Evolving Fuzzy Decision Tree Structure that Adapts in Real-Time

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

A fuzzy logic algorithm has been developed that automatically allocates electronic attack (EA) resources distributed over different platforms 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. The resource manager is made up of five parts, the isolated platform model, the multi-platform model, the fuzzy EA model, the fuzzy parameter selection tree and the fuzzy strategy tree. Automatic determination of fuzzy decision tree topology using a genetic program, an algorithm that creates other algorithms is discussed. A comparison to subtrees obtained using a genetic program and those constructed by hand from rules is made. Experiments designed to test various concepts in the expert system are discussed, including its ability to allow multiple platforms to self-organize.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Control theory, Plan execution, formation, and generation, Scheduling
I.2.11[**Artificial Intelligence**]: Intelligent agents Multiagent systems

General Terms

Algorithms, Performance, Theory.

Keywords

genetic program, genetic algorithm, fuzzy logic, self-morphing, resource management, 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), a device for detecting radar emissions; 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 data bases 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 game records the result of the RM and expert’s battle, automatically assembling a data base 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 data base.

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 two fuzzy concepts and their membership functions. Section 3 discusses the RM’s five major components. Section 4 introduces and discusses genetic programs and examines a significant fuzzy decision tree evolved by the genetic program. Section 5 introduces the concept of self-morphing fuzzy decision trees. Section 6 provides a brief discussion of the multi-platform decision tree (MPDT). Section 7 describes three significant experiments that illustrate the RM’s ability to allow multiple agents to self-organize

GECCO'05, June 25-29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

without the benefit of a central commander, tolerate corrupted input data and to determine platform intent in the face of multiple distinct enemy strategies. Finally, section 8 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 some concepts related to fuzzy decision trees and their related fuzzy concepts. The parameterization of root and composite concepts are discussed.

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:

$$A = \{(x, \mu_A(x)) | x \in X\} \qquad (1)$$

$\mu_A(x)$ 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*.

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." 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 [6,7].

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 the literature [6]. The RM has many root and composite concepts associated with it. Two such concepts are discussed below. They are "close" and "ranging."

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 (nmi) and range rate, the first time derivative of the range denoted as, $\dot{R} \equiv dR/dt$ (ft/sec). The membership function takes the form

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

The parameters to be determined by data mining are

$$\alpha, \quad R_{\min}, \quad \text{and} \quad \dot{R}_{\min} . \qquad (3)$$

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, $\ddot{R} \equiv d^2R/dt^2$, of the range as given below

$$\mu_{ranging} = \frac{1}{1 + \delta \Big/ \max\left(\left|\ddot{R}_i\right|, a_{\min}\right)} . \qquad (4)$$

The two parameters to determine through data mining for ranging are

$$\delta \quad \text{and} \quad a_{\min} . \qquad (5)$$

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 in [8]. The fuzzy parameter selection tree is designed to make optimal or near optimal selections of root concept parameters from the parameter data base assembled during previous optimization with the genetic algorithm. Finally, a strategy tree is a fuzzy tree that an agent uses to try to predict the behavior of an enemy.

4. DISCOVERING THE IPTD FUZZY DECISION TREE'S STRUCTURE USING A GENETIC PROGRAM

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, 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 full IPDT have been discussed extensively in the past [6,8].

In a previous paper a genetic algorithm (GA) was used as a data mining function to determine parameters for fuzzy membership functions [6]. This section introduces a different data mining function, a genetic program [9] (GP). The GP data mines fuzzy decision tree structure, i.e., how vertices and edges are connected and labeled in a fuzzy decision tree. Whereas, the GA based data mining procedures determine the parameters of and hence the form of fuzzy membership functions, the GP based procedure actually data mines fuzzy if-then rules.

A genetic program is a problem independent method for automatically creating computer programs. Like a genetic algorithm it evolves a solution using Darwin's principle of survival of the fittest. Unlike the genetic algorithm, of which it can be considered an extension: its initial, intermediate, and final populations are computer programs.

Like a genetic algorithm the fittest individuals in the population are copied and subject to two operations, crossover and mutation. Crossover corresponds to sexual recombination, a kind of mating between parent computer programs. The crossover operation is constrained to produce structurally valid offspring. Finally, the mutation operation is a random change in a computer program that is part of the evolving population.

4.1 Building the IPDT's subtree

This subsection discusses the automatic generation of a subtree of the RM's isolated platform decision tree (IPDT), as depicted in Figure 1, using a GP to data mine a data base of military scenarios. The IPDT of the fuzzy RM has been extensively discussed [6,8,10]. The IPDT uses sensor data to make decisions about the threat status of an incoming platform. A platform may be an airplane, helicopter, ship, etc. Previous versions of the IPDT were constructed by hand based on human expertise.

As in any data mining operation the first step is the construction of the data base that will be mined. The data base used for automatic construction of subtrees of the IPDT consists of sensor output for the various platforms involved in the engagement. Each record contains the range, bearing, elevation, ID information for the emitting platforms involved, etc. It also contains a number between zero and one, which represents an expert's opinion as to whether or not the emitter is attacking.

To use the genetic program it is necessary to construct terminal and function sets relevant to the problem. The terminal sets used for construction of subtrees of the IPDT typically consist of one or more fuzzy root concepts [6,8,10].

A typical terminal set is

T={close, heading_in, elevation, ranging, banking, …, (6)
elevating, interaction, friend, lethal, uncertain, marginal-ID}.

The root concepts "close" and "ranging" are discussed in section 2 and in the literature [6,10]. The fuzzy membership function for "friend" gives the degree of membership of the detected platform in the concept "friend," i.e., how much confidence does blue have that the emitter is a friend. Finally, the membership function for the root concept "lethal" is found by summing the membership functions for all the foe classes. All the concepts in (6) are explained in greater detail in the literature [10].

The function set, F, consist of the logical operations of "AND" and "OR" as well as the logical modifier "NOT," i.e.,

$$F=\{AND, OR, NOT\}. \quad (7)$$

The fitness function for data mining the IPDT's subtree is

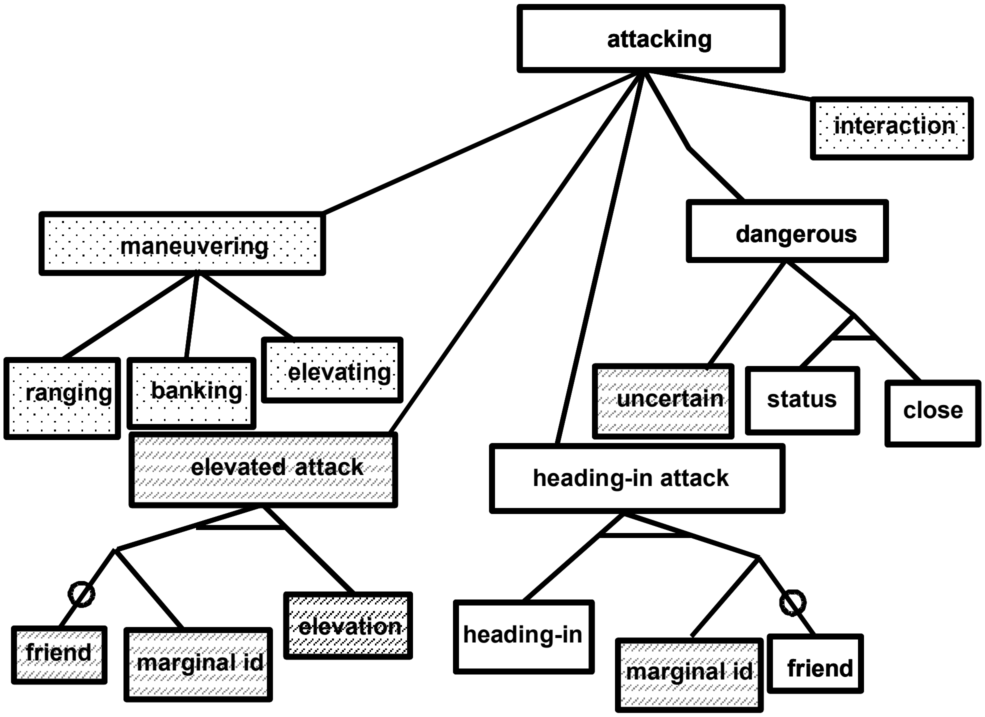


Figure 1: A significant subtree of the IPDT that was evolved by the genetic program. The highest priority concepts have boxes with a white background; intermediate priority concepts, boxes with lines in background; and lowest priority concepts, boxes with dotted backgrounds.

$$fitness(i) \equiv \frac{1}{n_{time} \cdot n_{db}} g(i, n_{db}, n_{time}, \tau) - \alpha \cdot l(i) \qquad (8)$$

with

$$g(i, n_{db}, n_{time}, \tau) \equiv \qquad (9)$$
$$\sum_{j=1}^{n_{db}} \sum_{k=1}^{n_{time}} \chi(\tau - | \mu_{gp}(i, t_k, e_j) - \mu_{\text{expert}}(t_k, e_j) |).$$

where e_j is the j^{th} element of the data base; t_k is the k^{th} time step; n_{db} is the number of elements in the data base; n_{time} is the number of time steps; τ is the tolerance; $\mu_{gp}(i,t_k,e_j)$ is the output of the fuzzy decision tree created by the GP for the i^{th} element of the population for time step t_k and data base element e_j; $\mu_{expert}(t_k,e_j)$ is an expert's estimate as to what the fuzzy decision tree should yield as output for time step t_k and data base element e_j; α is the parsimony coefficient; $l(i)$ is the length of the i^{th} element of the population, i.e., the number of nodes in the fuzzy decision tree corresponding to the i^{th} element; $\chi(t)$ is the Heaviside step function which is unity for $t \geq 0$ and zero otherwise.

Observe, that the form of (8,9) reflects that the expert's estimate, $\mu_{expert}(t_k,e_j)$ is uncertain, and need only be reproduced within a tolerance, τ. Also, to increase the robustness of the GP created tree, the fitness of the fuzzy decision tree used by the GP is found by averaging over time and the data base.

The parsimony pressure, $\alpha \cdot l(i)$, appearing on the right-hand-side of (8) provides a penalty that reduces the i^{th} population element's fitness if it is longer than needed. Thus given two trees that are both effective, the smaller tree will have the higher fitness. This provides a computational implementation of Occam's razor [11].

The genetic program terminates after one of the following occurs: the number of generations reaches a preset maximum, the fitness has not changed by a preset amount in a certain number of generations or the fitness is within an acceptable tolerance of the maximum value.

4.2 Genetic program generated IPDT subtrees

Figure 1 depicts the IPDT's subtree evolved by using GP based data mining. This subtree was originally drawn based on experts' intuition [10]. A line on a vertex denotes the logical connective "AND," a vertex without a line indicates the logical connective "OR," and a circle on an edge denotes the logical modifier "NOT." The concepts "close" and "ranging" are discussed in section 2. The other concepts have been developed in the literature [6,10].

This subtree of the IPDT has been rediscovered by data mining a data base of military scenarios using a GP. Other more sophisticated trees have been discovered by GP based data mining, but this simple tree is considered here to illustrate the process. The GP in many different runs was successful in constructing this subtree as expected, however, it did not always construct the same tree. The concept on the right-hand-side of the tree labeled "status" is a placeholder. In some trees constructed by the GP, its value was "NOT a friend" in others "status" took the value "lethal." Also, using different random seeds for each run, the number of generations required for the GP to stop varied. The GP's ability to construct the same subtree as that written down based on experts' rules provides a form of support for the subtree since it can be found in multiple ways. Finally, the GP's ability to construct other trees points up the potential non-uniqueness of the subtree.

The original subtree obtained from human expertise used the concept "lethal" in the position of "status" in Figure 1. This placed a heavy reliance on properly identifying an incoming platform as an enemy. The flaw in this is that if red is sufficiently deceptive, he may be able to convince the ID system that he is a neutral. This might allow him to get near enough to blue to execute a red action and subsequently win.

As observed above, in some experiments the GP recommended "lethal" be assigned to the node labeled "status" in Figure 1, while in other experiments the recommendation was to assign "NOT a friend" to this node. Using "NOT a friend" has the advantage that the RM can attack a platform designated by the ID system as a "neutral," if the platform exhibits hostile behavior and triggers "close." Finally, the disadvantage of assigning "NOT a friend" to the "status" node is that real neutral agents must exhibit cautious behavior near blue.

The GP's ability to find different fuzzy decision trees for the same problem most likely relates to the military data base that is being data mined, the fitness function and the parameters characterizing convergence. These are subjects of current research.

5. SELF-MORPHING PROPERTY

In Figure 1, the concepts are assigned one of three priority levels. A box with a pure white background represents a highest priority concept. Intermediate priority concepts have lines in the background of their box. Lowest priority concepts have boxes with a dotted background. When there is a significant amount of time available for evaluation all concepts are evaluated. When there is less time the lowest priority concepts are not evaluated, i.e., only the highest and middle priority concepts are considered. During the most time-critical periods only the highest priority concepts are employed. Finally, the number of priority levels need not be fixed at three, there can be more or less.

In fuzzy logic there is more than one mathematical form for the logical connectives "AND" and "OR." Common and computationally simple forms of "AND" and "OR" are the "min" and "max" forms, respectively [6]. Another form for "AND" that sometimes delivers better results is that of Zimmerman and Zysno [5]. This form is significantly more computationally intensive.

Just as there are priority levels for concepts, mathematical forms for logical connectives are also prioritized. When time is not critical more computationally intensive forms are used; during periods in which time is extremely critical the computationally simplest forms of "AND" and "OR" are used.

Based on input indicating the time-criticality of situations the RM can opt to evaluate or not evaluate certain concepts and also use simpler or more complicated mathematical forms for logical connectives. Deleting certain concepts effectively changes the shape of the fuzzy decision tree, i.e., the tree "morph's." Since the RM can elect to do this itself based on input, this ability is referred to as the *self-morphing property*.

Even though fuzzy logic is fast, compact and reliable if many agents within a decision support algorithm have their own fuzzy decision trees, then eventually the decision support algorithm becomes computationally intensive if each tree has its most complicated form. The self-morphing property allows the fuzzy decision trees to return high quality decisions even when operating in a complicated environment.

6. OTHER DECISION TREES IN THE RM

As it stands, the IPDT can not take full advantage of the resources distributed over different platforms. To do this another decision tree, the MPDT is required [10]. 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 multiple platforms with extensive resources. Finally, the MPDT requires many new rules, some analogous to rules found on the IPDT, but most quite distinct.

6.1 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 resources distributed over various platforms [10]. To do this another decision tree, the MPDT is required [6,8,10]. Using sensor output, the MPDT allows a group of platforms, connected by a communications network to work together in an optimal fashion.

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.

6.2 Some root and composite concepts on the MPDT

The first rule to be defined 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.

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", abbreviated as “PD” in Figure 2; and "KINETIC-ENERGY DESTRUCTIVE", abbreviated as “ED” in the same figure. 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 version of the MPDT differs from those found in the literature in two ways [6, 8, 10]. The first difference is that the concept “IS-N” which is an abbreviation for “IS-NEUTRALIZED” has been added to the tree. This is a concept that helps the RM make a decision as to whether a detected enemy platform has been neutralized by EA or otherwise. It does this using sensor and communications data. The final difference is the dashed line below “AVAILABLE ATTACKER.” This dashed line represents a crisp “AND” as opposed to the fuzzy logical connective “AND”. The crisp connective has been determined to be more effective in this case.

6.3 The fuzzy parameter selection tree

The fuzzy parameter selection tree allows the RM to select in real-time 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.

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.

Each tree in the RM has associated with it a fuzzy parameter selection tree. The one associated with the IPDT is the isolated parameter selector tree (IPST). A subtree of the IPST is depicted in Figure 3. Root concepts are depicted as gray boxes; composite concepts, as boxes with a white background. The root concept labels indicate that the IPST's decisions are influenced by the number of hostile platforms detected, those platforms that are closing in, precipitation, adverse terrain that could hide an enemy, and missiles that have been launched. There are many other concepts that are not displayed that could result in a parameter change, they will be the subject of a future publication.

6.4 The effect of the IPST on the RM's behavior

The IPST can significantly change the decisions that the RM makes during critical situations. This section illustrates this point with the results of two different experiments. In the first there is a blue platform and a red platform. Initially, the red platform is not considered threatening based on a priori information. As such, the over-all threat level is low in the beginning, resulting in the blue RM using less conservative parameters. However when the red

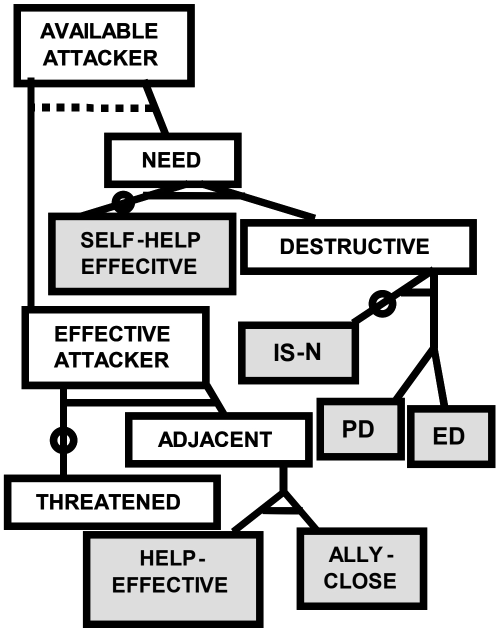


Figure 2: Subtree of MPDT.

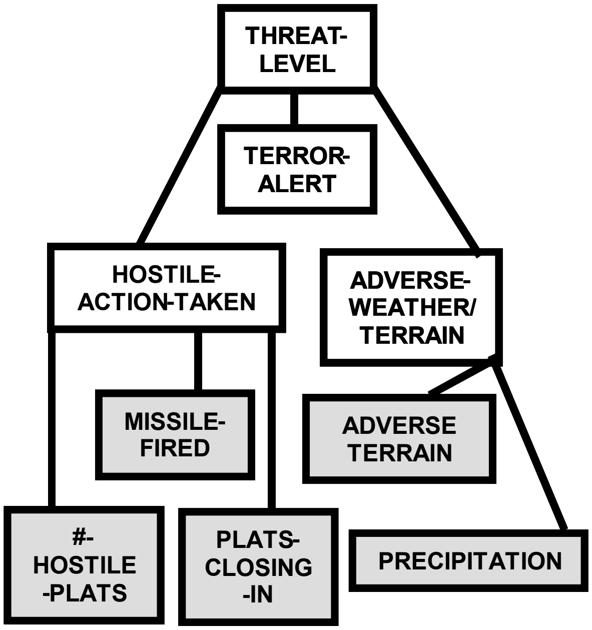
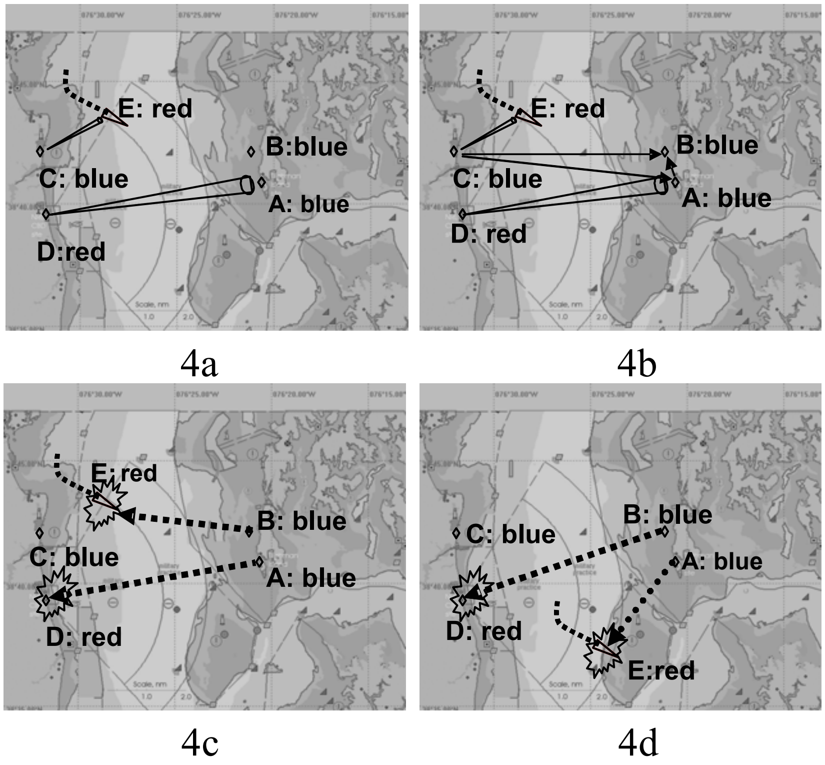


Figure 3: Subtree of IPST.



4a

4b

4c

4d

Figure 4: Three blue platforms in conflict with two red platforms. The basic scenario of experiments one through three.

platform fires a missile at blue, the IPST selects much more conservative parameters.

When “close” exceeds 0.5, blue fires chaff, a metallic airborne substance designed to interfere with radar detection [1]. The result is a neutralization of both the missile and the red radar platform. However, after the chaff loses its effectiveness, blue checks its EA library for the red radar platform and discovers that noise jamming [1] is the appropriate technique for this particular enemy. Noise jamming refers to putting obscuring noise on the enemy's radar display. As a result, blue executes noise jamming each time blue is in the main beam or sidelobe of red's radar.

The second experiment involves one blue platform and one red platform. The scenario begins with a blue platform moving toward a small mountain range. The blue platform discovers red, but the RM determines that because of distance and other factors that red is not a threat. However, as blue moves closer to the mountain range, the IPST increases the global threat level due to a possible ambush. The higher threat level results in more conservative parameters being selected for blue, resulting in significant changes in the values of the IPDT concepts "close", "banking", and "bearing_in".

6.5 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. The strategy tree concept has been discussed extensively in the literature [10].

6.6 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 take advantage of resources distributed over different platforms.

7. EXPERIMENTS

This section describes experiments that have been conducted to test various concepts on the fuzzy decision trees evolved by the GP. The tests were first conducted in full digital simulation with the idea of eventually conducting the same test using a combination of hardware and digital simulation, and finally field tests. The results reported here are from full digital simulation experiments.

7.1 Experiment one

The different platforms participating in experiment one are depicted in Figure 4. Platforms A and B are blue platforms with electronic support measures, electronic attack and communications resources. Neither A nor B has an onboard radar. Platform C is a blue land-based radar. Platforms A, B and C are allied blue platforms: each one runs its own copy of the RM.

Platform D is a red land based radar hostile to the blue platforms. Finally, platform E is a fast moving red platform, that is known to be hostile to the blue forces based on its ID determined by the blue ID system.

In Figure 4a, red platform D scans blue platform A, radar emissions are denoted by cones for both red and blue platforms. At the same time blue radar C has detected red platform E. Red platform E is a missile with a missile radar onboard. Platform E's trajectory is selected to produce a high grade of membership in the root concept "close" on platform C's IPDT. This is done by selecting red platform E's trajectory so it has a large radially inbound component of velocity directed toward blue platform C. In Figure 4b, blue platform A determines it is threatened by red platform D's radar scan. Blue platform A communicates this information to blue platform B. Communication is denoted by black arrows. Simultaneously, blue platform C communicates its detection of red platform E to blue platforms A and B.

For blue platforms A and B jamming is represented by a "dashed" arrow with its arrowhead surrounded by a "burst" pattern. As depicted in Figure 4c, the resource manager onboard blue platform B jams red platform E. Likewise platform A's RM determines red platform D should be jammed.

In Figure 4d, red platform E has moved. The RM's onboard blue platforms A and B exchange EA targets, that is, blue platform A jams red platform E and blue platform B jams red platform D. This is reasonable since as platform E moves, the blue platform that can most efficiently attack it changes.

7.2 Experiment two

The environment, platforms and resources in experiment two are identical to those of experiment one with the exception that in two, the blue ID system has incorrectly identified red platform E as a neutral.

Once the RMs on the blue platforms determine platform E is an enemy based on kinematics, the decisions of the RMs are the same as for experiment one when there was strong ID information. Experiments one and two, although simple, point out the RM's ability to determine a platform's intent using the output of an ID system or make such a determination based purely on kinematics.

On the IPDT, the concept "status" for these experiments has been allowed to take the value "NOT a friend." This is a concept suggested by the GP. The original concept in this box was "lethal." The concept "lethal" would have required that the ID system determine that platform E is an enemy before the concept "close" could be effective. Thus valuable kinematic information about red's hostile intent would never have been acted on. By using the GP's suggestion that "NOT a friend" be used in the "status," box, blue was able to act early against a particularly stealthy enemy. So experiment two supports the value of the GP evolved structure for the RM.

The RM requires no commanding platform for either experiment. Based on communications, in both experiments the roles of blue platforms A and B automatically switch at a reasonable time. This illustrates the RM's ability to allow a group of agents to self-organize. The RM's self-organizational properties arise from the MPDT. The MPDT must determine if the blue platform is "self-help-effective" and also if the red threat is in its queue of threats. The red platform may not be in its queue of threats if its is too far away, being dealt with by another blue platform, etc.

7.3 Experiment three

Experiment three differs only from experiment two in that at times red platform E is allowed to have a large absolute second time derivative of range so as to produce a high grade of membership in the root concept "ranging" on platform C's IPDT. As in the previous cases the RM gives the expected decisions and the various concept membership functions yield their anticipated values.

The three experiments of this section are in no sense the only experiments conducted to validate the RM. There has been many more, most of greater sophistication [6, 8, 10]. The simple experiments provided here were conducted to point of up various properties such as the ability of the RM to self-organize without a central commanding platform. Also the experiments show the RM's ability to make the best decision when good ID information is available from an ID system and to make good decision based on various kinematic properties of an incoming threat when information from an ID system is incorrect.

8. 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. A method for automatically determining fuzzy decision tree structure, and hence fuzzy if-then rules from military data bases is discussed. This method uses a genetic program, an algorithm that automatically creates other computer programs. The genetic program's structure is discussed as well as the terminal set, function set, the fitness function, termination criteria, population initialization, the operations of cross-over and mutation, and the construction of the data base used for data mining. The use of parsimony pressure to limit the length of the fuzzy decision tree, while maintaining the tree's effectiveness is discussed. An explicit fitness function including parsimony pressure is considered. An example of a fuzzy decision tree generated by this algorithm is examined. The advantages of a tree evolved through this process, differing from a related one obtained from human expertise are considered. The self-morphing property, the ability of a fuzzy decision tree to change its computational requirements in real-time is introduced. This concept is discussed in terms of a significant fuzzy decision tree evolved by a genetic program. Three experiments are discussed that exhibit the RM's self-organizing properties and its ability to make good decisions with or without good ID system information. The RM is shown to make good decisions based on the kinematic behavior of an enemy even when the output of an ID system is in error. Finally, the RM is shown to make good decisions even though the enemy uses significantly different strategies in each experiment.

REFERENCES

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

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

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

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

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

[6]Smith, III, J.F. and Rhyne, II, R. "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, CSREA Press, Las Vegas, NV, 1999, 669-675.

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

[8]Smith, III, J.F. and Rhyne, II, R.D., "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, CSREA Press, Las Vegas, NV, 2000, 421-428.

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

[10]Smith, III, J.F. and Rhyne II, R. D., "Fuzzy logic resource manager: tree structure and optimization**",** *Signal Processing, Sensor Fusion, and Target Recognition X*, I. Kadar, Vol. 4380, SPIE Proceedings, Orlando, 2001, 312-323.

[11] Gribbin, J. *Companion to the Cosmos*, The Orion Publishing Group Ltd, London, 1996, 299.