IFAC Proceedings Volumes, Volume 37, Issue 6, June 2004, Pages 1091-1096
16th IFAC Symposium on Automatic Control in Aerospace (ACA),
International Federation of Automatic Control, St. Petersburg, Russia, 14-18 June, 2004

ON-LINE EVOLUTIONARY ALGORITHM GUIDANCE FOR MULTIPLE MISSILES AGAINST MULTIPLE TARGETS

Evan J. Hughes and Brian A. White

Department of Aerospace, Power and Sensors,
Cranfield University,
Royal Military College of Science,
Shrivenham, Swindon, UK, SN6 8LA.
ejhughes@iee.org

Abstract
This paper details the application of a Cooperative Coevolution On-Line Evolutionary Algorithm (CCOLEA) to the guidance of a swarm of multiple missiles, against multiple targets. The CCOLEA trades the spatial distribution of missiles at impact, against the cost of re-aiming the missiles' seekers onto their final targets. A parallel approach is used where each missile optimises its own performance, based on limited information from the other missiles. The decision making process is thus distributed between the missiles giving distributed coordination. *Copyright ©2004 IFAC*

Keywords: Data Fusion, Genetic Algorithms, Guidance Systems, Missiles, Multi-objective Optimisation

1. INTRODUCTION

In the drive to increase the effectiveness of weapons, while reducing cost, simpler missiles with cheaper sensors are being developed, often using commercial-off-the-shelf equipment. The performance shortfall of using low cost sensors can sometimes be recovered by employing more sophisticated guidance and control algorithms. One key area of research is in multiple missile systems. Here multiple low cost missiles are flown as a self coordinating swarm in an attempt to increase reliability and overall kill probability. Previous work (Hughes, 2002*a*; Hughes, 2002*b*) has demonstrated that an On-Line Evolutionary Algorithm (OLEA) can be used to form the cooperative strategy of a command guided swarm.

In evolutionary guidance, the platform is first flown via a sequence of one or more points in space, before flying towards a predicted impact with the target, or a required rendezvous point. The points in space are evolved to generate a flight profile that is an optimal solution to a set of objectives and constraints. With multiple platforms, the flight profiles can be evolved simultaneously, each flight profile being evolved while accounting for the intended flight paths of the other missiles.

With the evolutionary guidance approach, for most of the engagement there is no direct, deterministic path between the sensors and the autopilot. Thus the initial stages of the flight path can be independent of the target position and motion, allowing different trajectories to be generated easily.

In the previous research, the swarm was developed from four missiles engaging a single moving target. In this command guided swarm, the processing for the on-line EA is all performed 'on the ground' and steering demands are communicated up to the missiles, with missile seeker information communicated back. In the command guided scenario, each chromosome

Published by Elsevier. This is the Author Accepted Manuscript issued with: Creative Commons Attribution Non-Commercial No Derivatives License (CC:BY:NC:ND 4.0).
The final published version (version of record) is available online at DOI:10.1016/S1474-6670(17)32325-X. Please refer to any applicable publisher terms of use.

of the OLEA describes a complete guidance scenario for all the missiles. All of the missiles are modelled together and the objectives can be assessed based on perfect knowledge of the anticipated behaviour of the missiles'.

This paper details the significant extension of the existing work to:

(1) Distributed cooperative coevolution with local processing on each missile giving asynchronous parallel operation, rather than a command guided system with a single processing system 'on the ground', and
(2) multiple missiles against multiple targets.

The CCOLEA allows complex swarm behaviours to emerge, yet only requiring low bandwidth communications, for example in this paper each missile broadcasts, ten times per second, the estimated time of impact, estimated impact angle, index of target being engaged and estimated speed at impact. The asynchronous nature of the parallel processing system makes the swarm robust to intermittent communication. With each missile carrying its own processing, heterogeneous swarms can be formed very easily using missiles with different performance capabilities, as the missiles only need detailed models of themselves and not the other members of the swarm.

Section 2 describes the missile and guidance model used in this paper, section 3 describes the structure and application of the OLEA, section 4 describes example scenarios and demonstrates the behaviour of the OLEA swarm guidance and section 5 concludes.

2. MISSILE MODEL

2.1 Missile Model

The missile model used in this paper is simple, but may be representative of a low-cost missile. A simple fixed wide-field-of-view Imaging Infra-Red seeker is assumed, and simple bang-bang control, creating a non-linear system. The scenario is that the missiles will be gun launched with a muzzle velocity $V \approx 600m/s$ and have no propulsion of their own. Drag is assumed along with a velocity dependent drag coefficient. The magnitude of the lateral acceleration, a_l, when the fin is not in the zero position, is given by (1), where L is a constant coefficient of approximately $L \approx 98/330$ giving $10g$ lateral acceleration at a speed of Mach 1.

$$a_l = L|V_f| \tag{1}$$

It is assumed that although the drag is proportional to velocity squared, the drag coefficient decreases linearly with velocity. The resulting longitudinal acceleration, a_f, is given by (2), where C is a constant coefficient of approximately $C \approx 0.005$, and $L_d \approx 0.1$ is a lift-drag coupling coefficient.

$$a_f = -C|V_f| - |a_l|L_d \tag{2}$$

To keep processing speed high, a simple fixed step integration process was used with a time step of 0.1 seconds for the missile simulations, and a 0.2 second step for the 'internal' trajectory predictions. The large step size can cause a rapid build up of numerical errors, so (3) is used to try to mitigate many large errors.

$$\begin{aligned} |V_{t+1}| &= |V_t| + a_f \delta t \\ \angle V_{t+1} &= \angle V_t + \frac{a_l}{|V_t|}\delta t \\ P_{t+1} &= P_t + V_t \delta t \end{aligned} \tag{3}$$

2.2 Guidance Heuristics

For simplicity, a *pursuit* guidance system has been used (Zarchan, 1997). Here the missile simply steers towards the current target position. The missile will steer towards the way-point until the distance to the way-point increases. When the way-point is active, the missile will then steer towards the current selected target. Although the missile is assumed to be estimating target velocity and acceleration and so could easily use more complex laws such as Augmented Proportional Navigation, pursuit guidance often results in a near tail-chase scenario and as such, the OLEA will have to work much harder in order to provide any useful shaping for the trajectory.

The control surface is assumed to be a fin with bang-bang control. The fin is assumed to have a zero latax central position and only moves if the missile's forward velocity vector is more than 1° from the estimated line-of sight to the target. The fin then pulls maximum available latax (1) until the line-of-sight to the target is within 1° of the velocity vector. The line-of-sight to the target may not be equal to the the seeker pointing angle if the missile is tracking one target but engaging another, or flying towards the way-point.

3. ON-LINE EVOLUTIONARY ALGORITHM (OLEA)

3.1 Introduction

Evolutionary Algorithms are optimisation procedures which operate over a number of cycles (generations) and are designed to mimic the natural selection process through evolution and survival of the fittest (Deb, 2001). A *population* of M possible solutions is maintained by the algorithm. Each potential solution is represented by one *chromosome*. This is the genetic description of the solution and may be broken into n sections called *genes*. Each gene represents a single parameter in the problem, therefore a problem that has eight unknowns, for example, would require

a chromosome with eight genes to describe it. The chromosome could be represented as vector $\vec{Q}$ where the elements of the vector are the genes. Each trial solution forms a single point in the parameter space.

The three simple operations found in nature: natural selection, mating and mutation are used to generate new chromosomes and therefore new potential solutions. Each chromosome is evaluated at every generation using an *objective function* that is able to distinguish good solutions from bad ones and the chromosome's performance is assigned a score. With each new generation, some of the old chromosomes are removed to make room for the new improved offspring. Despite being very simple to code, requiring no directional or derivative information from the objective function and being capable of handling a large number of parameters simultaneously, Evolutionary Algorithms can achieve excellent results.

3.2 OLEA structure

The basic OLEA is described in the algorithm:

(1) $\chi = U({}^{L}\chi, {}^{H}\chi)$
(2) $o_t = F(\chi_t, Z_t)$
(3) $Y_t = D(\chi_t, o_t, Z_t)$
(4) $\chi'_t = S(\chi_t, o_t)$
(5) $\chi_{t+1} = H(\chi'_t, Z_t)$
(6) $t = t + 1$, if not end, repeat from step 2.

In the algorithm, χ_t represents the array of chromosomes at time t. In systems where the chromosome is a fixed length vector, χ would be a matrix with one row for each member of the population and one column for each gene. The function $U()$ is a random distribution (usually uniform) between the upper and lower bounds of each gene value. Thus the first line of the algorithm generates a random population for the OLEA.

The array of objective values and constraint satisfaction indices at time t is represented by o_t. The objectives are calculated based on the model described by the function $F()$, the population χ_t and the observation of the environment Z_t. In general there will be multiple objectives calculated for each member of the population. The algorithm makes the implicit assumption that the measurement Z_t and the function $F()$ are subject to noise and uncertainty.

The OLEA output at time t is denoted by Y_t and is a function of the chromosomes χ_t, the objective and constraint values o_t, the observed state of the environment Z_t and the model function $D()$. In the multiple missile system, the output is the current best way-point and target index. In a system with multiple objectives that evolves a Pareto set, the function $D()$ is the decision maker that must choose one solution from the set to use as the current operating point.

The population update is performed by function $S()$. The function must perform the crossover, mutation and selection process, providing the selective pressure to drive the population towards a good solution. In a multi-objective system, the function must generate and maintain an estimate of interesting regions of the Pareto set. The output is a new population to evaluate in the next generation.

The fifth step uses $H()$ to create the population ready for the next generation by providing a state update if necessary. For example, if position and velocity and acceleration were being estimated from acceleration measurements, $H()$ would update the position from velocity, and the velocity from acceleration ready for the next iteration. Often $H() = 1$ is used where the chromosome remains unchanged between generations. The function $H()$ may also be a local optimisation process to allow hybrid and memetic algorithms to be created, where the chromosome is modified based on the results of a local optimisation process.

The algorithm is repeated, usually with one generation per time increment, until the estimated parameters are no longer required. The framework allows on-line, multiobjective, multi-species, parallel, memetic and hybrid algorithms, as well as very simple evolutionary algorithms to be represented with a wide range of crossover and mutation strategies.

3.3 Cooperative Coevolutionary OLEA

With parallel cooperative coevolution, each missile runs its own OLEA. The chromosome within each missile is of the form $[x \quad y \quad z \quad T]$, where the vector $[x \quad y \quad z]$ is the location of the way-point for the missile and T is the index of the target to aim at after the way-point is passed. The function $F()$ of the OLEA contains a simulation of the dynamic behaviour of the missile, given the input chromosome set χ_t and environmental state Z_t. The environmental state consists of: the current estimated missile position, velocity and seeker state; the current estimate of the parameters of the target of interest; and a set of information from the other missiles indicating their current intended action. Each missile may have its own unique $F()$, allowing a set of heterogeneous missiles to be used to form the swarm, each with a very different dynamic model. It is assumed that a separate data fusion process is used to capture predicted target data from other missiles, and possibly other sensors, to create a unified air-picture, allowing targets to be identified uniquely by an index number.

Each missile broadcasts its current intended action to the other missiles in the swarm ten times each second. The data rate may be reduced if the action has not changed, but at the minimum, a 'keep alive' signal should be sent to indicate the missile is part of the swarm. The data transmitted is the vector $[M \quad \alpha_i \quad t_i \quad T_i \quad |V_i| \quad L_i \quad \delta d_i]$, where M is the missile identification number, α_i is the predicted impact angle, t_i is the predicted impact time, T_i is the index of

the target being engaged, $|v_i|$ is the predicted impact speed, L_i is the predicted lateral acceleration at impact and δd_i is the predicted miss distance.

If the way-point is extreme and the prediction indicates that the missile cannot hit the target, the way-point is ignored and the missile aims at the selected target directly. The objective values are then assessed on the 'default' missile behaviour.

Each missile collects the broadcast information from the other members of the swarm and bases the score of each chromosome on the predicted collective behaviour of the entire swarm. Objectives such as *minimise spread of impact times, minimise longest engagement time, minimise worst latax at impact, maximise spread of impact angles for each target* etc. may be used to govern the behaviour of the swarm. Penalties may also be applied if the missile seeker is locked onto one target, but it intends to engage another. As the seeker will have to break-lock and reacquire the new target once the way-point is passed. There is a risk that the missile will not be able to acquire the new target in time and as such is undesirable behaviour that should be minimised, but not discouraged entirely. If all communication is lost with the other missiles, the platform defaults to heading towards the target that the seeker is currently locked to.

The function $D()$ which selects the solution to implement is simply to take the best performing chromosome and use it to describe the way-point and target to aim at. The missile demand is then derived by the guidance law steering towards the way-point or target if the way-point is not achievable. The state update function $H()$ is unity in this application as the way-point and target index are relative to fixed Earth coordinates, not to the missile body and orientation. In the initial population, $U()$ is a uniform random distribution within a region local to the estimated engagement envelope. A population of 100 has been used in all the simulations. The function $S()$ selects the best 20 solutions for breeding. An approach based on Differential Evolution (Storn and Price, n.d.) is used to generate 80 new solutions to replace the section of the population that was removed.

3.3.1. Differential Evolution Differential Evolution (DE) is an evolutionary technique that uses mutations that are related to the current spatial distribution of the population. The algorithm generates new chromosomes by adding the weighted difference between two chromosomes to a third chromosome. At each generation, for each member of the parent population, a new chromosome is generated. Elements of this new chromosome are then crossed with the parent chromosome to generate the child chromosome.

The size and direction of the difference between any pair of chromosomes is determined by the overall spread of the current population. Thus the DE algorithm self adapts to the fitness landscape, reducing the size of the mutations automatically as the search converges. In this way, no separate probability distribution has to be used for mutation which makes the scheme completely self-organising.

The trial chromosome Q_t may be described as in (4).

$$Q_t = F(Q_a - Q_b) + Q_c \tag{4}$$

Where chromosomes Q_a, Q_b & Q_c are chosen from the population without replacement and F is a scaling factor.

The crossover process is controlled by a crossover parameter C. The crossover region begins at a randomly chosen parameter in the chromosome and then a segment of length G genes is copied from Q_t to the parent chromosome to create the child chromosome. If the segment is longer than the remaining length of the chromosome, the segment is wrapped to the beginning of the chromosome. The length G is chosen probabilistically and is given by (5).

$$P(G \geq k) = (C)^{k-1}, k > 0 \tag{5}$$

In general, the scaling parameter F and the crossover parameter C lie in the range $[0.5, 1]$. Small values of F mean that the population spread reduces faster implying faster convergence. However the faster convergence is more likely to result in the algorithm converging quickly at a *local* minima, rather than the *global* minima. We have found that values of $F = 0.7$ and $C = 0.5$ are suitable for this application.

3.4 Objective Value Calculation

For simplicity, a simple weighted sum with penalty has been used. The weighted sum appears to provide adequate performance, but it is not yet known if the dynamic Pareto front is always convex during the engagement, and therefore multiple solutions could potentially cause oscillation in the decision space. At present the objective equation used is (6), where A is the total range of impact angles across all the missiles, N is the number of missiles with $\delta d_i > 50$ (i.e. definitely miss), S is the number of missiles who will eventually have to break-lock and acquire a new target, T is the longest flight time of the set of missiles and K is the number of missiles that could be hit, but are not targeted by this solution.

$$o = -2A + 1000N + 50S + 50T + 1000000K \tag{6}$$

4. EXAMPLE APPLICATION

The example simulation has three missiles engaging two targets. Figure 1 shows the situation just after launch.

The data fused sensor system of the missiles' was simulated by generating perfect knowledge of the target

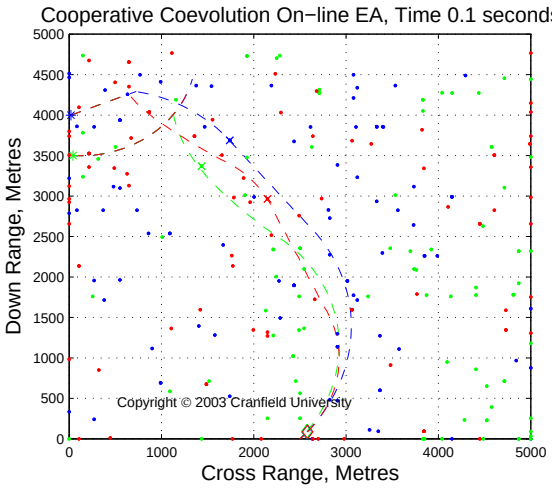


Figure 1. Scenario just after launch

location, velocity and acceleration, and then corrupting the data with Gaussian noise. Thus the forward predictions of the missiles vary considerably with each time step. The aerodynamic coefficients used in each of the local simulation models were corrupted by up to 10% from the true values used in the main missile simulation, thus introducing further errors and uncertainty into the predicted impact conditions. In the figures, the missiles are shown as diamonds, the targets as stars, the solid lines show trajectories actually flown and the dashed lines show the predicted trajectories. The dots indicate the spread of the top 20 way-points of each of the missile populations.

Figures 2, 3 & 4 show the scenario at stages through the engagement. It is clear that the intercept angles range over nearly 180° and that both targets have been engaged successfully. It can also be seen that the CCOLEA populations cluster rapidly and allow smooth trajectories to be formed for the missiles.

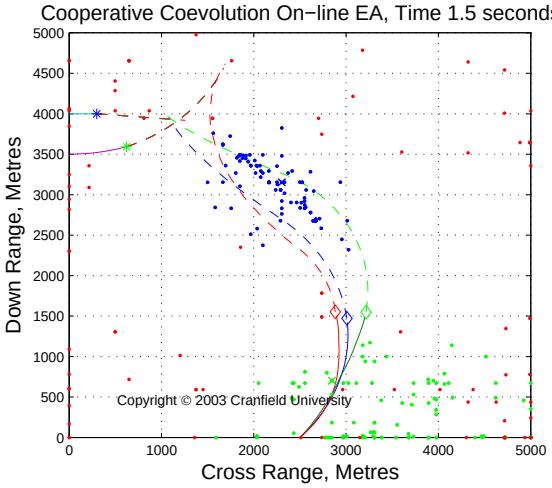


Figure 2. Scenario during engagement at 1.5 seconds

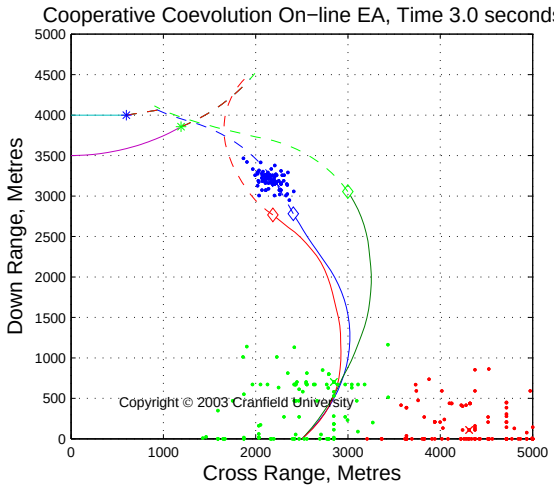


Figure 3. Scenario during engagement at 3 seconds

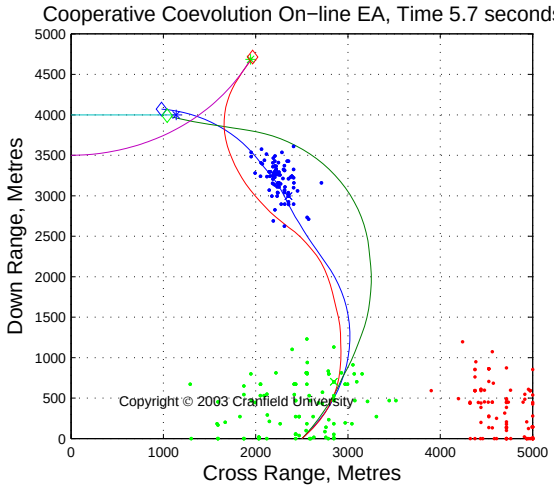


Figure 4. Final trajectories at 5.7 seconds

5. CONCLUSIONS

The CCOLEA approach provides a comprehensive framework allowing multiple missiles to coordinate attacks on single or multiple targets. The framework also allows for data from multiple sensors to be fused easily, as the guidance requires estimates of missile and target positions etc. in absolute coordinates to be used. The results were generated using a highly non-linear missile, combined with noisy measurements and uncertain system models. This shows clearly that the method is tolerant of complexity and many sources of error.

The parallel distributed processing allows todays high-speed processors to be exploited to the full. The larger the population size used, and the more accurate the missile simulations, the better the guidance will perform. As each missile only has to predict its own behaviour, swarms of heterogeneous swarms can be formed very easily.

REFERENCES

Deb, Kalyanmoy (2001). *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.

Hughes, Evan J. (2002*a*). Evolutionary guidance for multiple missiles. In: *IFAC World Congress Conference*. Barcelona, Spain. p. Paper 1801.

Hughes, Evan J. (2002*b*). Multi-objective evolutionary guidance for swarms. In: *World Congress on Computational Intelligence*. IEEE. Honolulu, Hawaii. pp. 1127–1132.

Storn, Rainer and Kenneth Price (n.d.). Differential Evolution- a simple and effective adaptive scheme for global optimization over continuous spaces.
http://http.ICSI.Berkeley.edu/storn/code.html.
last accessed February 2004.

Zarchan, Paul (1997). *Tactical and Strategic Missile Guidance*. 3rd ed.. American Institute of Aeronautics and Astronautics Inc. ISBN 1-56347-254-6.



https://dspace.org/

On-line evolutionary algorithm guidance for multiple missiles against multiple targets

Hughes, Evan J.; White, Brian A.

2004-06-18
Elsevier
https://doi.org/10.1016/S1474-6670(17)32325-X
Evan J. Hughes and Brian A. White. On-line evolutionary algorithm guidance for multiple missiles against multiple targets. IFAC Proceedings Volumes, Volume 37, Issue 6, June 2004, Pages 1091-1096

Downloaded from a DSpace repository. https://dspace.org/