Co-evolutionary Data Mining in Phase Space for Robust Fuzzy Resource Management

Dr. James F. Smith III
Naval Research Laboratory, Code 5741
Washington, D.C., 20375-5000
Telephone: 202.767.5358
Fax: 202.404.7690
jfsmith@drsews.nrl.navy.mil

Abstract-*An approach is being explored that involves embedding a fuzzy logic based resource manager in an electronic game environment. Game agents can function under their own autonomous logic or human control. This approach automates the data mining problem. The game automatically creates a cleansed database reflecting the domain expert's knowledge, it calls a data mining function, a genetic algorithm, for data mining of the data base as required and allows easy evaluation of the information extracted. The co-evolutionary fitness functions, chromosomes and stopping criteria for ending the game are discussed. The strategy tree concept and its relationship to co-evolutionary data mining are examined as well as the associated phase space representation of fuzzy concepts. Co-evolutionary data mining alters the geometric properties of the overlap region known as the admissible region of phase space significantly enhancing the performance of the resource manager. Significant experimental results are provided.*

Keywords: data mining, knowledge discovery, fuzzy logic, genetic algorithms, resource manager.

1 Introduction

Modern naval battleforces generally include many different platforms, e.g., ships, planes, helicopters, etc. Each platform has its own sensors, e.g., radar, electronic support measures (ESM), and communications. The sharing of information measured by local sensors via communication links across the battlegroup should allow for optimal or near optimal decisions. The survival of the battlegroup or members of the group depends on the automatic real-time allocation of various resources.

A resource manager (RM) based on fuzzy logic has been developed that automatically allocates electronic attack (EA) resources in real-time over a battlegroup of dissimilar platforms. The particular approach to fuzzy logic that is used is the fuzzy decision tree, a generalization of the standard artificial intelligence technique of decision trees [1].

The controller must be able to make decisions based on rules provided by experts. The fuzzy logic approach allows the direct codification of expertise forming a fuzzy linguistic description [2], i.e., a formal representation of the system in terms of fuzzy if-then rules. This has proven to be a flexible structure that can be extended or otherwise altered as doctrine sets, i.e., the expert rule sets change.

The fuzzy linguistic description builds composite concepts from simple logical building blocks known as root concepts through various logical connectives: "and", "or", etc. Optimization is used to determine the parameters that control the shape of the fuzzy root concept membership functions.

The optimization procedures employed here are a type of data mining. Data mining is defined as the efficient discovery of valuable, non-obvious information embedded in a large collection of data [3]. The genetic optimization techniques used here are efficient, the relationship between parameters extracted and the fuzzy rules are certainly not a priori obvious, and the information obtained is valuable for decision-theoretic processes. Also, the algorithm is designed so that when the scenario databases change as a function of time, then the algorithm can automatically re-optimize allowing it to discover new relationships in the data. The RM can be embedded in a computer game that EA experts can play. The game records the result of the RM and the expert's battle, automatically assembling a database of scenarios. After the end of the battle, the game makes a determination of whether or not to re-optimize the RM using the newly extended database.

To be consistent with terminology used in artificial intelligence and complexity theory [4], the term "agent" will sometimes be used to mean platform, also a group of allied platforms will be referred to as a "meta-agent". Finally, the terms "blue" and "red" will refer to "agents" or "meta-agents" on opposite sides of a conflict, i.e., the blue side and the red side.

Section 2 briefly introduces the ideas of fuzzy set theory, fuzzy logic, and fuzzy decision trees. Section 3 discusses optimization with a focus on genetic algorithms and co-evolutionary data mining. Section 4 advances a theory that allows automatic construction of co-evolutionary fitness functions. Section 5 references validation efforts. Section 6 discusses approaches and results for co-evolutionary data mining experiments. Finally, section 7 provides a summary.

2 A brief introduction to fuzzy sets and fuzzy logic

The resource manager must be able to deal with linguistically imprecise information provided by an expert. Also, the RM must control a number of assets and be flexible enough to rapidly adapt to change. The above requirements suggest an approach based on fuzzy logic. Fuzzy logic is a mathematical formalism that attempts to imitate the way humans make decisions. Through the concept of the grade of membership, fuzzy set theory and fuzzy logic allow a simple mathematical expression of uncertainty. The RM will require a mathematical representation of domain expertise. The decision tree of classical artificial intelligence provides a graphical representation of expertise that is easily adapted by adding or pruning limbs. Finally, the fuzzy decision tree, a fuzzy logic extension of this concept, allows easy incorporation of uncertainty as well as a graphical codification of expertise.

This section will develop the basic concepts of fuzzy sets, fuzzy logic and fuzzy decision trees. The parameterization of root and composite concepts are discussed.

2.1 Fuzzy set theory

This subsection provides a basic introduction to the ideas of fuzzy set theory. Fuzzy set theory allows an object to have partial membership in more than one set. It does this through the introduction of a function known as the membership function, which maps from the complete set of objects *X* into a set known as membership space. More formally, the definition of a fuzzy set [5] is

> If *X* is a collection of objects denoted generically by *x* then a fuzzy set *A* in *X* is a set of ordered pairs:

$$A = \{(x, m_A(x)) | x \in X\} \tag{1}$$

$m_A(x)$ is called the membership function or grade of membership (also degree of compatibility or degree of truth) of ***x*** in *A* which maps *X* to the membership space *M*.

2.2 Fuzzy decision trees

The particular approach to fuzzy logic used here is the fuzzy decision tree. The fuzzy decision tree is an extension of the classical artificial intelligence concept of decision trees. The nodes of the tree of degree one, the leaf nodes, are labeled with what are referred to as root concepts. Nodes of degree greater than unity are labeled with composite concepts, i.e., concepts constructed from the root concepts [6] using "and", "or", and "not". For "and" and "or" the standard "min" and "max" operations are used, respectively. One minus a membership function represents the action of the modifier "not". Each root concept has a fuzzy membership function assigned to it. The membership functions for composite concepts are constructed from those assigned to the root concepts using fuzzy logic connectives and modifiers. Each root concept membership function has parameters that are determined by optimization as described in section 3.

2.3 Example root concept membership function

For each root concept, a fuzzy membership function must be specified. There is not an a priori best membership function so a reasonable mathematical form is selected. This subjective membership function will be given in terms of one or more parameters that must be determined. The parameters may be set initially by an expert or they may be the result of the application of an optimization algorithm. The use of a genetic algorithm to determine the unknown parameters in root concept membership functions is discussed in section 3. The RM has many root and composite concepts associated with it. Four such concepts are discussed below. They are "close," "heading-in," "ranging" and "banking."

As an example of a membership function definition consider the root concept "close". The concept "close" refers to how close the target/emitter on track *i* is to the ship, or more generally platform of interest. The universe of discourse will be the set of all possible tracks. Each track *i* has membership in the fuzzy set "close" based on its range R_i (nmi) and range rate dR_i/dt (ft/sec). The membership function is

$$m_{close}(i) = \frac{1}{1 + a \cdot \dfrac{\max(R_i - R_{\min}, 0)}{\max(-\dot{R}_i, \dot{R}_{\min})}}. \tag{2}$$

The parameters to be determined by data mining are

$$a, \quad R_{\min}, \quad \text{and} \quad \dot{R}_{\min}. \tag{3}$$

A concept analogous to "close" is the fuzzy concept "heading-in." Its membership function is a function of the heading angle, $Q_{H,i}$, and the first time derivative of the heading angle with respect to time, $dQ_{H,i}/dt$,

$$m_{heading-in} = \frac{1}{1 + b\left|Q_{H,i} - Q_{HIN}\right| / \max\left(\dot{Q}_{H,i}, \dot{Q}_{HIN}\right)}. \tag{4}$$

The parameters to be determined by data mining are

$$b, \Theta_{HIN}, \quad \text{and} \quad \dot{\Theta}_{HIN}. \tag{5}$$

Ranging is a root concept that has a strong relationship to "close." The membership function for the concept "ranging" is a function of the second time derivative of the range as given below,

$$m_{ranging} = \frac{1}{1 + d \Big/ \max\left(\left|\ddot{R}_i\right|, a_{\min}\right)}. \tag{6}$$

The two parameters to determine through data mining for ranging are

$$d \quad \text{and} \quad a_{\min}. \tag{7}$$

The root concept "banking" has a strong relationship to "heading-in" analogous to the ranging's relationship to "close." The membership function of the concept "banking" is a function of the second time derivative of the heading angle as given below,

$$m_{banking} = \frac{1}{1 + c \Big/ \max\left(\left|\ddot{\Theta}_{H,i}\right|, \ddot{\Theta}_{HIN}\right)}. \tag{8}$$

The two parameters to determine through data mining for banking are

$$c \quad \text{and} \quad \ddot{\Theta}_{HIN}. \tag{9}$$

3 Genetic algorithm based optimization and data mining

The parameters of the root concept membership function are obtained by optimizing the RM over a database of scenarios using a genetic algorithm [6,7] (GA). Once the root concept membership functions are known, those for the composite concepts [6] follow immediately. At this point the necessary fuzzy if-then rules for the RM have been fully determined. A detailed discussion of the GA for data mining as well as the construction of the chromosomes and fitness functions are given in the literature [6].

The application of the genetic algorithm is actually part of the second step in a three-step data mining process. The first step is the collection of data and its subsequent filtering by a domain expert, to produce a scenario database of good quality. The second step involves the use of various data mining functions such as clustering [8] and association [9], etc. During this step, genetic algorithm based optimization is used to mine parameters from the database. These parameters allow the fuzzy decision tree to form optimal conclusions about resource allocation. In the third and final step of the data mining operation, the RM's decisions are analyzed by a domain expert to determine their validity.

One approach to constructing a database for re-optimization involves embedding the RM in a computer game, referred to as the scenario generator (SG). The game is designed so human EA experts can play it, in real-time against the RM. This approach is referred to as the human vs. computer (HVC) mode. The human player can control any of the red agents, but only one red agent per time step. The other red agents run under their own autonomous logic different from the blue RM.

The game also allows the RM to be matched against computerized red opponents running under their own autonomous logic with none of the red platforms controlled by a human player. This approach is referred to as computer vs. computer (CVC) mode.

The game software records the events of the game in both HVC and CVC modes. This record contributes to a database for re-optimization, allowing the RM to learn from human or computer opponents. Such a database is purer than one born of sensor data since environmental noise, sensor defects, etc., are not contaminating the data. This offers the advantage that the filtering stage of the data mining operation is simplified, but may result in a database that lacks some real world characteristics.

3.1 Co-evolution

In nature a system never evolves separately from the environment in which it is contained. Rather, both the biological system and its environment simultaneously evolve. This is referred to as co-evolution [10]. In a similar manner, the fuzzy resource manager should not evolve separately from its environment, i.e., enemy tactics should be allowed to simultaneously evolve. Certainly, in real world situations if the enemy sees the resource manager employ a certain range of techniques, they will evolve a collection of counter techniques to compete more effectively with the resource manager.

In a previous paper [6] an approach to co-evolution involving averaging over a database of military scenarios was reported. The current approach involves both blue and computerized red meta-agents each having fuzzy decision trees and strategy trees. Both types of tree will be subject to adaptation during optimization. A strategy tree differs from a decision tree in that it is one meta-agent's model of another meta-agent's decision tree.

Both CVC and HVC are co-evolutionary types of optimization. In CVC mode both red and blue meta-agent decision and strategy trees evolve [11]. In HVC mode, the blue meta-agents and computerized red agents decision and strategy trees evolve. The human player also evolves since in playing many games against the RM, he or she learns new techniques for dealing with the blue RM.

It is observed that HVC based co-evolution converges more rapidly than the equivalent approach in CVC mode [11]. The reason for this appears to be that the human player quickly exhaust their expertise, resulting in an RM that rapidly learns to beat human experts. In CVC based co-evolutionary optimization, membership function parameters fluctuate much longer than in HVC optimization suggesting that a computerized red agent provides a much more flexible enemy. Unlike the human player, the computerized red agent can exhibit many more strategies, instead of fixating on a small collection of techniques born of limited experience. A tentative conclusion is that the greater number of strategies that a computerized enemy can manifest forces the RM to be more adaptive and robust.

3.2 Strategy tree approach to co-evolutionary data mining

The approach to co-evolution is as follows. For each root concept membership function on the red strategy tree define a threshold, such that if the membership function exceeds this threshold and if red's strategy tree is a good representation of blue's decision tree, then red's intention is signaled to blue resulting in an action by blue. The membership function is typically a function of some physically measurable quantity *O* and its first derivative in time, *dO/dt*, or a function of the second time derivative of *O* alone. The two dimensional space resulting from plotting *dO/dt* vs. *O* is a phase space. The inequality between the root concept membership function and its threshold, upon inversion will give inequalities linear in *O* and *dO/dt*, typically. The resulting system of inequalities defines a region of phase space referred to as the admissible region where red can engage in activities without signaling its intent to blue. The membership function parameters that are found through data mining determine the boundaries of the admissible region of phase space. The admissible region can not in general be brought to zero area otherwise blue will carry out an action against everything it detects, resulting in fratricide and wasting valuable resources essential to its survival.

The region in Figure 1 below the range axis and to the right of lines QC and CB excluding the boundary defined by parabola DOFEH is the admissible region of the range-rate vs. range phase space determined using the above procedure for the root concept "close" on red's strategy tree. It is assumed red knows the exact mathematical form of blue's fuzzy membership function for "close", but red only knows the parameters for "close" approximately. Quantities with "*i*" subscripts refer to the i^{th} track, the *k* subscripts indicates values at the k^{th} time step, and "*0*" subscripts are used on red's initial values of range and range-rate. The symbol τ is the threshold that red's grade of membership in "close" should be less than, so as to not signal red's intent to blue. The quantity *d* is the desired distance red would like to be from blue before executing an action. This is referred to as the goal distance.

The RM has many concepts that overlap conceptually in phase space reducing the likelihood that red can win against blue. Each concept gives rise to an admissible region in a two-dimensional phase subspace. In a particular phase subspace, the overlap of various fuzzy concepts gives rise to an overall admissible region for the RM that is an intersection of the individual regions related to each fuzzy concept. This overall admissible region is referred to as the *combined admissible region*. In this way, by having many overlapping fuzzy concepts, the area of phase space that red can occupy without alerting blue is made smaller. Since sensors have finite resolution and time is considered discrete a smaller phase space area implies a smaller number of trajectories or patterns of behavior that red can execute safely, resulting in a more effective RM.

The fuzzy concepts “close” and “ranging” overlap in the range-rate versus range phase subspace. The parabolic arc OFEH in Figure 1 is a potential red trajectory that exists within the admissible regions defined by “close” and “ranging.” The combined admissible region for “close” and “ranging” for a given maximum radar range has an area much less than that of the admissible region of “close” alone [12].

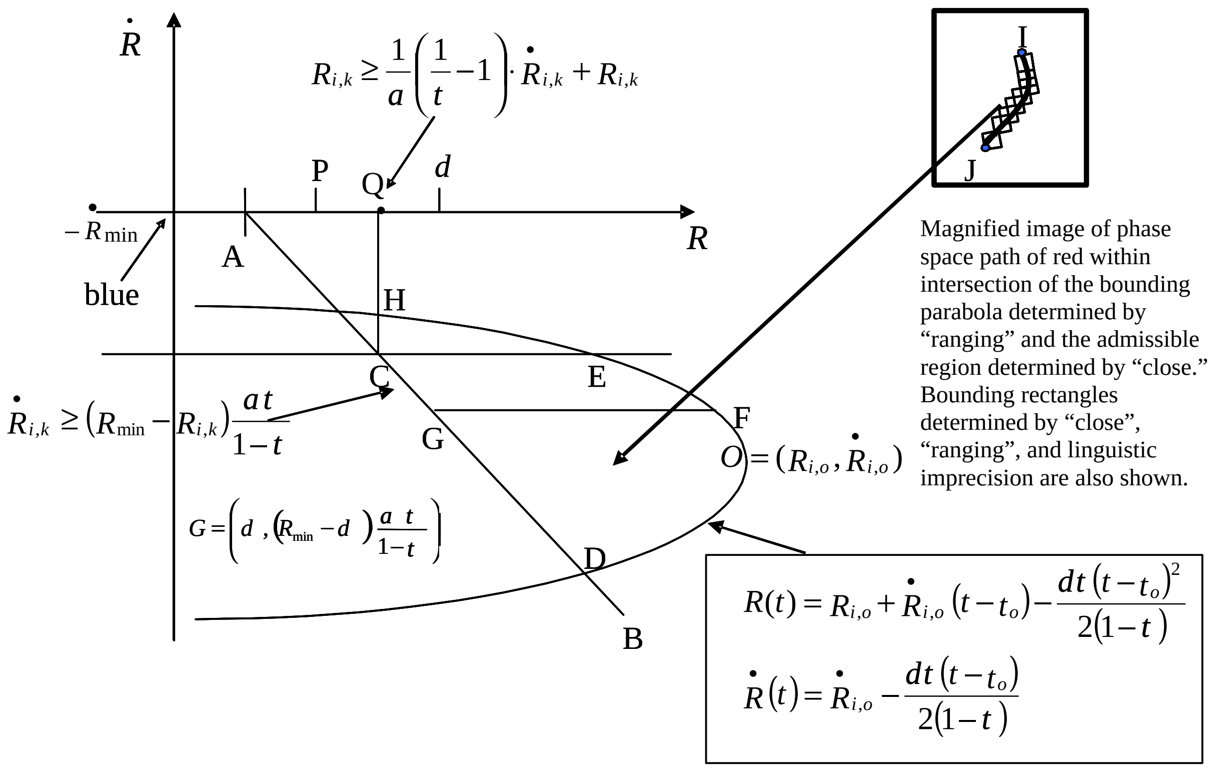

Figure 1: Trajectories within the combined admissible region of phase space determined by “close” and “ranging.”

The trajectories red may safely follow through this phase subspace are further limited. The combined admissible region gives the collection of trajectories red may follow safely, for many different possible initial conditions. Once red's initial conditions are established his change in range or range-rate is limited each time-step by "close" and "ranging." This is represented in Figure 1 in the upper right-hand-side subplot by rectangles along trajectory IJ that bound the possible points in phase space that red can safely move to in the next time step. When red determines these bounding rectangles, he must take "close" and "ranging" into account and also an estimate of his uncertainty in the parameters of both concepts. This uncertainty in parameter values is quantified by using fuzzy number theory [2].

Using the same reasoning as for the fuzzy concept "close," a phase space diagram can be constructed for "heading-in." In the heading-in vs. heading space, "banking" plays a role analogous to "ranging" in the range-rate vs. range space. Similar phase spaces exists for other concepts in the RM.

3.3 Stopping criteria for co-evolution

Just as with a genetic algorithm, in co-evolutionary game based data mining, stopping criteria must be defined. The stopping criteria used here were that a maximum number of co-evolutionary generations had been reached, or that one side, red or blue, had won a certain number of games in a row.

3.4 Chromosome structure for co-evolution

The chromosomes for co-evolution are vectors whose elements are the parameters of the fuzzy membership functions for the fuzzy concepts "close," "heading-in", "ranging," and "banking." The parameters for these functions are discussed in section 2.3.

4 Automatic construction of a fitness function for co-evolution

When re-optimizing, it is necessary to incorporate knowledge of an agent's history, specifically those events that led to re-optimization. A method of doing this is to construct fitness functions that contain a history of the agent and upon maximization result in agents that will not reproduce past mistakes. This subsection develops an algorithm for the automatic construction of such functions. Let $\mu_{attacking}$ refer to the output of blue's isolated platform decision tree (IPDT) or red's strategy tree that attempts to reproduce the decisions made by blue's IPDT [12]. In the development below Heaviside step functions are approximated by sigmoidal functions as described in the literature [13]. A logarithm is taken to produce a smoother fitness function. Replacing the Heaviside step function with the sigmoid produces a smoother approximation to the original fitness function. The sigmoid based fitness function carries approximately the same data base information as the one based on Heaviside step functions. Since it falls off more slowly, a chromosome with many good traits that would have had a zero probability of surviving in the population under the fitness function based on Heaviside step function now has a non-zero probability of survival. Taking the logarithm of the product of the sigmoidal functions also helps to accelerate convergence in some cases.

4.1 Red fitness

Let N_G be the number of co-evolutionary generations; and T, be a vector of length L_T containing the time-steps that will be used in the fitness calculation, up to and including the last time-step in each co-evolutionary generation. In practice, this vector contains the time-steps at which a blue platform detected a red platform on radar. Let $m_{t,g}$ be red's estimate of the value of blue's fuzzy grade of membership for "attacking", i.e., $m_{attacking}$ at time t in co-evolutionary generation g, and let t be the threshold value for $m_{attacking}$. The quantities b_1 and b_2 are constants that determine the weights of the next to last L_T - 1 time-steps and the last time-step, respectively. The parameter, lb, is introduced to establish a lower bound and C is a constant which ensures that the fitness is positive.

Different formulas are used depending on whether the position data being employed to evaluate the fitness comes from a game in which red won or one in which blue won. If blue won, then $m_{attacking}$ must have been above threshold at the last time-step, but below threshold at previous time-steps. In this case, the red fitness is given by:

$$fit = \sum_{g=1}^{N_G}\left\{\max\left[lb, \log\left(\frac{1}{1+\exp\left[b_2\left(t - m_{T(L_T),g}\right)\right]}\right)\right]\right\} + \sum_{g=1}^{N_G}\left\{\sum_{t=1}^{L_T-1}\max\left[lb, \log\left(\frac{1}{1+\exp\left[b_1\left(m_{T(t),g} - t\right)\right]}\right)\right]\right\} + C \quad (10)$$

If red won, then $m_{attacking}$ was never above threshold, and the red fitness is given by:

$$fit = \sum_{g=1}^{N_G}\sum_{t=1}^{L_T}\max\left[lb, \log\left(\frac{1}{1+\exp\left[b_1\left(m_{T(t),g} - t\right)\right]}\right)\right] + C \quad (11)$$

If there are multiple blue platforms, a red platform's fitness is the sum of its fitness scores for each blue platform. If there are multiple red platforms, the red meta-agent fitness is the sum of all individual fitness values. Finally, for all forms of the red fitness function the b_1 and b_2 values currently being used are $b_1 = 10^3$ and $b_2 = 10^6$.

4.2 Blue fitness

Let T_R and T_N be vectors with L_R and L_N elements, respectively, containing the time-steps that will be used in the fitness calculation. The vector, T_R ,contains those time-steps at which a blue platform detected a red platform on radar, and T_N contains the time-steps at which a blue platform detected a neutral platform. All other notation is the same as above with all quantities referring to blue's decision trees. The blue fitness is given by:

$$
\begin{aligned}
&fit = C + \\
&\sum_{g=1}^{N_G} \left\{ \sum_{t=1}^{L_R} \max\left[lb, \log\left(\frac{1}{1+\exp\left[b_1\left(t - m_{T_R(t),g}(red)\right)\right]} \right)\right]\right\} \qquad (12) \\
&+ \sum_{g=1}^{N_G} \left\{ \sum_{t=1}^{L_N} \max\left[lb, \log\left(\frac{1}{1+\exp\left[b_2\left(m_{T_N(t),g}(neu) - t\right)\right]} \right)\right]\right\}
\end{aligned}
$$

The arguments "red" and "neu" refer to membership function values associated with a red agent and neutral agents, respectively. If there are multiple red platforms, a blue platform's fitness is the sum of its fitness scores for each red platform. If there are multiple blue platforms, the team fitness is the sum of all individual fitness values.

For the blue fitness function, the b_1 and b_2 values vary based on which team won the game that is providing the data to evaluate the fitness function, and what circumstances caused that team to win. If red won by reaching the goal distance, *d*, then $b_1 = 10^4$ and $b_2 = 10^2$. If red won because blue attacked a neutral platform, then $b_1 = 10^2$ and $b_2 = 10^4$. If blue won, then $b_1 = 10^3$ and $b_2 = 10^3$.

In both fitness functions,

$$lb = -15 \ . \qquad (13)$$

A constant, *C*, is also added to all fitness values after an entire generation's fitness has been calculated to insure that all fitness values are positive.

4.3 Fitness with multiple red strategies

Let there be *N* red strategies. For each strategy, a game is played and an extension of the database is made. Let $F_j(i)$ be the fitness of the *i*th individual in the GA's population as calculated by the above functions using a database subset taken from a game in which the *j*th red strategy was used. The fitness of the *i*th individual over all *N* red strategies is given by:

$$fit(i) = \frac{1}{N} \sum_{j=1}^{N} \frac{F_j(i)}{\max(F_j)} \qquad (14)$$

4.4 Red strategies

In order to prevent blue from executing an action against red, red must behave in such a way that none of the blue membership functions exceed threshold, three red strategies for doing this are discussed below. The blue membership functions partition phase space into two disjoint regions made up of a combined admissible region, the region where the grade of membership is below threshold and no action is taken, and an inadmissible region, where the grade of membership is above threshold and a blue action is executed.

For this example, red uses three different trajctories to remain within the combined admissible region. For trajectory one, red begins at point O in Figure 1. Red follows arc OE, then line EC until the goal range, *d* is reached. For trajectory two, red begins at point O and follows arc OD, then line DA until the goal range, *d* is reached in Figure 1. Finally, trajectory 3 consists of red following line FG in Figure 1.

In all three trajectories, the heading angle assumes a value of p as red approaches the goal distance, *d*. Red can do this without initiating a blue action since blue's radar rotates and as such is not always looking at red.

There are many possible strategies that red can pursue in phase space. A more detailed analysis of potential strategies will be considered in a future publication.

5 Validation of the RM, information data mined and automatic rule discovery

The third step of the data mining problem involves validation, i.e., determination of the value of the information data mined. This is intrinsically coupled to the validation of the resource manager itself. Both data mined information and the RM have been subjected to significant evaluations using the scenario generator [11,12]. Through this process the data mined information has been shown to be extremely valuable and the decisions made by the RM of the highest quality.

6 Experimental results

The following simple experiment uses the fuzzy concepts "close," "heading-in," "ranging," and "banking" to illustrate the co-evolutionary approach. Red starts 28.28 nautical miles from blue. For this simple experiment the three red strategies of subsection 4.4 are used against blue. Blue is considered stationary throughout the experiment; and neutral travels a curved trajectory, but never gets closer to blue than 26 nautical miles. Also, neutral's heading is never directly toward blue: it is always off by at least three degrees.

In this data mining process there were 30 co-evolutionary generations. Of these 30, blue won 24 and red won six. Within the first 14 co-evolutionary generations blue's and red's α's converge to the same value. This implies that red learned blue's parameter. Also, it can be shown mathematically that the values of the final parameters of "close" selected for blue result in a RM that red can not beat when there is one blue agent, one red agent and one neutral agent. This mathematical procedure will be described in greater detail in a future publication.

Both red and blue have similar values of R_{min} at the end of the co-evolutionary data mining process. The values of R_{min} for both red and blue stop changing by the 19^{th} generation, it is observed that red learns blue's parameter within an acceptable tolerance. The other eight parameters in the chromosomes show similar behavior when plotted versus the number of co-evolutionary generations.

7 Summary

An approach is being explored that involves embedding a fuzzy logic based resource manager (RM) in an electronic game environment. Game agents can function under their own autonomous logic or human control. During the game both blue agents and red agents simultaneously evolve. This process is referred to as co-evolution. The blue agents are exclusively controlled by the RM. The red agents, the enemies of the blue force, are controlled by their own logic different from the RM. This process evolves a resource manager that is extremely robust. The robustness of the RM arises from the formidable red agents that are created by this process. This approach automates the data mining problem. The game automatically creates a cleansed database reflecting the domain expert's knowledge, it calls a data mining function, a genetic algorithm, for data mining of the data base as required and allows easy evaluation of the information extracted. The co-evolutionary fitness functions, chromosomes and stopping criteria for ending the game are discussed for both red and blue agents. The fitness function takes into account the presence of neutral platforms. Genetic algorithm based data mining procedures are discussed that automatically discover new fuzzy rules and strategies. The strategy tree concept and its relationship to co-evolutionary data mining are examined as well as the associated phase space representation of fuzzy concepts. The admissible region is that subset of phase space where a red agent can operate without initiating a reaction by a blue agent. The fuzzy concepts "close" and "ranging" map into the same phase space resulting in a reduction of the area of the combined admissible region of phase space as compared to the admissible region generated by "close," alone. This reduces the strategies available to red that will not result in an action by blue. A similar overlap between the fuzzy concepts "heading-in" and "banking" also serves to make the RM more effective in rapidly identifying threatening red behavior. Co-evolutionary data mining alters the geometric properties of the combined admissible region of phase space significantly enhancing the performance of the resource manager. Procedures for validation of the information data mined are discussed and significant experimental results provided.

Acknowledgements

This work was sponsored by the Office of Naval Research. The author would also like to acknowledge Mr. Joseph A. Blank, Mr. Edward Khoury, Mr. Robert Xander, Dr. Preston Grounds III, Dr. Joseph Lawrence III, and Mr. Zachary Firth.

References

[1] Samuel Blackman and Robert Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Boston, Massachusetts, 1999.

[2] Lefteri H. Tsoukalas and Robert E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, John Wiley and Sons, New York, New York, 1997.

[3] Joeseph P. Bigus, *Data Mining with Neural Nets*, McGraw-Hill, New York, New York, 1996.

[4] John H. Holland, *Hidden Order How Adaptation Builds Complexity*, Perseus Books, Reading, Massachusetts, 1995.

[5] Hans J. Zimmerman, *Fuzzy Set Theory and its Applications*, Chapter 1, Kluwer Academic Publishers Group, Boston, Massachusetts, 1991.

[6] James F. Smith, III and Robert Rhyne, II, "A resource manager for distributed resources: fuzzy decision trees and genetic optimization," In Hamid Arabnia, editor, *Proceeding of the International Conference on Artificial Intelligence, IC-AI'99*, pp 669-675, Las Vegas, Nevada, 26-29. June, 1999, CSREA Press, Athens, Georgia, 1999.

[7] David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Massachusetts, 1989.

[8] James F. Smith III, "A Fuzzy Clustering and Superclustering Scheme for Extracting Structure from Data," NRL Formal Report, NRL/FR/5740--96-9844, Naval Research Laboratory, Washington D.C. 20375-5000, 1996.

[9] James F. Smith III, "Fuzzy Logic Association: Performance, Implementation Issues, and Automated Resource Allocation," In Ivan Kadar, editor, *Signal Processing, Sensor Fusion, and Target Recognition VIII*, Vol,. 3720, pp. 228-238, Orlando, Florida, 5-9, April 1999 SPIE Proceedings, Bellingham, Washington, USA, 1999.

[10] D. Cliff and G. F. Miller, "Co-evolution of Pursuit and Evasion II: Simulation Methods and Results," In P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S.W. Wilson, editors, *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96)*, pp. 1-10, 9-13 Sept. 1996, MIT Press Bradford Books, Cambridge, Massachusetts, 1996.

[11] James F. Smith III and Robert D. Rhyne II, "Knowledge discovery through games and game theory", In Belur Dasarathy, editor, *Data Mining and Knowledge Discovery III*, Vol. 4384, pp. 92-103, Orlando, Florida, 16-20, April 2001, SPIE Proceedings, Bellingham, Washington, USA, 2001.

[12] James F. Smith III and Robert D. Rhyne II , "Fuzzy Logic Resource Management and Coevolutionary Game-Based Optimization," NRL/FR/5741--01-10001, Naval Research Laboratory, Washington D.C. 20375-5000, 2001.

[13] James F. Smith III and Robert D. Rhyne II , "Methods of Automated Rule Discovery in a Multi-agent Adversarial Distributed Environment," In Waleed Smari,, editor, *Proceeding of the International Symposium on Information Systems and Engineering, ISE'2001*, pp 25-31, Las Vegas, Nevada, 25-28, June 2001, CSREA Press, Athens, Georgia, 2001.