Fuzzy Rule Determination By Co-evolutionary Data Mining

Abstract - A fuzzy logic based expert system has been developed that automatically allocates electronic attack resources in real-time over many dissimilar platforms. A new approach is being explored that involves embedding the resource manager in an electronic game environment. The game allows a human expert to play against the resource manager in a simulated battlespace with each of the defending platforms being exclusively directed by the fuzzy resource manager and the attacking platforms being controlled by the human expert or operating autonomously under their own logic. This approach automates the data mining problem. The game automatically creates a database reflecting the domain expert's knowledge. It calls a data mining function, a genetic algorithm, for data mining of the database as required and allows easy evaluation of the information mined in the second step. The criterion for re-optimization is discussed. The mined information is extremely valuable as shown through demanding scenarios.

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

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 fuzzy logic algorithm has been developed that automatically allocates electronic attack (EA) resources in real-time. In this paper EA refers to the active use of electronic techniques to neutralize enemy equipment such as radar [1]. The particular approach to fuzzy logic that will be used is the fuzzy decision tree, a generalization of the standard artificial intelligence technique of decision trees [2].

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 [3], i.e., a formal representation of the system in terms of fuzzy if-then rules. This will prove 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 will build composite concepts from simple logical building blocks known as root concepts through various logical connectives: “or”, “and”, etc. Optimization has been conducted to determine the form of the membership functions for the fuzzy root concepts.

The optimization procedures employed here are a type of data mining. Data mining is defined as the efficient discovery of valuable, non-obvious information from a large collection of data [4]. 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. Alternatively, the resource manager (RM) can be embedded in a computer game that EA experts can play. The software records the result of the RM and expert’s interaction, automatically assembling a database of scenarios. After the end of the game, the RM makes a determination of whether or not to re-optimize itself using the newly extended database.
To be consistent with terminology used in artificial intelligence and complexity theory [5], 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 will briefly introduce the ideas of fuzzy set theory, fuzzy logic, fuzzy decision trees, and five major components of the RM. Section 3 discusses optimization with a focus on genetic algorithms and data mining. Section 4 discusses co-evolution, the re-optimization criterion, the co-evolutionary stopping criterion, automatic fitness function construction, and software tools for data mining. Section 5 provides examples of the RM’s response for multi-platform scenarios. Finally, section 6 provides a summary.

2 A brief introduction to fuzzy sets, fuzzy logic and the fuzzy RM

The RM 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 [6]. The RM requires 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. The fuzzy decision tree, a fuzzy logic extension of this concept, allows easy incorporation of uncertainty as well as a graphical codification of expertise [2]. Finally, a detailed discussion of the particular approach to fuzzy logic and fuzzy decision trees used in the RM is given in reference [7].

The resource manager is made up of five parts, the isolated platform model, the multi-platform model, the communication model, the fuzzy parameter selection tree and the fuzzy strategy tree. As previously discussed the isolated platform model provides a fuzzy decision tree that allows an individual platform to respond to a threat. The multi-platform model allows a group of platforms to respond to a threat in a collaborative fashion. The communication model describes the means of communication or interaction between the platforms. The fuzzy parameter selection tree is designed to make optimal or near optimal selections of root concept parameters from the parameter database assembled during previous optimization with the genetic algorithm. Finally, the strategy tree is a fuzzy tree that an agent uses to try to predict the behavior of an enemy. A more detailed discussion of the structure of the RM as well as explicit forms for fuzzy membership functions can be found in [7].

3 Optimization of the root concept’s parameters using a genetic algorithm for 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 (GA) [8]. Once the root concept membership functions are known, those for the composite concepts [7] follow immediately. At this point the necessary fuzzy if-then rules for the RM have been fully determined. Detailed discussions of the GA for data mining as well as the construction of the chromosomes and fitness functions are given in reference [7].

The optimization procedures employed here are a component of a data mining operation. Data mining is defined as the efficient discovery of valuable, non-obvious information from a large collection of data [4]. 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 RM 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 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 and association, etc. During this step, the 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.

Data mining to re-optimize the RM and real-time operation of the RM may occur simultaneously on different computers. Since the RM is designed to operate on a collection of platforms, even during very active use of the RM, some computer resources may be available for additional optimization and other data mining related activities. Thus, the multi-platform scheme allows frequent re-optimization of the RM, while the previously optimized version of the RM continues to function in real-time.

Typically the database is constructed from data taken from sensors of different types. The data will be sparse intermittent and noisy. To assemble a
defines a region of phase space referred to as the admissible region, typically. The resulting system of inequalities 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.

4 Co-evolutionary data mining

In nature a system never evolves separately from the environment that contains it. Both biological systems and environment simultaneously evolve. This is referred to as co-evolution [21]. 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.

4.1 Real-time 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 action by blue. The membership function parameters that are found through data mining determine the boundaries of the admissible region of phase space. 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.

4.2 Tools for visualization of data mined information

To facilitate data mining, co-evolution and validation of the RM, a software tool known as the scenario generator (SG) has been created. It automatically creates simulated blue and red platforms with user defined assets. It also creates a map or battlespace and automatically places the red and blue platforms in this space where they can interact. Each red platform is controlled by its own copy of the fuzzy RM.

The SG has two modes of operation. In the computer vs. computer (CVC) mode each red platform is controlled by its own controller distinct from the fuzzy RM used by the blue platforms. In the second mode, the human vs. computer (HVC) mode, a human player controls a red platform through an interactive graphical user interface (GUI). There can be multiple red platforms. At each time step, the human player can control any of the red platforms, but only one of them per time step. Those red platforms not under human control run under their own logic as in the CVC mode.

From the SG software three different GUI's can be easily accessed. These GUI's are the “scenario builder”, the “map builder,” and the “human control player interface” (HCPI).

The scenario builder GUI allows the construction of blue and red agents with general characteristics. Through this GUI both blue and red agents can be given various assets such as different types of radars, ESM, EA systems, etc. This GUI allows the creation of a terrain map that is discussed below. The scenario created can be placed in a database for further data mining and co-evolutionary analysis. Effects due to weather, system losses, atmospheric attenuation, multi path, clutter, etc., can be included in the calculation, although they can not be currently called from the GUI.

The map builder allows the construction of various maps on which the red and blue agents can interact as the scenarios are played out. The map builder GUI can be called from the scenario generator GUI. The map defines a battlespace that can include various environments such as oceans, forest, deserts, cities and jungles. Maps created by the map builder can be saved in a database for reuse.
4.3 Criterion for re-optimization

The criterion for re-optimization is formulated based upon a determination that a particular parameter set has become ineffective. In the EA community, failure to delay, disrupt, or deny information can be the basis for labeling a parameter set ineffective. Outright loss of platforms is another simple measure of ineffectiveness that can be employed. Kindred to how humans change and replace strategies, the RM uses the re-optimization criterion as a point of re-building and re-construction of strategies. When the re-optimization criterion is triggered, a GA is called to re-optimize the root concepts of the decision and strategy trees.

One simple approach to determining this criterion is an analysis of root concept membership functions' values over time when blue loses to red. If it is determined that blue's failure related to certain root concept membership functions not triggering an action by the RM, these membership functions could be made more sensitive.

After many co-evolutionary generations it is possible that both the blue and red groups will have evolved to the point that they are very effective in dealing with each other, but no longer effective in dealing with agents from past generations. For a real system running the RM, this could be a deadly defect, as it is not uncommon to encounter opposing systems manufactured at many different times.

4.4 Stopping criterion for co-evolution

Just as with a genetic algorithm, in a co-evolutionary game based optimization, a stopping criterion must be defined. Upon completion of the game, several iterations of a particular scenario can be played. A criterion for re-optimization determines when the RM will re-optimize its parameters. This optimization is kindred to the scenario-based optimization discussed in reference [7], however the scenarios optimized over are recordings of the previous games since the last optimization. Since the re-optimization criterion determines how many scenarios the optimization is taking into account, this criterion is non-trivial.

4.5 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, referred to as symbolically recursive fitness functions.

This first step in producing a symbolically recursive fitness function involves multiplying the fitness function used in the previous co-evolutionary generation for blue optimization by a product of Heaviside step functions for the current co-evolutionary generation. The fitness function for the first co-evolutionary generation is formed by multiplying the Heaviside step functions by the fitness function used during the initial genetic algorithm based data mining process, referred to as the zeroth order fitness function.

The product of Heaviside step functions includes one Heaviside step function as a factor for each offending root concept evaluated at each time step since the last re-optimization. The argument of each Heaviside step function is the difference between the offending root concept membership function evaluated appropriately at each time step and a threshold. The resulting product fitness function is referred to as a symbolically recursive fitness function. The idea is that unless the GA optimization produces a root concept membership function which for this set of input data, exceeds the appropriate threshold, the symbolically recursive fitness function will return a value of zero. Using the symbolically recursive fitness function the GA can be used to ensure that the membership function value for a particular root concept will be above a certain threshold. This triggers an appropriate action by the RM the next time red exhibits the behavior that led to blue's loss and subsequent re-optimization. A similar procedure is employed when re-optimizing red.
By evaluating the fitness over the current co-evolutionary generations as well as previous co-evolutionary generations, the resulting parameter sets will be effective for the current red strategy, as well as the previous red strategies. This allows the RM to adapt to current strategies without being vulnerable to previous strategies that could be employed by older red agents.

### 4.6 A simple example of co-evolutionary optimization using the fuzzy concept "close"

This subsection provides a simple example of co-evolutionary data mining using the fuzzy root concept "close." CVC and HVC co-evolution are considered as well as a comparison between the techniques.

The concept "close" refers to how near the target/embrmer on track $i$ is to the ship, or more generally the platform of interest [22-23]. 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 $R$ (nmi) and range rate $dR/dt$ (ft/sec). The fuzzy membership function for "close" takes the form

$$\mu_{close}(i) = \frac{1}{1 - \alpha |R_i - R_{min}| / \max(-\dot{R}_i, \dot{R}_{min})}$$

The parameters to be determined by optimization are $\alpha$, $R_{min}$, and $\dot{R}_{min}$.

The parameters for close were initially determined using a genetic algorithm [22-24]. The fitness function used for initial optimization, i.e., before the beginning of the co-evolutionary process is described in the references. This fitness function is the zeroth order fitness function for co-evolution.

For both HVC and CVC modes a loss by blue results in immediate re-optimization of blue's parameter set. A loss by a computerized red agent results in re-optimization of the red agent's parameter set. The stopping criterion for re-optimization is a maximum number of co-evolutionary generations. A co-evolutionary generation refers to a single battle followed by re-optimization of red or blue.

A blue loss occurs if one of blue's agents is disabled, due to the successful delivery of a red missile. A probabilistic model determines the effectiveness of the fired missile. A blue win occurs if the blue agent group is able to delay red a certain number of time steps $t$. Finally, a red loss occurs if blue wins.

In HVC mode, the human player acting as a red agent can locate blue agents using the PPI display described in subsection 4.2. When a blue agent is located on the screen, the user clicks on the target region and presses the fire button located in the lower right hand corner to launch a missile.

One simple class of experiments that has been conducted consists of one blue agent versus one red agent. It was typically found that all three parameters in blue's version of "close" showed little change for the last 33% of the co-evolutionary generations. The human opponent operating the red agent tended to fixate on the same strategies. This suggested that in HVC optimization the human player quickly reached the limits of his or her expertise resulting in the RM's parameters reaching a constant value. Thus the optimization of blue converged rapidly.

In CVC mode, there is no human player controlling a red agent. The red agents are controlled by their own logic that includes a strategy tree. Each blue agent is controlled by a copy of the RM as in HVC mode. The blue agent's decision tree has the root concept "close" on it. The red agent has a strategy tree with his perception of "close." It is assumed that red has very good intelligence about blue, hence the mathematical form that red is using for "close" is the same as the one blue uses. Red is uncertain about the value of blue's parameters for close and as such how they slightly differ from those of blue.

Both blue and red can change during the co-evolutionary process. Red's parameters for his version of "close" determine the admissible region of phase space that red attempts to occupy so as not to invoke an action by blue. Assuming a given initial position and velocity for red, these parameters in turn determine red's value of acceleration, hence red's trajectory.

In a simple experiment with one blue agent versus one red agent operating in CVC mode convergence was not nearly as fast as in HVC mode. The computer controlled red agent is typically capable of exhibiting many more strategies than the human controlled red agent in HVC mode. Thus the co-evolutionary process ends up exploring the combined red-blue parameter space longer, resulting in a greater likelihood of a global maximum being found for the fitness function, resulting in a RM that is more robust than in the HVC case.

The more robust RM obtained through use of the CVC optimization can be understood intuitively as follows. If red can exhibit more strategies by using CVC mode than in HVC mode then the blue RM is forced to be more adaptive to compete.

There is a risk during co-evolution that with both red and blue co-evolving, they will become very specialized in dealing with each other. For example without taking proper precautions blue agents of the 100th co-evolutionary generation might be effective against red agents of that generation, but ineffective against agents of generations 100 through 999. Fortunately, the structure of
the symbolically recursive fitness function prevents this, because its form retains the past history of the agents, forcing the blue agents of the 1000th generation to be effective against red agents of the preceding or current generation.

5 Examples of multi-platform response

In this section a specific example of the fuzzy RM’s ability to optimally allocate electronic attack resources is examined. Input requirements and output characteristics are considered, and illustrated through the actual output of the current implementation of the RM. Many examples like those included in this section point out that the information data mined is extremely valuable [22-27]. The software described in subsection 4.2 is extremely useful for evaluation of the RM and determination of the value of information data mined in the second data mining step. The natural output of the scenario generator is a computer-generated movie. This subsection includes frames from such a movie for the scenario described below. It illustrates the operation of the RM while the SG runs in the CVC mode. The scenario generator also creates a corresponding database reflecting the RM’s decisions for later analysis and subsequent data mining to improve the RM’s adaptive response. By observing the RM’s operation in movies, it can be seen that the RM’s response is faster and more adaptive to change when the best parameter sets are used.

The following are the events leading up to the fictitious battle. A blue helicopter is downed and a blue platform group is sent in to rescue it. The blue group consists of four ESM/EA equipped ships including: one carrier, one cruiser, and two destroyers. There is also a blue rescue helicopter and an ESM/EA equipped blue support plane. The blue group will encounter a threat mix consisting of three red fighter planes (each with multi-role radars) and two red land based search and acquisition radar (LSAR). Due to the geopolitical diversity of the region, ID’s are only given with 50% certainty. The RM handles uncertainty in ID very well.

Figures 1 and 2 display the simulation created by the scenario builder and map builder for time steps three and seven. Darker regions indicate land and lighter regions water. The particular time step that the picture corresponds to is given at the bottom in the left-hand corner. Each platform is indicated by an asterisk labeled with the platform’s type. The platform’s activity at each time step is displayed next to its type. A blue platform’s jamming process is depicted as a line emerging from the blue platform and ending on the red platform.

A copy of the RM running on each blue platform determines its behavior. An algorithm different from the blue RM controls the red platform’s behavior. The events of the battle are saved in a database for data mining operations and co-evolutionary analysis.

During the first time step (not pictured) the southern red land-based search and acquisition radar acquires the blue rescue helicopter. The southern LSAR communicates this information to the rest of the red group. The blue helicopter is subsequently acquired by the remaining red platforms.

The following are the events leading up to the fictitious battle. A blue helicopter is downed and a blue platform group is sent in to rescue it. The blue group consists of four ESM/EA equipped ships including: one carrier, one cruiser, and two destroyers. There is also a blue rescue helicopter and an ESM/EA equipped blue support plane. The blue group will encounter a threat mix consisting of three red fighter planes (each with multi-role radars) and two red land based search and acquisition radar (LSAR). Due to the geopolitical diversity of the region, ID’s are only given with 50% certainty. The RM handles uncertainty in ID very well.

Figures 1 and 2 display the simulation created by the scenario builder and map builder for time steps three and seven. Darker regions indicate land and lighter regions water. The particular time step that the picture corresponds to is given at the bottom in the left-hand corner. Each platform is indicated by an asterisk labeled with the platform’s type. The platform’s activity at each time step is displayed next to its type. A blue platform’s jamming process is depicted as a line emerging from the blue platform and ending on the red platform.

A copy of the RM running on each blue platform determines its behavior. An algorithm different from the blue RM controls the red platform’s behavior. The events of the battle are saved in a database for data mining operations and co-evolutionary analysis.

During the first time step (not pictured) the southern red land-based search and acquisition radar acquires the blue rescue helicopter. The southern LSAR communicates this information to the rest of the red group. The blue helicopter is subsequently acquired by the remaining red platforms.
Figure 1 depicts the events of time step three. The northern LSAR has acquired prior to this time step the blue support plane and destroyer 1. The blue group determines, based on sensor data, that CTAR systems are active. The fuzzy RM running on the entire blue group directs the support plane to engage in EA against red fighter 1 and the northern red multi-role attack plane. In doing so, the support plane not only protects itself, but also the helicopter. The support plane attacks fighter 1 and the multi-role attack plane, not the northern LSAR radar. The fuzzy RM directed the support plane to attack the more threatening emitters. This relates to a concept known as “lethality” used by the fuzzy RM to determine a queue of platforms to attack at each time step, and the fact that the support plane has only two EA beams and limited power.

Figure 2 shows that by time step seven the RM running on each member of the blue group has directed them to engage in simultaneous EA against all northern red threats. During time steps eight and nine (not pictured) the RM determines that simultaneous EA against all northern and southern threats is required. Cooperative EA against all threats is initiated in time step nine. Thus, the fuzzy RM has rendered the battlespace secure for the blue group by time step 11 (not pictured).

6 Summary

A fuzzy logic based resource manager (RM) for optimal allocation and scheduling of electronic attack resources distributed over many platforms is under development. Five components of the RM are discussed. Genetic algorithm based co-evolutionary data mining is introduced. Co-evolution refers to a process where both friend and foe agents and meta-agents simultaneously evolve in a complex simulated environment perceived by various sensors. Construction of the database, which is used for data mining and optimization was summarized. An algorithm for automatically creating fitness functions that contain past co-evolutionary history is given. Two methods of co-evolutionary optimization, computer versus computer (CVC) and human versus human (HVC) optimization were discussed. CVC optimization involves evolution with a computer-controlled opponent(s); HVC optimization, with a human-controlled opponent or both human and computer controlled opponents. Experimental results for each form of co-evolutionary optimization were discussed and a comparison of both methods was outlined. It was found in HVC optimization that the human player quickly reached the limits of his or her expertise resulting in the RM’s parameters reaching a constant value. In CVC optimization, the RM’s computerized opponent proved more resilient than a human player resulting in blue and red parameters, which change rapidly in time, unlike in HVC mode. The more resilient computerized opponent in CVC mode exposed the RM to more types of strategies with the potential for a more adaptive and robust RM.

Examples of the resource manager’s multi-platform response are given to illustrate the RM’s excellent performance and as a method of determining the value of the information data mined.

7 Acknowledgements

This work was sponsored by the Office of Naval Research. The authors would also like to acknowledge Mr. Edward Khoury, Mr. Robert Xander, Dr. Joseph Lawrence III, and Ms. Kristin Fisher.

References


