of Competition: Coevolving Red and Blue Simulation Parameters. In: Proceedings of the 37th Winter Simulation Conference. pp. 1008–1012. ACM (2005)
23. Stanley, K., Miikkulainen, R.: Competitive Coevolution through Evolutionary Complexification. Journal of Artificial Intelligence Research 21(1), 63–100 (2004)
24. Wong, A.C.H., Chua, C.L., Lim, Y.K., Kang, S.C., Teo, C.L.J., Lampe, T., Hingston, P., Abbott, B.: Team 1: Applying Automated Red Teaming in a Maritime Scenario. In: In Scythe 3: Proceedings and Bulletin of the International Data Farming Community. pp. 3–5 (2007)
25. Xu, Y.L., Low, M.Y.H., Choo, C.S.: Enhancing Automated Red Teaming with Evolvable Simulation. In: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation. pp. 687–694. ACM (2009)
26. Yang, A., Abbass, H., Sarker, R.: Characterizing Warfare in Red Teaming. IEEE Transactions on Systems, Man, and Cybernetics, Part B 36(2), 268–285 (2006)
27. Zitzler, E., Brockhoff, D., Thiele, L.: The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In: Proceedings of The 4th International Conference on Evolutionary Multi-criterion Optimization, Lecture notes in computer science. vol. 4403, pp. 862–876. Springer (2007)