Fuzzy Logic Resource Manager: Multi-Agent Fuzzy Rules, Self-Organization and Validation

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

Abstract- *Modern naval battleforces generally include many different platforms each with onboard sensors such as, radar, ESM, etc. The sharing of information measured by local sensors across the battlegroup should allow for near optimal decisions. A fuzzy logic algorithm has been developed that automatically allocates electronic attack (EA) resources in real-time. The controller must be able to make decisions based on rules provided by experts. The fuzzy logic approach allows the direct incorporation of expertise. Genetic algorithm based optimization is conducted to determine the form of the membership functions for the fuzzy root concepts. Two algorithms for allocating power when faced with uncertainty are discussed. The self-organizing abilities of groups of agents controlled by the resource manager are considered. Methods of validating the resource manager are examined. These include evaluation through the scenario generator, a digital war game environment, the creation of mathematical measures of effectiveness, and other experimental test.*

Keywords: fuzzy logic, resource management, genetic algorithms, expert systems, distributed AI algorithm

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 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 [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 logic and discusses optimization with genetic algorithms. Section 3 discusses the RM's five major components. Section 4 discusses two fuzzy logic based power allocation algorithms that allocate power in the face of uncertainty. Section 5 discusses five approaches to validating the RM. Finally section 6 provides a summary.

2 Fuzzy logic, genetic algorithms and the 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 [5,6]. 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 [6].

Each root concept on the fuzzy decision tree has a membership function with parameters to be determined. The parameters of the root concept membership function are obtained by optimizing the RM over a database of scenarios using a genetic algorithm (GA) [7]. 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 is given in [6].

3 Subtrees of the RM

The resource manager is made up of four decision trees, the isolated platform decision tree (IPDT), the multi-platform decision tree (MPDT), the fuzzy parameter selection tree and the fuzzy strategy tree. The EA decision algorithm, which can be called by the IPDT or the MPDT, is an expert system for assigning electronic attack techniques. The IPDT provides a fuzzy decision tree that allows an individual platform to respond to a threat [6]. The MPDT allows a group of platforms connected by communication links to respond to a threat in a collaborative fashion [8]. The communications model used for simulation purposes is described elsewhere [8]. 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.

This section discusses the four major decision trees that make up the RM, the fuzzy EA decision algorithm and how they make efficient use of the Network-Centric paradigm. The Network-Centric paradigm refers to strategies that make optimal use of multiple allied platforms linked by communication, resources distributed over different platforms, and decentralized command.

3.1 The isolated platform decision tree

The IPDT allows a blue platform that is alone or isolated to determine the intent of a detected platform. It does this by processing data measured by the sensors, e.g., ESM, radar, IFF, etc. Even when an incoming platform's ID is very uncertain, the IPDT can still establish intent based on kinematics. When faced with multiple incoming platforms the IPDT can establish a queue of which platforms to attack first. Various subtrees of the IPDT have been discussed extensively in the past [6,8].

3.2 The multi-platform decision tree

The IPDT made limited use of the Network-Centric paradigm, using the other networked platforms for surveillance and electronic intelligence. However, it offers the advantage that it can make use of the Network-Centric paradigm by exploiting multiple platforms to gain geometric, physical and tactical advantage by employing multi-platform techniques that are more effective than standard techniques. Such techniques require coordination and communication from platform to platform, as well as some command and control structure.

3.2.1 Platform to platform interactions

The IPDT allowed an isolated platform to respond to an incoming emitter. The RM running on the isolated platform based its decisions and hence response on standard sensor output, e.g., range, range-rate, heading, heading-rate, etc. The isolated platform's response can range from simply continuing to monitor the environment, to deciding to engage in EA. If a decision to engage in EA is made by the RM, a call is made to the fuzzy EA decision algorithm, which is discussed below.

As it stands, the IPDT can not take full advantage of the Network-Centric paradigm. To do this another decision tree, the MPDT is required. Using sensor output, the MPDT allows a group of platforms, connected by a communications network to work together in an optimal fashion to take advantage of the full potential of the Network-Centric paradigm. It allows self-organization. By self-organization it is meant that the MPDT allows the blue agents to automatically organize without any blue platform acting as a fixed central commander.

The MPDT requires many new rules, some analogous to rules found on the IPDT, but most quite distinct. The following examines, at a coarse level some of these rules and their related fuzzy concepts.

3.2.2 Some root and composite concepts on the MPDT

Figure 1 displays a significant subtree of the MPDT. A vertex with a line is read as "AND", a vertex without a line represents "OR" and a circle on an edge is read as "NOT". The first rule on the subtree to be considered is the fuzzy concept of a platform's need. If the RM aboard a blue platform determines a threat is "attacking" by using the IPDT, then the detector should alert other platform's to its "need" for assistance.

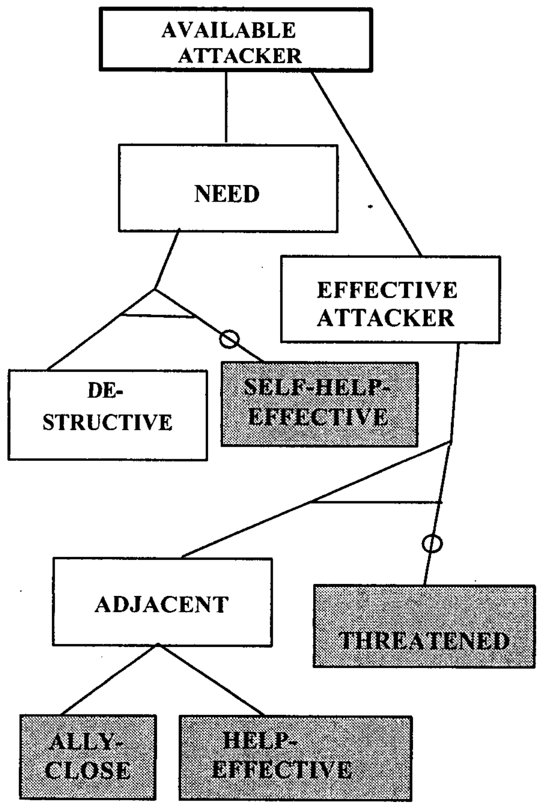

Figure 1 : Subtree of the multiplatform decision tree

A platform's "need" is a function of its ability to respond to a threat, and how destructive the threat is perceived to be. The composite concept "need" is constructed using the membership functions for the root concepts "self-help-effective" and "destructive". The membership function for "self-help-effective" is a function of the EA resources aboard the i^{th} platform, where "need" is being determined.

The composite concept, "destructive" is constructed from the root concepts "potentially-destructive" and "kinetic-energy destructive" (not pictured). The fuzzy membership function for "potentially-destructive" is actually an index between zero and one, assigned by experts detailing how threatening the emitter is perceived to be in terms of its onboard hardware. The fuzzy membership function for "kinetic-energy-destructive" is a function of the emitter's estimated translational and rotational kinetic energy.

The composite concept of "need" reduces the amount of data that has to be sent over the network. It does this by sending processed information over the network, as opposed to raw data.

The composite concept "adjacent" checks platform/threat disposition, along with resources onboard the potential "helper" platform. A helper platform is one that is not threatened, but has received a communication message, that another platform is threatened, i.e, the threatened platform is communicating to the helper that it has "need." The fuzzy root concept "ally-close" relates to how close, the threatened ally is to the platform that is evaluating its ability to help in terms of the concept "adjacent." The root concept "help-effective" relates to how effective the helping platform might be if it should come to the assistance of the threatened platform that has "need."

The composite concepts "effective attacker" and "need" are combined through an "and" connective to construct the composite concept "available attacker". If the membership function for "available attacker" exceeds a certain threshold the helping platform comes to the assistance of the platform with need. Note that the parts of the tree leading up to "need" are calculated on the threatened platform. The subtree for "effective attacker" and the final "and" operation between "need" and "effective attacker" are calculated on the helping platform. This allows the RM to take advantage of multiple computers within the blue platform group. This allows automatic self-organization.

Figure 2 depicts a battle between six blue agents and five red agents. The RM running on each blue agent automatically organizes the blue effort so that without the action of any fixed central commanding agent, the blue support plane defended itself and blue destroyer one. It is important to note that the RM on the blue plane directed it to immediately attack the two northern red fighters as opposed to attacking the northern red radar. The red fighters are considered more threatening to the blue group than the red radar. This was a difficult resource allocation problem since the blue plane was expected to respond immediately and it only has two EA beams. The RM's ability to organize the red agents into a queue of those who should be attacked first is essential to blue's survival. Eventually as information becomes available through communication links to the other blue platforms, their own copies of the RM determine they should engage in battle against the northern red threats. As more information comes in the blue agents engage in combined electronic attack against all northern and southern red threats rendering the battlespace secure for blue (not pictured).

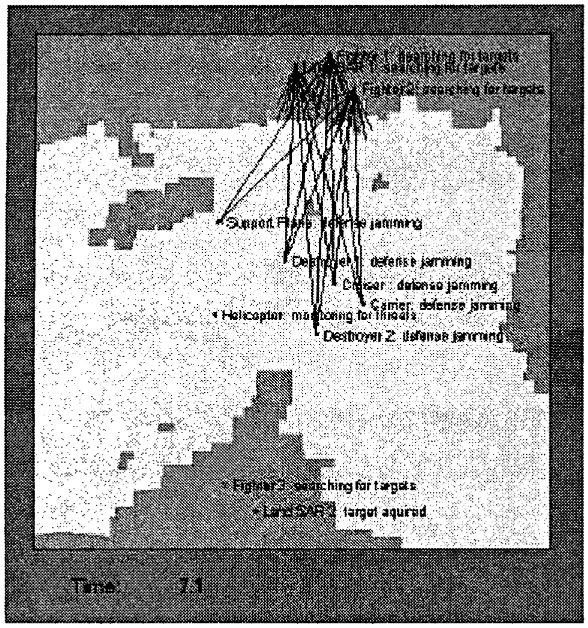

Figure 2 : SG output showing self-organization

3.3 The fuzzy parameter selection tree

The fuzzy parameter selection tree can be called by the IPDT, MPDT, the fuzzy strategy tree, and the fuzzy EA algorithm. It allows each tree to select the best parameters determined off-line using genetic optimization. The selections are a function of emitter ID, uncertainty in ID, intelligence reports, battlespace geometry, geography, weather, etc. These parameters can include probabilities for the best strategy calculated using game theory.

By selecting specialized parameters sets for different situations the RM can use the same decision trees and functional forms for the fuzzy membership functions. This also allows the RM to be employed on many different types of blue platforms and deal with very general red threats.

3.4 The fuzzy strategy tree

A strategy tree is an agent's concept of an opposing agent's decision tree. If an agent has sufficient knowledge of the enemy's past behavior the strategy tree can be very useful for predicting future behavior [8].

3.5 The fuzzy EA decision algorithm

Once the IPDT or the MPDT determines an action is required, the fuzzy EA decision algorithm becomes active. This fuzzy algorithm allows the RM to pick the best EA technique(s) to use against the incoming emitters as a function of sensor data.

The fuzzy EA decision algorithm is an expert system based partially on military doctrine obtained by interviewing experts, preferred techniques found in the literature, and entirely new classes of techniques invented that exploit the Network-Centric paradigm.

4 Minimum EA power allocation in the face of uncertainty

This section will consider the EA power allocation problem for n blue platforms versus one red platform. For the analysis considered below there will be one blue platform in the main beam of a red victim radar and $n-1$ blue platforms in the sidelobes of the red radar. The single blue platform in the main lobe of the red radar will be referred to as a self-screening agent. Quantities related to a self-screening agent will carry "ss" subscripts. The other $n-1$ blue platforms will be referred to as support agents, with related mathematical quantities carrying a "su" subscript. The more general problem, EA power allocation for n blue platforms versus m red platforms will be considered in a future publication. The algorithm below has been simplified for space consideration.

4.1 Two approaches to power allocation

Intrinsic to EA technique allocation is determining the amount of power each blue platform should direct at a red platform. It is desirable to use as little power as is possible while taking into account the uncertainties underlying the conflict. If too much power is used against one red platform there might not be sufficient power to use against another red platform. Excessive use of power by blue may allow an enemy to home in on blue's position possibly without use of other sensors. Also, excessive power usage during application of EA techniques may interfere with other systems. Finally, if power is used in an intelligent fashion red may misinterpret blue's EA activity as a natural phenomena or a system failure giving blue an additional advantage.

It can be shown easily using the techniques of [9] that the following system of inequalities (1) and (2) govern the power allocation problem

$$\vec{\Pi} \cdot \vec{Y} \ge 1 \qquad (1)$$

$$\Pi_{min,j} \le \Pi_j \le \Pi_{max,j} \qquad (2)$$

where $\Pi_{\min j}$ and $\Pi_{\max, j}$ are the minimum and maximum values of the j^{th} component for for $j = 0$ to $n-1$. Let the power vector be defined as

$$\vec{\Pi} \equiv (\rho_{1,ss}, \rho_{1,su}, \ldots, \rho_{n-1,su}) \tag{3}$$

with $\rho_{1,ss}$, the power output for the self-screening blue agent and $\rho_{i,su}$, the power output for the i^{th} support blue agent for $i=1$ to $n-1$. The components of the vector $\vec{Y}$ in (1) and defined in (4-7) consist of many factors representing the parameters governing the operation of the sensor and EA systems on the blue platforms and the red radar. The parameters characterizing the red radar are confined to the factor c_R, defined in (7).

$$Y_o \equiv (\frac{\lambda}{4\pi})^2 \left(\frac{G_j}{L_j B_j} \right)_{ss} \frac{G_r}{L_a R_{sso}^2 c_R} \tag{4}$$

$$Y_i \equiv (\frac{\lambda}{4\pi})^2 \left(\frac{G_j}{L_j B_j} \right)_{su,i} \frac{G_{r,ji}}{L_{a,i} R_i^2 c_R} \tag{5}$$

$$\vec{Y} \equiv (Y_o, Y_1, \ldots, Y_{n-1}) \tag{6}$$

$$c_R \equiv \frac{P_r \tau_{pw} f_r \tau G_T^2 \sigma \lambda^2}{(4\pi)^3 R_{sso}^4 L_l D_q(n, P_{fa}, P_D)} - kT_s \tag{7}$$

The subscripts "ss" and "su,i" in (4,5) refer to the self-screening agent and the i^{th} support agent. The appearance of these subscripts on a bracket is a short-hand implying that each of the quantities in the bracket are labeled with the relevant subscript. The parameters appearing in equations (4-7) are L_j, the loss, excluding atmospheric, for the respective agent; B_j, the bandwidth of the respective agent; G_r, the radar antenna gain in the direction of the ss-agent; L_a, the atmospheric loss for the ss-agent; R_{sso}, the range of the ss-agent; G_j, the agent antenna gain in the direction of the radar; $G_{r,ji}$, the radar antenna gain in the direction of the i^{th} support agent; La,i, the atmospheric loss for the i^{th} support agent; R_i, the range of the i^{th} support agent; τ_{pw}, the pulse width of the victim radar; f_r, PRF of the victim radar; τ, the time on target of the victim radar; L_l, the loss factor of the victim radar; σ, the cross section of self-screening agent; P_r, the peak power of the victim radar; G_T, the gain of the radar transmitter; P_D, the minimum probability of detection required by the victim radar to detect blue; P_{fa}, the probability of false alarm for the victim radar; n, the number of pulses integrated by the victim radar; q, the number of the Swerling model [9] appropriate to how victim the radar "sees" blue; $D_q(n, P_{fa}, P_D)$, the detectability factor [9]; T_s, the thermal noise temperature of the radar; λ, the red radar's wavelength, and k, Boltzman's constant. All ranges are measured with respect to the red victim radar.

To determine the appropriate power level each beam of each blue platform must use, the blue meta-agent power output must be related to the minimum estimated power received by the red agent that is required for a particular type of detection. Blue may want red to conclude that he is only seeing noise born of some natural process. The power delivered to red should probably only be larger than a certain minimum detection threshold. If blue desires that red detect ultimately what is a false target then blue's power output should be much larger.

Inequalities (1) and (2) specify the constraints on power. The set of power vectors that satisfy each inequality define a closed half spaces in R^n, the space of n-tuples of real numbers. The intersection of a finite number of closed half-spaces in R^n defines a polyhedral convex set. To extract a unique solution a linear function in power

$$f(\vec{\Pi}) = \vec{\mu} \cdot \vec{\Pi} \tag{8}$$

is defined on and minimized over the polyhedral convex set specified by (1-2)

The elements of the power coefficient vector

$$\vec{\mu} = (\mu_o, \ldots, \mu_{n-1}) \tag{9}$$

in (9) are fuzzy grades of membership reflecting the importance of conserving the beam's power. Minimization of a linear function over a polyhedral convex set is a standard linear programming problem. Since the polyhedral convex set is bound, the Fundamental Extreme Point Theorem [10] guarantees that the minimum of the linear function (8) occurs at a corner point. A simplex algorithm is used to rapidly search the corner points for the correct solution.

The physical interpretation of the linear function in (8) is that it is the total weighted power sum over the individual blue beams with the weights reflecting the importance of conserving the beam's power. It is of course useful to minimize power as discussed above. Although physical, the use of a linear function is arbitrary, it is used to obtain an unique solution. There is nothing to exclude the selection of a nonlinear function as shown below.

A natural nonlinear function in power to minimize is

$$f(\vec{\Pi}) \equiv \sqrt{\sum_{j=0}^{n-1} (\Pi_j / \Pi_{\max, j})^2} \tag{10}$$

The function in (10) can be minimized over the polyhedral convex set defined by (1,2) through a straightforward application of the Cauchy-Bunykovskii-Swartz Theorem [11] yielding the following power vector components as a solution to the nonlinear minimization problem

$$\Pi_j = Y_j \cdot \Pi^2_{max,j} \Big/ \sum_{k=0}^{n-1} \left(Y_k \cdot \Pi_{max,k}\right)^2 \qquad (11)$$

where it is assumed that $\Pi_{min,j}=0$ for $j= 0$ to $n-1$.

Note the solution given by (11) is closed form and exact up to model assumptions. Furthermore, it is computationally much faster than the linear programming approach and uses very little memory. The linear programming approach has the advantage it is much easier to generalize for more complicated scenarios. It should be noted that both power allocation algorithms allow the determination of power levels each blue agent uses without a fixed central commander, i.e., they contribute to the self-organizing aspect of the RM.

4.2 Simulation results

The following subsection provides a simulation that allows a comparison between the linear programming and nonlinear programming approaches to power allocation. It also gives a basic discussion of the uncertainties associated with the various parameters and how these uncertainties can be effectively dealt with through the use of the theory of fuzzy numbers and fuzzy programming techniques.

The simulation model is formulated as follows. As described in section 4.1 there are assumed to be n blue platforms engaging one red platform. The self-screening blue platform is in the main lobe of the red radar, and the $n-1$ blue support platforms are in the red radar's sidelobes. The parameters specified on the right of equations (4,5,7) are considered to be Gaussian random variables. Each blue platform estimates the signal to noise ratio required by the red radar for a successful detection of a particular phenomena based on using a detectability factory [9]. This assumes that blue has an estimate of not only his own parameters and uncertainties, but also those of red. Each blue agent uses the best estimates of these parameters for the relevant calculations. Within the simulation, these parameters are actually values of Gaussian random variables. The actual power received by red is then evaluated using the true parameters and a probability of detection is calculated using a Swerling model. If red's probability of detection is greater than or equal to 50%, then red is considered to have made a successful detection. So it is to blue's advantage to keep red's probability of detection below 50% while keeping power usage below a certain threshold for the reasons outline above. The simulation employed an ensemble average to get statistics.

Due to the significant uncertainties that blue can encounter in the parameters listed below (7), the values of the parameters must be selected carefully. This is typically done by having an expert select the parameters. Even with this approach it is easy to use excess power which can contribute to interference or allow red to have more than 50% probability of detection. It has proved useful to develop a formal way of quantifying uncertainty. This was done through the theory of L-R symmetric fuzzy numbers [12]. Fuzzy number theory was used for both the linear and the nonlinear programming approaches. This allowed Negoita's approach to fuzzy linear programming [12] to be used.

Figure 3 depicts the results of the simulation. The vertical axis gives the probability of detection at the red radar. Red's radar power and gain were multiplied by a Gaussian random variable with mean unity and width equal to the quantity measured on the horizontal axis. The solid curve represents the results of the fuzzy linear programming approach. The dashed curve represents the results for the fuzzy nonlinear programming approach. Both power allocation algorithms are effective in keeping red's probability of detection below the desired 50% probability. Both power allocation techniques over estimate the amount of power required when the width of the Gaussian random variables is small. The fuzzy linear programming approach effectively allocates less power when dealing with uncertainty than the nonlinear approach. It is also more computationally intensive than the nonlinear approach, a consideration that can be significant in some real-time situations.

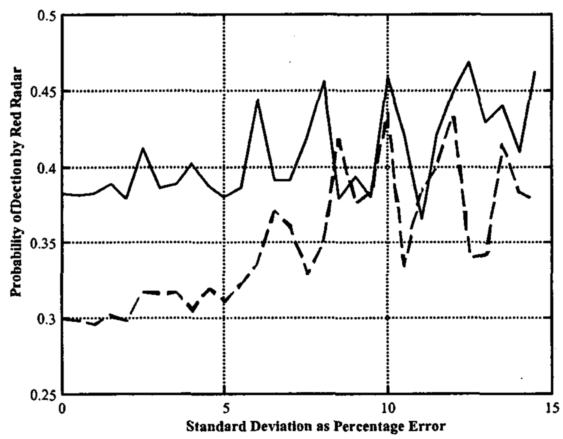

Figure 3: True probability that red will detect blue vs. the standard deviation of Gaussian noise. Solid curve is fuzzy linear programming; and the dashed curve, fuzzy nonlinear programming.

5 Validation of the RM

There have been to date four different approaches to the validation of the RM conducted. These approaches are the evaluation of the RM within a digital war game environment [13]; the development of measures of effectiveness (MOE), one of which is described below; testing the RM using a hardware simulator, and using evolutionary algorithms to evolve parameters and fuzzy decision tree structure from a random starting point using constraints based on expertise as opposed to explicit rules as is more conventional in expert system theory.

The scenario generator (SG) and the related activity of co-evolutionary data mining is described in detail elsewhere [13]. So only a quick summary will be given here. The SG allows the creation of a very general battlespace that may have a battle map with desert, forest, jungle, urban areas, and water. Very general blue agents, i.e., the defending platforms each one of which runs its own copy of the RM can be placed within the battlespace. The agents can be ships, planes, helicopters, soldiers, decoys, etc. The SG allows the agents to be equipped with very general sensors, weapons, etc. Likewise, the SG allows very general red agents to be created and well equipped. The SG has two modes of operation, computer vs. computer (CVC) mode, and human vs. computer mode (HVC). In both modes each blue agent has its own copy of the RM. The blue agent's RM exercises all control functions over that agent and only that agent. In CVC mode each red agent is controlled by its own computerized logic different from the RM. In HVC a human expert controls one red agent per time step through a GUI with a PPI radar display, the controls for firing red missiles and motion controls. The human player can select a different red agent each time step to control: those red agents not under human control run under computer logic as in CVC mode. Many different battles can be fought using the SG, the results are stored in a data base and also a computer movie. Human experts have evaluated many of the computer movies and agreed on the RM's excellent decisions.

Figure 2 is a frame from such a movie depicting six blue agents engaging in conflict with five red agents. The battlespace consists of ocean and desert depicted as light and dark areas, respectively.

Another approach to validation that has been pursued is to write down, based on expertise, measures of effectiveness (MOE). These are typically fuzzy decision trees. The MOE's allows quantification of the value of the RM's responses. The RM has been very successful when evaluated in this way. Figure 4 provides a subtree of a MOE tree that has been used to determine the effectiveness of EA technique allocation on red's PPI display. This subtree helps to evaluate the RM's effectiveness when using false targets. The full MOE tree will be described in greater detail in a future publication.

A third validation effort involves the use of a hardware simulator referred to as the search radar electronic warfare simulator (SRES). This type of evaluation is similar to the work done with the SG, but in this case the digitally simulated radars, electronic warfare equipment, communication systems, and sensor displays of the SG are replaced with real hardware systems. In this application the RM is used as a controller on SRES, allowing evaluation of the RM and EW techniques. As in the previous two approaches to validation the RM has also passed its test at this level.

Figure 5 is taken from one of the PPI displays of SRES. It shows a row of false targets and a noise technique that copies of the RM running on two blue agents used to confuse red. Output of this type is used to judge the effectiveness of the RM.

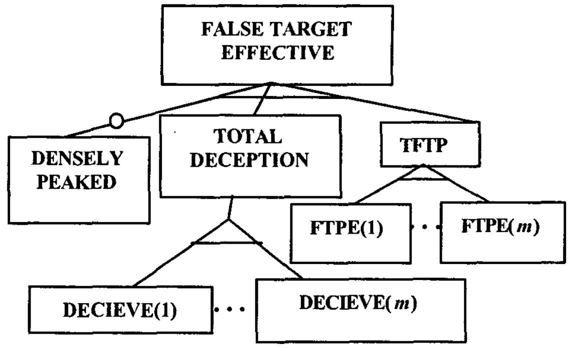

Figure 4: A significant subtree of the MOE tree

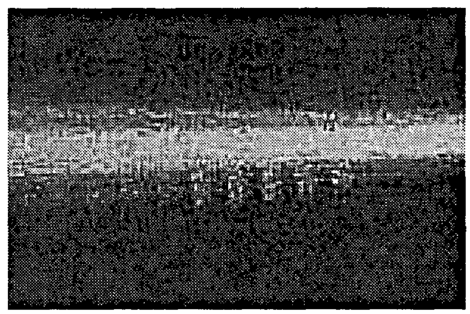

Figure 5: False target and noise output of the SRES hardware simulator that uses the fuzzy resource manager as a controller.

The fourth contribution to the validation effort consists of using a genetic program (GP) to mine military data bases for fuzzy decision tree structure and along with them fuzzy rules [14,15]. The original approach to rule determination was to consult domain experts or existing doctrine. In the GP based approach using a minimum of constraints imposed by domain experts and random starting points, fuzzy decision tree structures and also fuzzy rules are evolved. In many cases the same rules and fuzzy decision trees written down based on experts rules or pre-existing doctrine are re-obtained. The GP's rediscovery of rules and fuzzy decision trees already in uses provides support and hence a kind of validation. Of course, the GP's are also being used to data mine new rules and fuzzy decision tree structures [15].

Finally, even though the RM has been very successful when subjected to the test described above, field test are planned. Experiments in the field, i.e., on the ocean and other demanding terrain will expose the RM to conditions difficult to simulate digitally or in hardware.

6 Summary

A fuzzy logic based algorithm for optimal allocation and scheduling of electronic attack resources distributed over many platforms is under development. Optimization of the resource manager is conducted by using a genetic algorithm as a component of a data mining process. Construction of the database, which is used for optimization was summarized. The four decision trees making up the resource manager are discussed. The RM's ability to self-organize platforms, i.e., for the platforms to carry-out their functions without a fixed central commanding platform is examined. The fuzzy EA algorithm is discussed including two new EA power allocation algorithms. One of these algorithms uses a version of fuzzy linear programming; the other, fuzzy nonlinear programming. Both make use of the theory of L-R symmetric fuzzy numbers to deal with uncertain parameters. Simulated results are provided showing the excellent performance of these two algorithms. Five approaches to validation of the RM are considered including mathematical measures of effectiveness, evaluation of the RM in a digital war game environment, testing the RM by using it as a controller for a hardware simulator, using a genetic program to evolve fuzzy decision tree structure and fuzzy rules from a random starting point for subsequent comparison to decision tree structure and rules obtained by interviewing experts, and field test in challenging environments using real hardware assets.

7 Acknowledgements

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

8 References

[1] D. C. Schleher, *Electronic Warfare in the Information Age*, Chapter 1, Artech House, Boston, 1999.

[2] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Chapter 11, Artech House, Boston, 1999.

[3] L.H. Tsoukalas and R.E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, Chapter 5, John Wiley and Sons, New York, 1997.

[4] J. H. Holland, *Hidden Order How Adaptation Builds Complexity*, pp. 1-15, Perseus Books, Reading, 1995.

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

[6] J.F. Smith, III and R. Rhyne, II, "A Resource Manager for Distributed Resources: Fuzzy Decision Trees and Genetic Optimization," *Proceeding of the International Conference on Artificial Intelligence, IC-AI'99*, H. Arabnia, Vol. II, pp. 669-675, CSREA Press, Las Vegas, 1999.

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

[8] J.F. Smith, III and R.D. Rhyne, II, "Genetic Algorithm Based Optimization of a Fuzzy Logic Resource Manager: Data Mining and Co-evolution," *Proceeding of the International Conference on Artificial Intelligence, IC-AI'2000*, H. Arabnia, Vol. I, pp 421-428, CSREA Press, Las Vegas, 2000.

[9] D. C. Schleher, *Introduction to Electronic Warfare*, Chapter 6, Artech House, Boston, 1986.

[10] A. M. Glicksman, *An Introduction to Linear Programming and the Theory of Games*, p. 52, Dover Publications, Inc., Mineola, NY, 1963.

[11] R. G. Bartle, *The Elements of Real Analysis*, p. 56, John Wiley and Sons, New York, 1976.

[12] T. Terano, K. Asai, M. Sugeno, *Fuzzy System Theory and Its Applications*, Chapter 7, Academic Press, Inc., Boston, 1992.

[13] J.F. Smith, III and R.D. Rhyne, II, "Co-evolutionary Data Mining to Discover Rules for Fuzzy Resource Management", *Proceeding of the International Conference on Artificial Intelligence, IC-AI'2000*, H. Arabnia, Vol. I, pp 421-428, CSREA Press, Las Vegas, 2001.

[14] J.R., Koza, , F.H. Bennett III, D. Andre, and M.A. Keane. *Genetic Programming III: Darwinian Invention and Problem Solving*. Chapter 2, Morgan Kaufmann Publishers, San Francisco, 1999.

[15] James F. Smith, III, "Data Mining for Multi-agent Fuzzy Decision Tree Structure and Rules", *Fusion2002: Proceedings of the 5th International Conference on Information Fusion*, to be published 2002.