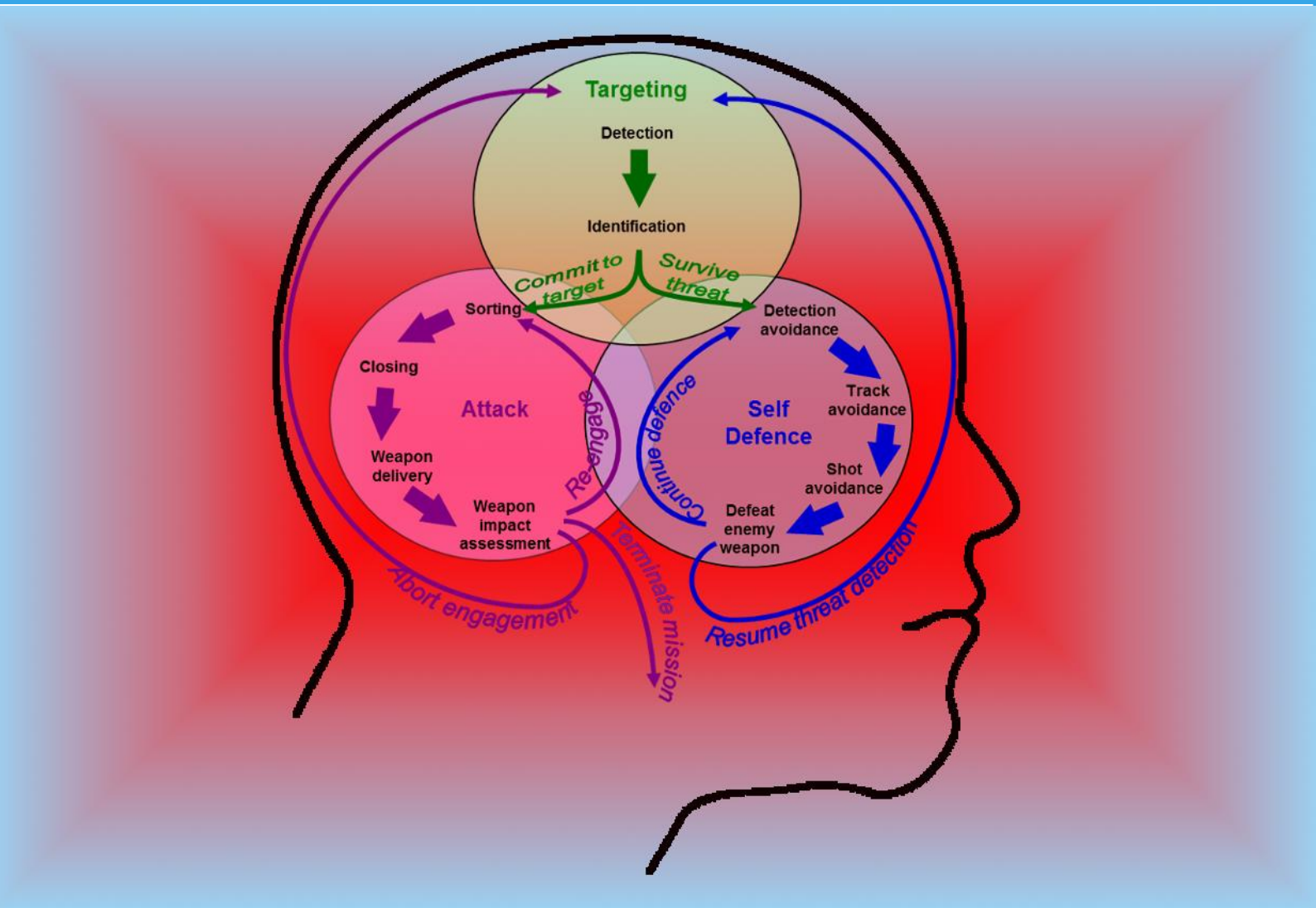
Co-Evolutionary Learning for Cognitive Computer Generated Entities

Customer

National Aerospace Laboratory NLR

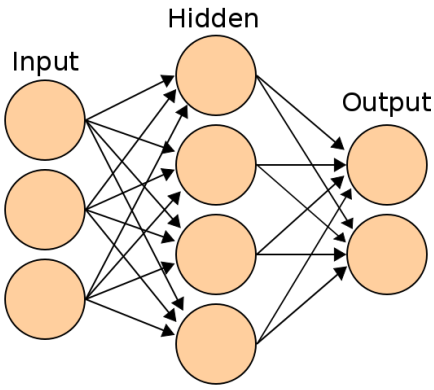
NLR-TP-2014-157 - May 2014



National Aerospace Laboratory NLR
Anthony Fokkerweg 2
1059 CM Amsterdam
The Netherlands
Tel +31 (0)88 511 3113
www.nlr.nl

EXECUTIVE SUMMARY

Co-Evolutionary Learning for Cognitive Computer Generated Entities



Problem area

As an alternative to training fighter pilots in a 'live' environment – a resource expensive task which is difficult to organize – an air force might allow these pilots to train within a simulated tactical environment instead, thereby fighting with and against software agents: the Computer Generated Entities (CGEs). Such a training scenario will typically allow for the indirect (e.g. via the environment) or direct (e.g. via a transaction) interaction between its entities, be it friendly or hostile. In order for this interaction to proceed smoothly, it is paramount that all those that participate behave according to what is expected of them. In the case of CGFs, this may be accomplished by guiding their behaviour by means of a cognitive model. However, finding the most effective one is a resource-expensive task, given that this typically entails the thorough testing of many potential models, thereby requiring many hours of knowledge-elicitation sessions with experts on various domains. Therefore it may be preferable to automate this process, such as letting the given CGFs learn the most effective models by themselves.

Report no.
NLR-TP-2014-157

Author(s)
W.E.X. Wilcke
M. Hoogendoorn
J.J.M. Roessingh

Report classification
UNCLASSIFIED

Date
May 2014

Knowledge area(s)
Training, Simulation and Operator PerformanceTraining,
Missiesimulatie en Operator Performance

Descriptor(s)
Intelligent Agents
Computer Generated Entities
Evolutionary Computing
Machine Learning
Air Combat

This report is based on a paper to be published in Lecture Notes in Computer Science (27th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2014, June 3-6, 2014 Kaohsiung, Taiwan, Proceedings), by Springer, Germany.

Description of work

Constructing a self-learning agent is one of the specialisations that one can find within the field of machine learning, a branch of artificial intelligence. One such specialisation is that of evolutionary computing; a group of optimization techniques that follow the theory of natural evolution as originally described by Charles Darwin (1809-1882). Due to their properties, these techniques are particularly well suited for learning agent behaviour in an open environment. To this end, an evolutionary algorithm will be developed, of which its applicability will be determined on the task of finding the most effective cognitive models for simulated fighter pilots.

Where earlier research on this topic focussed on determining the optimal parameters of the CGFs' model of Situation Awareness (SA), the current study will lay the emphasis on determining the best approach to optimize the topology of the cognitive models that guide the behaviour of said CGFs. Furthermore, this study will experiment with coevolution; a notion which will allow multiple (opposing) CGFs to learn simultaneously with the added benefit of inducing a form of arms race between opposing sides.

Results and conclusions

Five separate experiments have been conducted, with the first and second forming a baseline with Random Search (RS) and Random-Restart Hill Climbing (RRHC), respectively. The remaining three experiments focus on the performance of the Evolutionary Algorithm (EA), with the first of these lacking coevolution. In contrast, the second and third *do* Feature Coevolution (EA featuring Coevolution, EA ft. C), with the sole difference being that the latter starts from scratch while the former uses a partially-evolved set of potential models at initialization.

The results seem to support the algorithm's ability to learn the most-effective behaviour, as well as to optimize the cognitive models. However, these models appear to have suffered from overfitting, which is thought of being caused by the overly simplistic combat scenarios that lack the need for complex awareness. Therefore, the authors recommend that future research on this topic should prioritize the goal of developing a method to detect and prevent such overfitting.

Applicability

A key property of machine learning is the generalizability of its methods such that they are applicable to a wide variety of data. Hereto, an optimal model is typically treated as the solution to a numerical-optimization problem. Therefore, such a developed technique may be adapted to other problems with relatively little effort, such as learning (behavioural) models for more complex scenarios (four-versus-four, eight-versus-eight, etc.) with different fighter planes (F-16, F-35, etc.), different armament (Active, Semi-Active and Passive type of missile guidance systems, etc.), or different mission types (Offensive Counter Air, Defensive Counter Air). This may even be extended to other domains, such as autonomous manoeuvring of Remotely Piloted Aircraft.

National Aerospace Laboratory NLR
Anthony Fokkerweg 2, 1059 CM Amsterdam,
P.O. Box 90502, 1006 BM Amsterdam, The Netherlands
Telephone +31 (0)88 511 31 13, Fax +31 (0)8 511 32 10, Website: www.nlr.nl

Co-Evolutionary Learning for Cognitive Computer Generated Entities

W.E.X. Wilcke[1], M. Hoogendoorn[2] and J.J.M. Roessingh

[1] National Aerospace Laboratory / VU University Amsterdam
[2] VU University Amsterdam

Customer
National Aerospace Laboratory NLR
May 2014

This report is based on a paper to be published in Lecture Notes in Computer Science (27th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2014, June 3-6, 2014 Kaohsiung, Taiwan, Proceedings), by Springer, Germany.

Customer	National Aerospace Laboratory NLR
Contract number	- - -
Owner	NLR
Division NLR	Air Transport
Distribution	Unlimited
Classification of title	Unclassified
Date	May 2014

Approved by:

Authors	Reviewer	Managing department
W.E.X. Wilcke/ J.J.M. Roessingh	R.T.A. Meiland	H.G.M. Bohnen
Date 07-04-2014	Date 14-04-2014	Date April 2014

Summary

In this paper, an approach is advocated to use a hybrid approach towards learning behaviour for Computer Generated Entities (CGEs) in a serious gaming setting. Hereby, an agent equipped with cognitive model is used but this agent is enhanced with Machine Learning (ML) capabilities. This facilitates the agent to exhibit human like behaviour but avoid an expert having to define all parameters explicitly. More in particular, the ML approach utilizes co-evolution as a learning paradigm. An evaluation in the domain of one-versus-one air combat shows promising results.

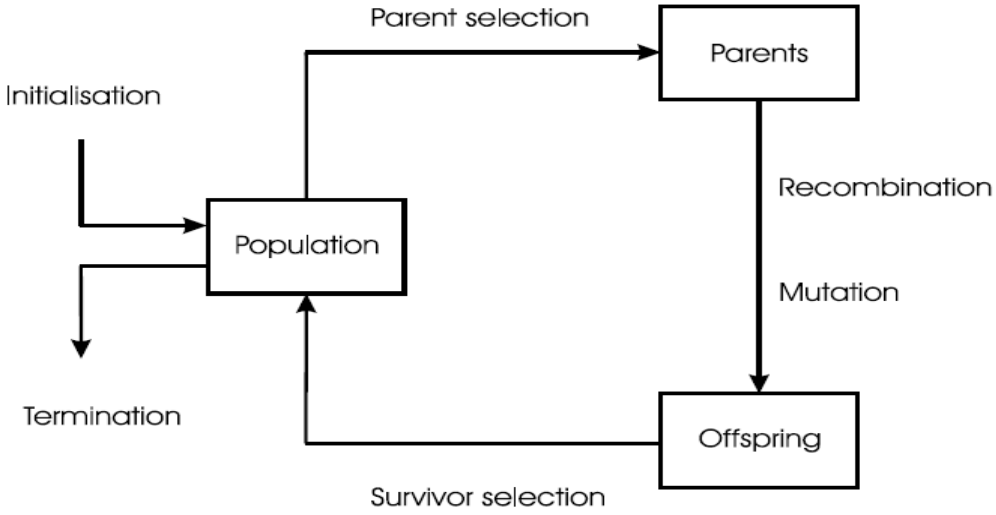


Fig. 0. *General loop of an evolutionary algorithm (taken from [20])*

Content

Abbreviations

Acronym	Description
AI	Artificial Intelligence
ANN	Artificial Neural Network
BVR	Beyond-Visual-Range
CAP	Combat-Air-Patrol
CAS	Close-Air-Support
CGE	Computer_Generated Entity
CGF	Computer-Generated Forces
DAG	Directed A-cyclic Graph
DM	Decision Making
EA	Evolutionary Algorithm
EA ft. C	Evolutionary Algorithm featuring Coevolution
FLOT	Forward Line of Own Troops
GCI	Ground-Controlled Interception
HC	Hill Climbing
ML	Machine Learning
NLR	National Aerospace Laboratory
NN	Neural Network
RRHC	Random-Restart Hill Climbing
RS	Random Search
RWR	RADAR-Warning Receiver
SA	Situation Awareness
SB	Smart Bandits

1 Introduction

Serious gaming is playing a more and more prominent role to facilitate training in a variety of domains [1, 13]. The advantages of taking a serious gaming approach opposed to 'real life' training include (but are certainly not limited to) the ability to train realistic scenarios that are difficult to perform in the real world, the lower cost and the possibility to frequently repeat key learning events. In order to maximize the benefits, serious games should be populated by realistically behaving agents that for instance act as adversaries or teammates for the trainee, the so-called Computer Generated Entities (CGEs). Imagine a scenario in which an F-16 fighter pilot is trained in a virtual environment to use a specific tactic. Without having a realistic enemy to fight against, the training will not have the desired impact, while having to invite another fighter pilot to play the role of the enemy is inefficient in terms of training and tedious for the role-player.

One approach to obtain realistic behaviour of CGEs is to distil knowledge from domain experts and build entities that incorporate the knowledge. Here, models that utilize this knowledge can be of a cognitive nature to establish human-like behaviour. Another approach is to use pure learning-based techniques (e.g. reinforcement learning, evolutionary learning) and let the computer learn appropriate behaviour. Both approaches however have severe disadvantages: for complex domains with limited access to domain experts the knowledge-based approach might be very difficult whereas the learning-based approaches are not guaranteed to provide realistic computer-generated entities as they might learn completely different strategies.

This paper takes a hybrid approach. It departs from a graph-based cognitive model which incorporates partial knowledge about the domain and applies evolutionary learning techniques to fine-tune the model towards a specific scenario. To be more specific, the cognitive model used is a Situation Awareness (SA) model (cf. [7]) which has been extended with a simple Decision Making (DM) model. Previously, attempts to apply learning techniques in this context have shown promising results (see [3] and [10]) but also revealed that a substantial amount of additional knowledge was needed to establish this behaviour: in [3] the desired responses of the entities in all situations were needed whereas [10] has shown that learning a complex scenario can be troublesome due to characteristics of the fitness landscape. In this paper a solution to both problems is proposed. A co-evolutionary approach is used which drives two competing entities to more and more complex behaviour (see e.g. [6]). It needs a limited amount of expert knowledge and circumvents the problem previously found with the relatively flat fitness landscape. The approach has been evaluated for the domain of fighter pilots.

This paper is organized as follows. In Section 2 related work is discussed as well as the domain of application. The approach is proposed in Section 3 whereas Section 4 presents the case study to investigate the effectiveness of the approach. Section 5 shows the results obtained and the paper is concluded with a discussion in Section 6.

2 Background

This section discusses the background of the approach, more in specific, it discusses cognitive modelling, the combination of cognitive models and machine learning, and the specific machine-learning approach utilized in this paper, namely evolutionary algorithms.

2.1 Cognitive models

Cognitive models deal with the symbolic-information processing level and are thought to be largely independent of models at the physiological (neurological, millisecond) level. Some of those models are based on so-called unified theories of cognition. The goal of such a model is to show how a single control structure can handle all of the cognitive processes of which the human mind is capable. Soar [11], ACT-R [2], EPIC [9], CLARION [14], are frequently cited in connection with this class of models. In both Soar and ACT-R, cognition is largely synonymous with problem solving. They are both based on production systems (basically if .. then ..-systems) that require two types of memory: 'declarative' memory for facts and 'procedural' memory for rules. In contrast, a variety of component cognitive models have been defined that facilitate the generation of human-like behaviour of role-playing agents in simulations for complex skill training. More specifically, in a previous research, a component model, a Situation Awareness (SA) model in the form of directed, a-cyclic graphs (DAGs) has been devised (cf. [7]) which is re-used in this research.

The SA model is meant to create high-level judgments of the current situation following a psychological model from Endsley [4]. The model essentially comprises of three forms of beliefs, namely simple beliefs, which can be directly derived from observations per-formed by the agent, or other simple beliefs. Furthermore, complex beliefs express combinations of simple beliefs, and describe the current situation on an abstracted level, and finally, future beliefs express expectations of the agent regarding projected (future) events in the scenario. All these beliefs are assigned an activation value between 0 and 1 and are connected via a network, in which each connection between a pair of beliefs has a particular strength (with a value between -1 and 1). Several update properties are specified that express how new knowledge obtained through observations is propagated through the network using the activation value of beliefs as a rank ordering of priority (the most active beliefs are updated first). For the sake of brevity, the details of the updating mechanism of the algorithm have been omitted, see [7] for an in-depth treatment of the update rules. The domain knowledge that is part of the model is a so-called belief network.

Once a judgment of the situation has been created using the model above, a decision on the action given this situation should be selected. For this purpose a Decision Making (DM) model can be specified. This can be rather simple, in this case in the form of connecting the states in the SA model to actions using certain weights, i.e. again a weighted graph.

2.2 Machine Learning and Cognitive Models

As argued in the introduction, a combination of cognitive models and learning combines the benefits of both approaches whereas it also takes away some of the separate disadvantages. Hereby, cognitive models are based on coarse knowledge, and a machine-learning technique fine-tunes this knowledge. In the current effort, neuro-evolution in the form of a co-evolutionary process is applied to agents that contain a cognitive model that combines both SA and DM (that is, both models are adaptive and the DM process is contingent on the SA process), see Section 3.

In co-evolutionary learning, one can appoint coevolving species into one of the following two groups; symbiotic or parasitic. With the former variant the species cooperate and everyone benefits, while with the latter form there is usually a winning and a losing side. Irrespective, both types guide the evolution by continuously moving towards the optimum situation for all species involved. When developing an evolutionary algorithm to find a solution to a problem that involves multiple interacting agents (e.g. opposing fighter jets), one may choose to incorporate coevolution. By simultaneously evolving more than one population, and by letting the individuals from either population be each other's opponents, a situation is created in which the gain in mean fitness of the one population is directly related to a loss in mean fitness of the other population.

To enable the evolution of a cognitive model (in this case, a DAG which is very much like a neural network), a neuro-evolutionary approach can be deployed. Neuro-evolution [18] concerns a group of evolutionary algorithms which specifically aim at adapting or learning a (neural) network. Most popular approaches, such as NEAT [17], EANT [8] and EANT2 [15] grow a network from the ground up, continuously increasing complexity by adding new nodes and connections. Some do however start with a large network and try to prune it. For example, EPNet [26] starts with random overly-bloated network topologies and removes those nodes and connections that it deems redundant or irrelevant. These "pruning" algorithms have the benefit of being able to find the more optimized topologies at a lower cost than if a network-growing approach were to be used.

3 Approach

This section explains the learning approach that has been used as well as the slightly adjusted model which is subject to learning in more detail.

3.1 Cognitive Model

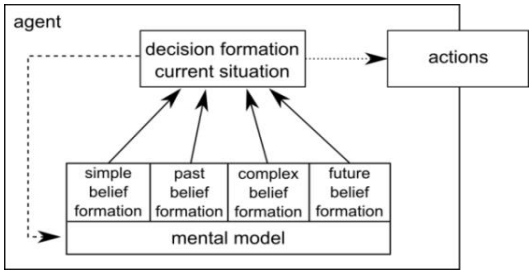


Fig. 1. *Framework of the Decision Model. Decisions – sets of executable actions – are formed based on the currently active beliefs in the mental model.*

The cognitive model which forms the basis of the agent will be an extended version of the SA model described in [7] combined with a DM model. More specific, the default set of concepts as described in Section 2.1 will be joined by past beliefs (cf. [12]). These are beliefs on past events, which broaden the set of options to learn an appropriate model. Furthermore, the DM model used has been developed specifically for the purpose of this research, which also requires learning of appropriate weights. The DM model is shown in Figure 1, and consists of beliefs and decisions, with the latter being defined as a non-empty set of executable actions. A sole exception to this rule will be the decision to do nothing, which will be chosen by default when none of the decisions have a certainty c_d that exceeds its certainty threshold τ_d. For a decision to be valid, it will be required to have at least one incoming connection, whereby the source of this connection should be a belief contained within the SA model. Based on the activation value of this belief v_b and the weight $w_{db} \in [-1, 1]$ of its relation with a decision, the certainty of the latter will be calculated as formulated in Equation 1. Here, a positive outcome will denote the certainty that the selected decision is the correct one (activator), while the opposite will be true for a negative value (inhibitor).

$$c_d := \sum_{i=0}^{n-1} v_i w_{di} \qquad \textit{Equation 1}$$

3.2 Evolutionary Learning Algorithm

Two populations of agents, both equipped with the described cognitive model, will compete with each other, thus enabling parasitic co-evolution to occur between these populations. In addition, individuals may migrate between the populations, thus lowering the chance of a single population getting stuck in local optima.

Initially, both populations will be filled with N randomly-generated individuals, each individual representing the connections, with associated weights, of an instance of the cognitive model. To minimize bias of an 'unfair' match, each individual will be tested against Q unique opponents, randomly picked from the hostile population. Furthermore, as to minimize situational advantage, every individual-opponent pair will be evaluated R times in different (randomly-picked) scenario's. Individual fitness will then be based on the scenario's outcome O – victory (1), defeat (0), or draw (0.5) – and on the ratio of the combined size of both models g to the maximum size found amongst the population G. This ratio, in which the size of a model is defined as its total number of concepts and relations, will provide selection pressure to the more parsimonious model. Balancing this advantage will be achieved by taking the parameter for graph-size influence $I \in [0, 1]$, into account as well. Together, the two measures and the single parameter will contribute to the fitness f as formulated in Equation 2. Note that, in order to guard the bounds of the fitness range, the terms on the left and right side of the sum sign will be scaled by $(1 - I)$ and I, respectively.

$$f := O(1 - I) + I(1 - \frac{g}{G}) \qquad \textit{Equation 2}$$

Independently in both populations, parents will be appointed by tournament selection. Depending on a probability$P_{crossover}$, this will involve either mutation or crossover. The crossover operator will primarily have an exploratory function. This will be accomplished on the chromosome level by inheriting one sub model from each of the two parents (i.e. given parent couple SA_XDM_X and SA_YDM_Y, two new chromosomes SA_XDM_Y and SA_YDM_X will be formed).

In contrast with crossover, the more important mutation operator will perform on the genetic level, i.e. on the models. Depending on a probability$P_{topologyMutation}$, one of an individual's models will undergo either a (Gaussian) mutation of its weights, or a random mutation of its topology. The topology will be represented as weighted DAGs with the vertices and edges taking on the role of concepts and their relations, respectively (see Section 2.1). Therefore, any change in topology will encompass either the addition or removal of a vertex or edge, with several simple rules guarding the validity of the resulting models. For example, a topology mutation might involve the addition of a new *observation*. Alternatively, a weight mutation might modify the

weight w_{qp} of edge e_{qp}, thereby adding a value x to said weight followed by rescaling the weights of all 'sibling' edges w_{qi} with $i \in [0, n_q)$ to distribute this change evenly.

After evaluation, the offspring will be inserted into their parents' population. Alternatively, depending on a probability$P_{migration}$, a small number of individuals may switch populations. Irrespective, any surplus will subsequently be dealt with by following an elitist approach.

4 Case Study

In order to test the suitability of the approach, a ‘one versus one’ (1v1) combat engagement scenario has been devised, featuring two opposing agents (simulated fighter aircraft, ‘Attacker’ and ‘Defender’). Hence, their goal is to destroy one another. Note that this scenario is used to determine the fitness of individuals.

The primary sensor on-board the aircraft for detecting, identifying, tracking and locking the opponent is the radar. Good behavioural performance for the agent in control of the aircraft is defined as correctly detecting other aircraft on its radar, correctly identifying such aircraft, and subsequently engaging an aircraft in case it is hostile. During the engagement, the aircraft intercepts its opponent, while tracking the opponent via radar. When the opponent is within a distance that can be bridged by a missile (‘weapons range’), a ‘lock’ can be made on the opponent (that is, focusing radar energy on its opponent), followed by the firing of the aircraft’s radar-guided missiles. A scenario has been created in which this desired behaviour can be exhibited. Initial positions of aircraft and initial angles between the flight paths of the aircraft are randomized at the start of each run of the scenario. However, ‘Attacker’ will come from the direction of the so-called FLOT (the Forward Line of Own Troops) and ‘Defender’, will fly towards the FLOT, such that the two aircraft will generally head towards each other and will detect each other by radar at some point (not necessarily at the same time).

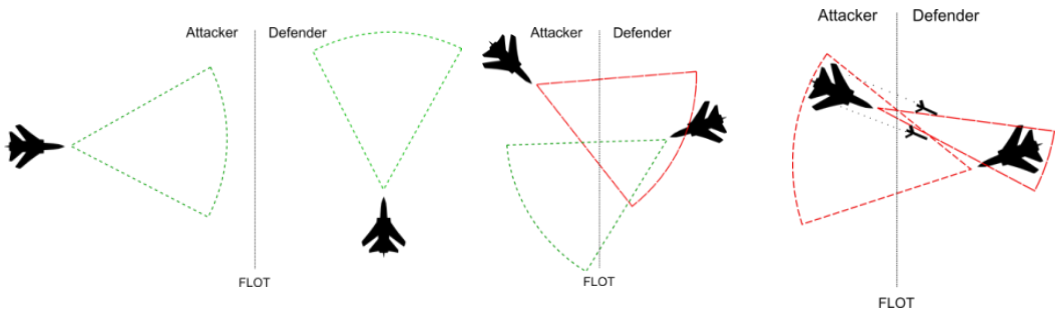


Fig. 2. *Overview of 1v1 scenario.*

Each aircraft may perform any number of actions, among which are manoeuvres that minimize the probability of detection on radar by the enemy, evasive manoeuvres, tracking and intercepting an opponent, making a radar lock on the opponent, and firing missiles on the opponent. In addition, an aircraft is free in its ability to roam around, provided that the movements are limited to the horizontal plane. When the opponent succeeded to make a lock and fire a missile, the aircraft will attempt to defeat said missile, for instance by manoeuvring in such fashion that the missile is unable to reach the targeted aircraft. Figure 2 sketches the three

main stages of such a scenario. First, both aircraft are on their own side of the FLOT, and are unaware of each other's presence (Left).Then, 'Attacker' detects the 'Defender' on its radar (Middle). After achieving a weapons-lock on 'Defender', 'Attacker' fires two missiles. The scenario will end with the destruction of an agent, or after a pre-set maximum amount of time has passed.

5 Results

In total, four series of experiments were conducted to compare the proposed Evolutionary Algorithm (EA) with a simpler EA without coevolution and two baseline algorithms (Random Search and Random Restart Hill Climbing).[1] Section 5.1 provides the results of the EA approach presented in Section 3.2, but without co-evolution. Section 5.2 presents the results of the full EA. Section 5.3 provides the results of experiments with the two baseline algorithms.

5.1 Simple EA (without coevolution)

In this first series of experiments only a single population was maintained, of which the individuals (in the 'Defender' role) were evaluated against a scripted 'Attacker' (identical to the one used in [10]). Hence, these individuals were required to adapt to an opponent that demonstrated non-adaptive behaviour. This provided a less dynamic fitness space, thus lowering the difficulty of the task.

The entire experiment was repeated five times with 2000 generations each, thereby starting with a fresh randomly-generated population of size $N = 1000$ on every restart. During evaluation, each individual participated in $Q = 10$ engagements, with each engagement consisting of $R = 3$ tries. The fitness in turn, was averaged each generation over these $Q \times R = 30$ evaluations, together with a graph-size influence of $I = 0.2$. In addition, a crossover probability $P_{crossover} = 0.05$ was set, as well as a topology-mutation probability$P_{topologyMutation} = 0.3$. Note that migration was omitted, as only a single population was maintained.[2]

[1] The project comprised of more extensive experiments than are reported in this paper (because of page restrictions imposed by the publisher). Additional experiments have been reported in a Technical Report [21]. The latter report describes, among other experiments, an experimental comparison between an initial population of untrained attackers, that will subsequently be trained using coevolution and an initial population of attackers that has been trained already against 'static' (scripted) opponents. To learn about the results of these additional experiments, the reader is referred to the Technical Report, which can be obtained from National Aerospace Laboratory NLR.

[2] The number of generations (2000) was sufficient to guarantee convergence to a stable fitness level (see figure 3). Other parameter values are partly based on own experience, partly based on empirical figures encountered in literature (e.g. [20]).

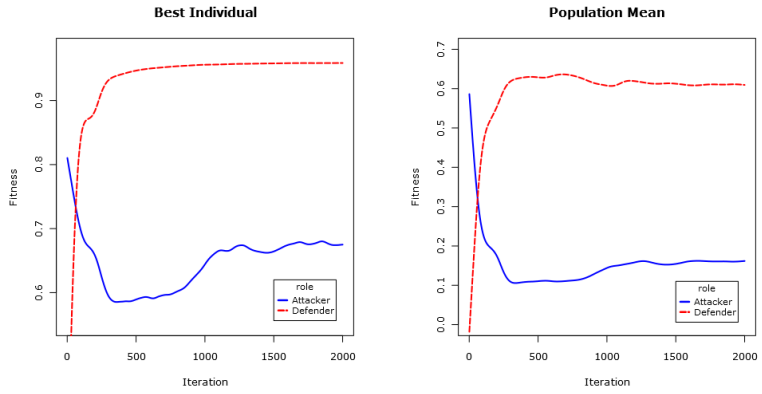


Fig. 3. *Performance of EA without coevolution. Left) Fitness of both populations' best individual per generation. Right) Mean fitness of both populations.*

Figure 3 shows the results. As evident by the sharp bend in the fitness curve of 'Defender', i.e. the learning agent, the algorithm found the optimal behaviour fairly early: at around $t = 400$. In the case of the best individual, the curve levels out at a fitness between 0.94 and 0.96. This level, at which every engagement was won, shows a slight incline during the remaining generations which reflects the optimization of the best individual's cognitive model. In contrast, the population's mean fitness appears to show a slight decline after the curve peaked $(f \pm 0.62)$ at around $t = 400$.

The fitness of the best scripted individual, i.e. 'Attacker', appears to follow an inverse pattern of that of 'Defender', showing a bend (towards a global minimum) at $t = 400$. However, from that point on, fitness increases until about t= 1200 generations. Finally, the mean fitness of the 'Attacker' population drops fast to a point between 0.1 and 0.15 after which a slight increase is observed, apparently countering the mean fitness of the 'Defender' population.

As a result of the EA, the cognitive models have lost most of their complexity through removal of redundant elements. This results in behaviour that favours immediate action, with high priority for intercepting and destroying the opponent.

5.2 Full EA

The second series of EA experiments included co-evolution of the 'Attacker' and 'Defender' population, thus providing the more difficult task for the individuals of training on a dynamic fitness landscape. The parameter settings used were equal to those applied in Section 5.1. Moreover, migration was featured with a probability $P_{migration} = 0.30$ per 25 generations.

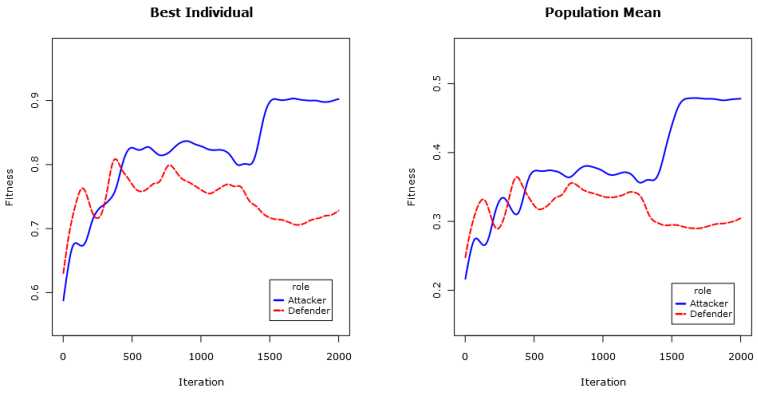


Fig. 4. Performance of EA with coevolution. Left: Fitness of both populations' best individual per generation. Right: Mean fitness of both populations.

Figure 4 shows that, with coevolution, the patterns at the individual and at the population level look remarkably similar, with only a constant difference between Best Individual and Population Mean of approximately 0.4. In either case, both sides appeared to be roughly evenly matched ($\Delta f \leq 0.08$) in the first 500 generations, with 'Attacker' and 'Defender' oscillating around each other, after which 'Attacker' gains advantage with a sudden larger gap in fitness between Attacker and Defender at approximately 1400 generations. As with the EA without coevolution, the resulting models and behaviour have lost most of their complexity. That is, redundant and rudimentary elements were removed, and immediate action was the preferred course of action.

5.3 Comparison with two baseline algorithms

In a set of experiments with two baseline algorithms (random search and random-start hill climbing), the initial solution for each run was similar to that of an EA's randomly-generated individual. These solutions were subsequently evaluated – 5 times per solution ($R = 5$) – in the same randomly-generated scenarios as those during the simple EA. In addition, both algorithms were repeated 3 times, of which the results were averaged. Similar as with the EAs, a graph-size influence of $I = 0.2$ was set. With random search, each of the runs involved 10.000 randomly-created solutions. With random-restart hill climbing, every run was allowed 10 restarts, with each one continuing until no improvement was found for 1.000 iterations. Any improvement with random search appeared to stall after at most 4.000 iterations. At that point, fitness values between 0.31 and 0.35 were reached. No further increase was seen during the remaining iterations. During random-restart hill climbing, the better fitness values were between 0.225 and 0.27. However, most tries resulted in a lower fitness, with a minimum around 0.2.

6 Discussion

In this paper, an approach has been presented to learn parameters of cognitive models using co-evolution. Hereby, a situation-awareness model as well as a decision making model have been used which include a large number of parameters. This co-evolutionary approach has been applied to the creation of virtual opponents for fighter-pilot training. This Case Study shows that the co-evolutionary approach results in higher fitness than typically observed with evolutionary algorithms without coevolution and baseline algorithms used as benchmarks. Only with the co-evolutionary approach, CGEs can be trained against opponents that adapt themselves, which may offer additional advantages, such as better generalizability of performance against different opponents. Therefore, it is difficult to make a precise comparison between the presented approaches. When comparing the results with other work, the most obvious comparison is with the work presented in [10]. Here, a similar SA model was trained by adapting only the weights with a learning algorithm. While moderate performance was achieved, it was theorized that restricting the model's topology might have limited the solution space too severely. Therefore, the research focused on both topology and weights to be learned. Unfortunately however, building a model from scratch would result in the loss of interpretability, due to the inability of such methods to take context into account in the labelling of newly created nodes or vertices. Instead, a pruning approach was followed, such that an existing and overly-redundant model may evolve to be both slim and effective. A co-evolutionary approach has also been proposed by Smith *et al.* [16] which aim at finding new strategies in 1-v-1 air combat. However, they do not deploy a cognitive model, making the level of explainability as well as the replication of human behaviour for effective training troublesome. For future work, it is envisioned to train the CGEs in the role of fighter pilots opponents for, more complex, scenarios and focus on an expert evaluation.

Acknowledgements

LtCol Roel Rijken (Royal Netherlands Air Force) provided a first version of the simulation environment used in this work. The authors also thank Pieter Huibers for his assistance with the simulation environment, and Armon Toubman, Remco Meiland and two anonymous reviewers for scrutinizing the work.

References

1. Abt CA (1970) Serious games. Viking Press, New York.
2. Anderson JR (1984) The architecture of cognition. Harvard University Press, Cambridge.
3. Gini ML, Hoogendoorn M, Lambalgen RM van (2011) Learning Belief Connections in a Model for Situation Awareness. In: Kinny D, Hsu D (eds) PRIMA'11: Proceedings of the 14th International Conference on Principles and Practice of Multi-Agent Systems. Lecture Notes in Artificial Intelligence. Springer Verlag, pp 373-384.
4. Endsley MR (1995) Toward a theory of Situation Awareness in dynamic systems. Human Factors 37(1):32-64.
5. Floreano D, Nolfi, S (1997) God Save the Red Queen! Competition in Co-Evolutionary Robotics. In: Koza JR, Deb K, Dorigo M et al (eds) Genetic Programming 1997: Proceedings of the Second Annual Conference, San Francisco, CA. Morgan Kaufmann, pp 398-406.
6. Hillis DW (1991) Co-evolving parasites improve simulated evolution as an optimization procedure. In: Langton CG, Taylor C, Farmer JD et al (eds) Artificial Life II: SFI Studies in the Sciences of Complexity. Addison-Wesley, Redwood City, California, 10:313–324.
7. Hoogendoorn M, Lambalgen RM van, Treur J (2011) Modeling Situation Awareness in Human-Like Agents using Mental Models. In: Walsh T (ed) IJCAI'11: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, pp 1697-1704.
8. Kassahun Y, Sommer G (2005) Efficient Reinforcement Learning Through Evolutionary Acquisition of Neural Topologies. In: Proceedings of the Thirteenth European Symposium on Artificial Neural Networks, Bruges, Belgium. D-Side Publications, pp 259-266.
9. Kieras DE, Meyer DE (1997) An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. Human-computer interaction, pp 391-438.
10. Koopmanschap R, Hoogendoorn M, Roessingh JJ (2013) Learning Parameters for a Cognitive Model on Situation Awareness. In: IEA-AIE 2013: Proceedings of the 26th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, LNCS, Spinger Verlag, pp 22-32.
11. Laird J, Rosenbloom P, Newell A (1987) Soar: An Architecture for General Intelligence. Artificial Intelligence 33:1-64.

12. Merk RJ (2013) Making Enemies: Cognitive Modeling for Opponent Agents in Fighter Pilot Simulators. Ph.D. thesis, VU University Amsterdam, Amsterdam, the Netherlands.
13. Michael D, Chen S (2006) Serious games: games that educate, train and inform. Thomson Course Technology, Boston, MA.
14. Naveh I, Sun R, (2006) A cognitively based simulation of academic science. Computational and Mathematical Organization Theory, pp 313-337.
15. Siebel N, Bötel B, Sommer G (2009) Efficient Neural Network Pruning during Neuro-Evolution. Neural Networks, pp 2920-2929.
16. Smith RE, Dike BA, Mehra RK et al (2000) Classier Systems in Combat: Two-sided Learning of Maneuvers for Advanced Fighter Aircraft. In: Computer Methods in Applied Mechanics and Engineering 186(2-4):421-437.
17. Stanley K, Miikkulainen R (2002) Evolving Neural Networks through Augmenting Topologies. Evolutionary Computation 10(2):99-127.
18. Yao X (1999) Evolving Artificial Neural Networks. IEEE 87(9):1423-1447
19. Yao X, Liu Y (1997) A New Evolutionary System for Evolving Artificial Neural Networks. IEEE Transactions on Neural Networks 8(3):694-713.
20. Eiben, A. & Smith, J. (2003). Introduction to Evolutionary Computing, Amsterdam: Springer, 2003.
21. Wilcke, W.E.X. (2013). Applying Evolutionary Computing to Agent-Behaviour Optimization -Learning Optimal Cognitive Models for Computer-Generated Fighter Pilots. Technical Report NLR-TR-2013-395. National Aerospace Laboratory NLR, Amsterdam, the Netherlands.

WHAT IS NLR?

The NLR is a Dutch organisation that identifies, develops and applies high-tech knowledge in the aerospace sector. The NLR's activities are socially relevant, market-orientated, and conducted not-for-profit. In this, the NLR serves to bolster the government's innovative capabilities, while also promoting the innovative and competitive capacities of its partner companies.

The NLR, renowned for its leading expertise, professional approach and independent consultancy, is staffed by client-orientated personnel who are not only highly skilled and educated, but also continuously strive to develop and improve their competencies. The NLR moreover possesses an impressive array of high quality research facilities.

***NLR** – Dedicated to innovation in aerospace*

www.nlr.nl