Proceedings of the 18th World Congress
The International Federation of Automatic Control
Milano (Italy) August 28 - September 2, 2011

A Hierarchical Fuzzy Decision Maker for the Weapon Target Assignment

M. Alper ŞAHİN * **Kemal LEBLEBİCİOĞLU** **

* *TUBITAK, Ankara, TURKEY*
(e-mail: alper.sahin@iltaren.tubitak.gov.tr)
** *Middle East Technical University, Ankara, TURKEY*
(e-mail: kleb@metu.edu.tr)

Abstract: The Weapon Target Assignment problem is the problem of assigning weapons to threats with the objective of minimizing the expected survival value of threats. In this work, a Hierarchical Fuzzy Decision Maker is considered as an alternative decision-aid system for the Weapon Target Assignment. Thus, a novel procedure is proposed to identify Hierarchical Fuzzy Decision Makers such that entire hierarchy satisfies a desired approximation performance even with a strict constraint on the size of rule base. The idea is to increase approximation performance through cascaded stages. The proposed procedure is applied to implement a Hierarchical Fuzzy Decision Maker for a number of randomly sampled Weapon Target Assignment instances on a realistic scenario. The simulation results show that identified systems have satisfactory performance and serviceable on the battlefield.

1. INTRODUCTION

The Weapon Target Assignment (WTA) refers to the reactive assignment of asset's weapons to engage or counter identified threats. The problem is to identify which weapon should be assigned to which threat. The main concern is minimizing the total expected survival value of the threats after all the engagements. On a real battlefield the WTA process requires to decide what to do, i.e., acts, close to real-time such that the acts are consistent with the mission's objectives, and compliant with platform and environmental constraints. Considering combinatorial complexity of the problem and the strict time constraint, a rule-based expert system seems to be a good alternative in helping commanders to decide on proper WTA. Besides, there exists imprecision and uncertainty on the observations in the battlefield. Thus we apply approximate reasoning theory to construct a rule-based solution for the particular WTA problem. The idea behind approximate reasoning is to associate the decision rules with fuzzy sets, then to model relations between rules by fuzzy relations and finally to compose rules by an inference method.

However, there is a serious limitation in training standard fuzzy systems with many variables which is termed as curse of dimensionality problem by Bellman (1961). To overcome the curse of dimensionality problem, hierarchial fuzzy systems were proposed in early 90s by Raju et al. (1991). A number of research has been done in identifying hierarchical fuzzy systems: A possible solution is first to supply the required hierarchical structure knowledge needed to develop a system and then to identify parameters of the system. Recently, Chang and Dunn Chung and Duan (2000) proposed a procedure to identify both incremental and aggregated hierarchical structures based the idea of ranking input variables. Wang Wang (1999) also studied on ranking input variables. Once the structure of the hierarchical system is carefully determined, the parameters of the system are optimized by gradient-descent techniques. See Campello and Amaral (2000), Chung and Duan (2000), Joo and Lee (2002), Wang (1999). However, they remain largely ineffective due to the difficulty caused by intermediate variables. Besides, we require the input space to be divided into very narrow partitions in order to guarantee universal approximation as in Wang (1999), Zeng and Keane (2005). Fukuda et al. (1994), Furuhashi et al. (1997), Tachibana and Furuhashi (1998), Wang et al. (2008) adopted genetic and evolutionary algorithms to identify both structure and parameters of the hierarchical fuzzy systems. However, it is well known that the global search algorithms are time consuming and computational prohibitive when a very high-dimensional problem is being considered.

Considering the current handicaps and deficiencies of hierarchical fuzzy modeling discussed above, we propose a novel procedure to identify hierarchial fuzzy decision makers such that the entire hierarchy satisfies a desired approximation performance even with a strict constraint on the size of rule base. The idea is to increase approximation performance through cascaded stages. The procedure applies the grid partitioning approach by Guillaume (2001) while generating rules. In this work, we apply the proposed procedure to identify a hierarchical fuzzy decision maker (HFDM) for the WTA problem. The simulation results show that identified systems have satisfactory performance and serviceable on the battlefield.

2. PROBLEM FORMULATION: RULE-BASED WEAPON TARGET ASSIGNMENT (WTA)

Research on WTA dates back to the 1950s where the modeling issues on the WTA were first investigated. See Day (1966), Manne (1958). Interested readers can refer to Eckler and Burr (1972), Murphey (1999) for a comprehensive review of the literature on the WTA

problem. The problem is formulated as a nonlinear integer programming problem with the objective of minimizing the survival value of threats. The problem variables are kill probabilities for each weapon-threat pairs and survival values of threats. Consider the assignment problem of w_x number weapons to t_x number threats. Let $p_{tw} \in [0,1]$ be the kill probability of the weapon w for the threat t and $v_t \in [0,1]$ be the survival value of the threat t where $t = 1, ..., t_x$ and $w = 1, ..., w_x$. The WTA problem is formulated as the following nonlinear integer programming problem:

$$\textit{minimize} \quad \sum_{t=1}^{t_x} v_t \prod_{w=1}^{w_x} (1 - p_{tw})^{d_{tw}} \tag{1a}$$

$$\textit{subject to} \quad \sum_{t=1}^{t_x} d_{tw} \leq 1, \quad \forall w = 1, ..., w_x \tag{1b}$$

$$d_{tw} \in \{0, 1\} \tag{1c}$$

where d_{tw} is the decision value indicating if w^{th} weapon is assigned to t^{th} threat. This formulation allows a preferential defense under a heavy attack such that it may be optimal to leave the threats aimed at low valued assets undefended and concentrate on destroying the threats aimed at high valued assets.

In this paper, we propose a decision-aid system to help commanders in WTA on a real battlefield. The idea is to define a rule-based mapping by approximate reasoning theory. Note that, a rule-based mapping is known to be efficient in computation time by its nature. Consider the assignment problem of w_x number weapons to t_x number threats. Let us collect all problem variables, i.e., v_t and p_{tw} where $t = 1, ..., t_x$ and $w = 1, ..., w_x$, in a $n = t_x + t_x w_x$ dimensional vector as follows:

$$\mathbf{x} = (x_1, x_2, ..., x_n) \tag{2a}$$

$$= (v_1, ..., v_{t_x}, p_{11}, ..., p_{t_x 1}, ..., p_{t_x w_x}) \in I \tag{2b}$$

where I is the set of all input vectors, i.e., $I = \{(x_1, x_2, ..., x_n) | x_i \in [0,1], \forall i = 1, ..., n\}$. Similarly, let us also collect all decision values, i.e., d_{tw} where $t = 1, ..., t_x$ and $w = 1, ..., w_x$, in a $m = t_x w_x$ dimensional vector as follows:

$$\mathbf{y} = (y_1, y_2, ..., y_m) \tag{3a}$$

$$= (d_{11}, ..., d_{t_x 1}, d_{12}, ..., d_{t_x 2}, ..., d_{t_x w_x}) \in O \tag{3b}$$

where O is the set of decision vectors, i.e., $O = \{(y_1, y_2, ..., y_m) | y_i \in \{0,1\}, \forall i = 1, ..., m\}$.

The set of decision vectors is actually a finite set of classes, i.e., $O = \{class_1, class_2, ..., class_c\}$ where $class$ is a possible assignment decision and c is the total number of possible assignment decisions, i.e., $c = 2^m$. For example, $class_1 = (0, 0, 0, ..., 0)_{1xm}$ means no assignment, $class_2 = (1, 0, 0, ..., 0)_{1xm}$ means that weapon $w = 1$ is assigned to only threat $t = 1$ and all other weapons are not assigned to any threat. In this context, the WTA problem can be modeled as a classification problem where an input pattern, i.e., $\mathbf{x} \in I$, is assigned to one of the given set of decisions, i.e., $\mathbf{y} \in O = \{class_1, class_2, ..., class_c\}$. Thus,

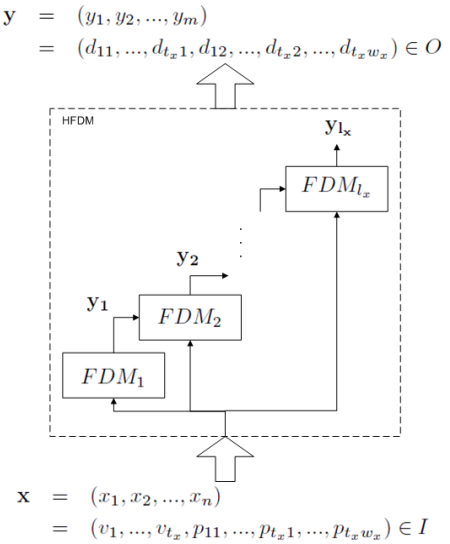

Fig. 1. The serial hierarchical structure of the Hierarchical Fuzzy Decision Maker (HFDM).

a FDM for WTA can be built as a single classifier, e.g., Mamdani-type classifier.

Suppose that, we have a set of q data tuples:

$$S = \{\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle_1, \langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle_2, ..., \langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle_q\} \tag{4}$$

where $\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle$ is the data tuple of the input vector $\bar{\mathbf{x}}$ and the corresponding decision vector $\bar{\mathbf{y}}$. Note that, the training data set S is supposed to be obtained off-line. Therefore, we may apply any exact solver since there is no time constraint on obtaining the data set S. Considering combinatorial complexity of the problem and the strict time constraint on the battlefield, a rule-based expert system seems to be a good alternative in helping commanders to decide on proper WTA. In the following, we discuss how to build a Hierarchical Fuzzy Decision Maker (HFDM) on a given training set, i.e., S.

3. MODELING OF HIERARCHICAL FUZZY DECISION MAKER (HFDM) FOR WEAPON TARGET ASSIGNMENT (WTA)

In the Fig. 1, we present an n input serial hierarchial fuzzy decision maker. We see that the hierarchy has l_x stages each of which is a fuzzy decision maker (FDM). The mathematical model of the hierarchy is written as follows:

$$\mathbf{y} = HFDM(\mathbf{x}) \tag{5}$$

$$= FDM_{l_x}(\mathbf{x}, FDM_{l_x - 1}(\mathbf{x}, FDM_{l_x - 2}(...)))$$

where F_l is the FDM corresponding lth stage, i.e., $l = 1, ..., l_x$. Note that, entire hierarchy reaches a final decision through contributions of each stage, i.e., FDM.

As depicted in the Fig. 1, the lowest stage, i.e., $l = 1$, takes only the input variables. Hence, the input is defined by the vector $\mathbf{x} = (x_1, x_2, ..., x_n) \in I$. Note that, $x_i \in [0,1]$ for all $i = 1, ..., n$. Suppose that, the domain interval of

each input variable, i.e., $x_i \in [0,1]$, is partitioned into h_i fuzzy sets, i.e., $A^i_{j_i}$ where $j_i = 1, ..., h_i$. Then, the lowest stage is characterized by the following Mamdani-type rule set:

$$R^1_{j_1,j_2,...,j_n} : \text{IF } \mathbf{x} \in A^1_{j_1,j_2,...,j_n} \text{ THEN } \mathbf{y_1} \text{ is } \mathbf{C^1_{j_1,j_2,...,j_n}} \quad (6)$$

where $A^1_{j_1,j_2,...,j_n}$ is the fuzzy set defining the rule's premise, i.e., $A^1_{j_1,j_2,...,j_n} = A^1_{j_1} \times A^2_{j_2} \times ... \times A^n_{j_n}$ and $\mathbf{C^1_{j_1,j_2,...,j_n}}$ is the rule's output. Note that, $\mathbf{C^1_{j_1,j_2,...,j_n}}$ is the most observed crisp decision, i.e., $\bar{\mathbf{y}} \in O$, in the set $S^1_{j_1,j_2,...,j_n} = \{\bar{\mathbf{y}} | \langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S, \bar{\mathbf{x}} \in A^1_{j_1,j_2,...,j_n}\}$.

For a Mamdani-type classifier, the output is determined by majority voting method. That is, if the rule $\text{RULE}^1_{j_1,j_2,...,j_n}$ has the largest firing strength for the input $\mathbf{x}$, the consequent of that rule is returned as the crisp decision. Let $\mu_{A^1_{j_1,j_2,...,j_n}}(.)$ be the membership function of the fuzzy set $A^1_{j_1,j_2,...,j_n}$. Thus, the lowest stage is expressed as follows:

$$\begin{aligned} \mathbf{y_1} &= FDM_1(\mathbf{x}) \\ &= \mathbf{C^1_{j_1^*,j_2^*..,j_n^*}}, \\ & \quad j_1^*, j_2^*.., j_n^* = \underset{j_1,..,j_n \in H}{\arg\max}\, \mu_{A^1_{j_1,...,j_n}}(\mathbf{x}) \end{aligned} \quad (7)$$

where $H = \{j_1, j_2, j_3, .., j_n | j_1 = 1, 2.., h_1;\ j_2 = 1, 2.., h_2;\ ...;\ j_n = 1, 2.., h_n\}$.

As depicted in the Fig. 1, any upper stage, i.e., $l > 1$, takes both of the crisp decision from its lower stage and the input variables. Let $\{class_1, class_2, ..., class_c\}$ be the set of all possible crisp decisions of the lower stage, i.e. $l-1$. Hence, the input is defined by $\mathbf{y_{l-1}} \in \{class_1, class_2, ..., class_c\} \subset O$ and $\mathbf{x} = (x_1, x_2, ..., x_n) \in I$. Note that, the idea is training each stage only on a certain region of the input domain where the approximation performance of the hierarchy is not satisfactory. Based on the given idea, we propose to apply grid partitioning according to an importance level indicating the approximation performance of the hierarchy anywhere on the input domain. Thus, the upper stages are characterized by two kind of rules: The first kind partitions only a certain region where approximation performance of the hierarchy is not satisfactory according to some criteria and called as tune rule. The second kind transfers the output from its lower stage to its upper stage for the inputs belonging the unpartitioned region and called as transfer rule. Note that, the tune rule overwrites the output from its lower stage. On the other hand, the transfer rule accepts the output from its lower stage and directs to its upper stage.

Let $U \subset I$ be the region where the approximation performance is identified to be unsatisfactory. Hence, we apply grid partitioning only on the region U. Suppose that, the domain intervals belonging to U, i.e., $x_i \in u_i$, is partitioned into h_i fuzzy sets, i.e., $A^i_{j_i} \in u_i$ where $j_i = 1, ..., h_i$. Recall, $\mathbf{y_{l-1}} \in \{class_1, class_2, ..., class_c\} \subset O$. Then, the tune rules are defined as follows:

$$R^l_{k,j_1,j_2,...,j_n} : \text{IF } \mathbf{y_{l-1}} \text{ is } class_k \text{ and } \mathbf{x} \in A^l_{j_1,j_2,...,j_n} \text{ THEN } \mathbf{y_l} \text{ is } \mathbf{C^l_{k,j_1,j_2,...,j_n}} \quad (8)$$

where $A^l_{j_1,j_2,...,j_n}$ is the fuzzy set defining the rule's premise, i.e., $A^l_{j_1,j_2,...,j_n} = A^1_{j_1} \times A^2_{j_2} \times ... \times A^n_{j_n}$ and $\mathbf{C^l_{k,j_1,j_2,...,j_n}}$ is the rule's output. Note that, $\mathbf{C^l_{k,j_1,j_2,...,j_n}}$ is the most observed crisp decision, i.e., $\bar{\mathbf{y}} \in O$, in the set $S^l_{k,j_1,j_2,...,j_N} = \{\bar{\mathbf{y}} | \langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S, \mathbf{y_{l-1}} = class_k, \mathbf{x} \in A^l_{j_1,j_2,...,j_n}\}$. Meanwhile, the transfer rule is defined as follows:

$$R_{I \setminus U} : \text{IF } \mathbf{x} \in (I \setminus U) \text{ THEN } \mathbf{y_l} \text{ is } \mathbf{y_{l-1}} \quad (9)$$

where $(I \setminus U)$ is the region where the approximation performance of the hierarchy is identified to be satisfactory and hence is the unpartitioned region.

Let $\mu_{A^l_{j_1,j_2,...,j_n}}(.)$ and $\mu_{class_k}(.)$ be the membership functions corresponding the fuzzy set $A^l_{j_1,j_2,...,j_n}$ and the crisp decision $class_k$, respectively. Note that, the lower stage, i.e., $\mathbf{y_{l-1}}$ is labeled as satisfactory if $\mathbf{y_{l-1}} \in \{I \setminus U\}$. Thus, the upper stages are characterized as follows:

$$\begin{aligned} \mathbf{y_l} &= FDM_l(\mathbf{x}, \mathbf{y_{l-1}}) \\ &= \begin{cases} \mathbf{y_{l-1}}, & \mathbf{y_{l-1}} \in \{I \setminus U\} \\ C^l_{k^*,j_1^*,...,j_n^*}, & \text{e.w.} \end{cases} \end{aligned} \quad (10)$$

where $k^*, j_1^*, .., j_n^*$ is the index of the rule with highest firing strength:

$$k^*, j_1^*, .., j_n^* = \underset{k,j_1,..,j_n \in G}{\arg\max}\, \mu_{class_k}(\mathbf{y_{l-1}}) \mu_{A^l_{j_1,...,j_n}}(\mathbf{x}) \quad (11)$$

where $G = \{k, j_1, j_2, j_3, .., j_n | k = 1, 2, ..., c;\ j_1 = 1, 2.., h_1; j_2 = 1, 2.., h_2; ...; j_n = 1, 2.., h_n\}$.

3.1 Grid Partitioning on the Importance of Input Variables

In grid partitioning of multi input systems, the choice of grid count carries significant consequences for each input variable. If it is too small, the rule base wont be accurate enough. On the other hand, if it is too large, the rule base will be too complex. To avoid fixing the grid count in grid partitioning, we propose to dynamically choose grids such that we concentrate on the region where approximation performance is not satisfactory.

Let $U \subset I$ be the region where the approximation performance is unsatisfactory. Suppose that, $|u_i|$ is the length of domain interval of the input variable x_i belonging to the region U, i.e., $x_i \in u_i \subset U$ where $i = 1, 2, ..., n$. Once the length of the domain intervals are identified, we evaluate the importance level of the input variables as follows:

$$i_i = \frac{|u_i|}{\sum_{j=1}^{n} |u_j|} \quad (12)$$

Suppose that we have an upper bound on the size of rule-base for each FDM, i.e., rc_x. Note that, in grid partitioning, the rule count is equal to the multiplication of the grid counts:

$$rc = \prod_{i=1}^{n} h_i \quad (13)$$

Having the importance levels, we propose to set the grid counts as follows:

$$h_i = \left\lfloor \sqrt[n]{\frac{rc_x}{n}} i_i \right\rfloor \quad (14)$$

where i_i is the importance level of ith input variable, i.e., x_i where $i = 1, 2, ..., n$.

3.2 A Procedure for HFDM Modeling

The idea is to train FDMs aware of approximation performance of the hierarchy such that the FDM will overwrite the decision where approximation performance is not satisfactory and transfer the decision where approximation performance is satisfactory. The procedure is flexible enough to build optimal hierarchical fuzzy decision makers by simple FDMs even with narrow rule sets. Based upon the idea just described, we may identify and implement a serial hierarchial fuzzy decision maker using the following procedure:

Step 1 Initialize the procedure

1.1) Set the size of the rule-base for each FDMs: rc_x.
1.2) Set the desired approximation error for the entire hierarchy: e_n.
1.3) Initialize the stage index: $l = 1$.

Step 2 Building stage $l = 1$

2.1) Evaluate the importance level for each input variable as in Chung and Duan (2000) where they proposed a method to rank input variables of hierarchical systems based on nonlinear regression.
2.2) Generate the rule-base:
  a) Set the grid counts, i.e., h_i where $i = 1, ..., h$, by (14).
  b) Partition the domain interval of each input variable, i.e., x_i, into h_i grids.
  c) Generate a rule for each partition by (6).
2.3) Build lowest stage as (7) and add to hierarchy.

Step 3 Evaluate the approximation performance

3.1) Initialize an error surface to indicate approximation performance of the hierarchy. To do so, the domain interval of each input variable is discretized into g equal partitions. That is, $Z = [0]_{nxg}$.
3.2) For each data tuple, i.e., $\langle \bar{\mathbf{x}}, \mathbf{y_{l-1}}, \bar{\mathbf{y}} \rangle \in S$, apply following steps. Note that, for the case of lowest stage, i.e., $l = 1$, the data tuple is in the form of $\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S$.
  a) Evaluate the output of uppermost FDM, i.e., lth stage:

$$\mathbf{y_l} = \begin{cases} FDM_1(\mathbf{x}), & l = 1; \\ FDM_l(\mathbf{x}, \mathbf{y_{l-1}}), & l > 1. \end{cases}$$

  b) Set $\mathbf{y_{l-1}} = \mathbf{y_l}$ and update the data tuple: $\langle \bar{\mathbf{x}}, \mathbf{y_{l-1}} = \mathbf{y_l}, \bar{\mathbf{y}} \rangle$.
  c) Evaluate the error of uppermost output, i.e., $\mathbf{y_l}$, to the optimal decision, i.e., $\mathbf{y}^*$, by the cost function characterizing the minimization problem (1): $e = COST(\mathbf{y_l}) - COST(\mathbf{y}^*)$.
  d) Update the error surface Z with the approximation error just evaluated. To do so, first, determine the location of $\mathbf{x}$ on the Z. Next, add the approximation error to identified location: $Z_{ig_i} = Z_{ig_i} + e$ where $g_i = \lfloor x_i/g \rfloor$ for all $i = 1, 2, ...n$. Note that, $x_i \in [0, 1]$.
  e) Update mean approximation error with the approximation error just evaluated: $e_c = e_c + e/q$ where q is the number of data tuples in the training set.

Step 4 Building stage $l > 1$

4.1) Identify the region where approximation performance of the hierarchy is unsatisfactory. To do so, first, determine a threshold level on the approximation performance to divide input set to satisfactory and unsatisfactory regions. Next, identify the region where approximation performance of the hierarchy is unsatisfactory: $U = \{(u_1, u_2, ..., u_n) \in I | Z_{ig_i} < th, g_i = \lfloor u_i/g \rfloor\}$ where $th = \sum_{i=1}^{n} \sum_{g_i=1}^{g} \frac{Z_{ig_i}}{gn}$.
4.2) Generate the tune rules:
  a) Evaluate the importance levels for each input variable by (12).
  b) Partition the domain interval of input variables, i.e., $x_i \in u_i$ into h_i grids.
  c) Generate a rule for each partition by (8).
4.3) Generate the transfer rule by (9), i.e., for the region where approximation performance of the hierarchy is identified to be satisfactory, i.e., $I \setminus U$.
4.4) Construct the lth stage by (10) and add to hierarchy.
4.5) Increment the stage index, i.e., $l = l + 1$, and go to step 3.

Termination Condition

T) Terminate the procedure if the mean approximation error, i.e., e_c, is lower than the desired error level, i.e., e_n; else go on with next stage.

4. RESULTS AND DISCUSSIONS

In this section, we analyze and discuss the performance of the HFDM model and the effectiveness of the proposed training procedure. We choose an illustrative function to evaluate the value of each threat as follows

$$v_t = 1 - r_t/r_x \quad (15)$$

where $r_t \in [0, r_x]$ is the range of the threat t and r_x is the maximum range. We choose the Gaussian function to characterize the kill probability of each weapon as follows

$$p_{tw} = a_t e^{-\frac{(r_t - b_t)^2}{c_t^2}} \quad (16)$$

where a is the maximum kill probability, b is the range where weapon is with the maximum kill probability and c is the width of the function's bell indicating the standard deviation in kill probability. As an example, we set the maximum kill probabilities as $a_1 = 0.95$, $a_2 = 0.8$, $a_3 = 0.7$ and $a_4 = 0.5$; the ranges as $b_1 = 5000$, $b_2 = 20000$, $b_3 = 40000$ and $b_4 = 80000$; the standard deviations as $c_1 = 3000$, $c_2 = 10000$, $c_3 = 10000$ and $c_4 = 40000$. Note that, the scenario is kept simple as compared to real scenarios since the main concern of the paper is to show the possibility to apply approximate reasoning theory to the WTA problem. In order to test how the HFDMs perform in practice, we apply K-fold cross validation in evaluating performance of the trained HFDM models. Note that, in K-fold cross validation, the original sample set is randomly partitioned into k subsets. $k-1$ subsets are used in training

Table 1. Analysis results

Solver	Approximation Error	Computation CPU Time (sec)
Hierarchical Fuzzy Decision Maker (HFDM) Stage rule count: 100 Depth of the hierarchy: 190	< 0.01	0.03
Hierarchical Fuzzy Decision Maker (HFDM) Stage rule count: 10000 Depth of the hierarchy: 89	< 0.01	1.43
Branch and Bound Algorithm (BBA) Branching and Bounding: Maximum Marginal Return Search: Breadth-First	0.0	13.56
Genetic Algorithm (GA) Population Size: 6 Generation Count: 6000 Crossover Probability / Operator: 0.8 / One-Cut Point Mutation Probability / Operator: 0.4 / Inverse Mutation	0.08	1.71

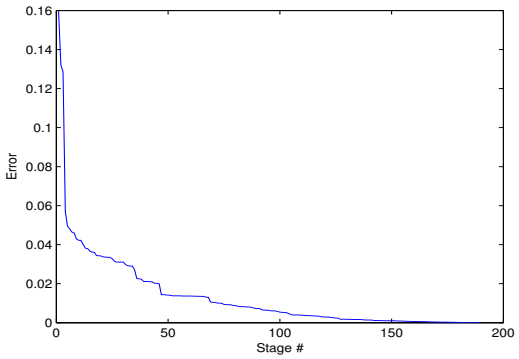

Fig. 2. The decrease of the approximation error at each stage through training. (100 Rule)

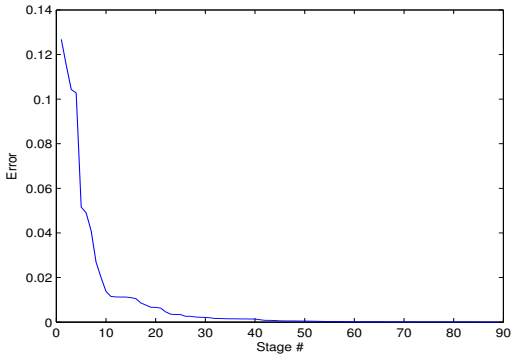

Fig. 3. The decrease of the approximation error at each stage through training. (10000 Rule)

and remaining one is kept as validation subset. Then, the cross-validation process is repeated k times such that each subset is used once as the validation set. Thus, all subsets are used both for training and validation. We implement models and perform all our tests with MATLAB 7.0 on a 2.4 GHz P4 processor with 3 GB RAM PC.

Defining the scenario, we sample 10^6 instances to form the data set where each instance corresponds an arbitrary vector of threat ranges. For each instance, we set threat values and kill probabilities by (15) and (16). Next, we calculate the exact decisions d_{tw} using a branch and bound algorithm for the particular v_t and p_{tw}. Next, we form each data tuple $\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle$ by (2) and (3). Finally, we apply 10-fold cross validation in testing the approximation performance of the HFDM.

First, we set the maximum size for the FDMs' rule-bases as 100, i.e., $rc_x = 100$. Next, we apply the procedure to train the hierarchy. As seen from the Fig. 2, the approximation error decreases steadily to zero. The depth of the hierarchy is 190. Note that, we apply K-fold cross validation in testing the trained models. Hence, we choose one of them as an illustrative example and give the approximation error curve in the Fig. 2. Second, we set the maximum size for the FDMs' rule-bases as 10000, i.e., $rc_x = 10000$. Next, we apply the procedure to train the hierarchy. As seen from the Fig. 3, the approximation error decreases steadily to zero. However, the hierarchy is more shallow compared to the hierarchy with $rc_x = 100$. Note that, this time the depth of the hierarchy is only 89.

As the performance test of trained HFDMs, we compute approximation error by comparing the HFDM output, i.e., $\mathbf{y}$, with the optimal assignment, i.e., $\mathbf{y}^*$. The approximation error is the difference between cost of optimal assignment with the cost of the HFDM output and normalize the difference by the cost of optimal assignment, i.e., $\left(COST(\mathbf{y}) - COST(\mathbf{y}^*)\right)/COST(\mathbf{y}^*)$. Note that, $COST(.)$ is the cost function characterizing the minimization problem (1). We report results in the Table 1 in the "Approximation Error" column. Besides, we also compare the approximation performance of the HFDM with the an exact solver and a suboptimal solver. To do so, we also implement a branch and bound algorithm by Ahuja et al. (2003) and a genetic algorithm by Lee et al. (2003) for the weapon target assignment problem. Note that, the approximation errors are the average of error in solver outputs to the optimal assignments. Furthermore, in the "Computation CPU Time" column, the time it takes to return the solution is given for each solver. Note that, the solvers are all implemented in MATLAB on a PC. It is quite possible to decrease the computation time considerably by implementing solvers on more sophisticated platforms as well as with more sophisticated programming tools. Therefore, it is quite better to check computation time comparatively.

The simulation results show that the HFDMs have satisfactory approximation performance as compared to exact and suboptimal solvers. Furthermore, the HFDMs are far beyond efficient in the sense of computation time compared to other solves as seen from the "Computation CPU Time" column due to the concurrent processing capability of the fuzzy systems. Note that, BBA and GA would have worse computation time as the cost calculation function gets more complex. On the other hand, the complexity of

the cost function does not any effect on the performance of the HFDM.

As we compare the HFDMs, we notice that the depth of the hierarchy nearly doubles as the rule count decreases by 1/100. As the complexity of each stage decreases, i.e., rule count decreases, the hierarchy becomes deeper. However, the HFDM having 100 rules per stage needs less time to return a solution. This is because the procedure trains a HFDM deeper than it should be. Since the procedure identifies only the grid counts in grid partitioning not the optimum grid centers, the rule bases are not necessarily optimal for any stage. Nevertheless, we believe that the procedure can be improved to identify the optimum grid centers. For example, a global search algorithm like the genetic algorithm can be easily merged to the procedure to identify the optimum grid centers.

5. CONCLUSIONS

This work proposes the HFDM as a decision-aid system for the WTA problems. Moreover, we propose a novel procedure to train the HFDMs such that each stage is trained only on a certain region of the input domain where the approximation performance of the hierarchy is not satisfactory. The trained HFDM returns a fine decision simply by tuning a coarse decision through cascaded stages. In this work, we apply the grid partitioning approach and we choose grids according to an importance level indicating the approximation performance of the hierarchy anywhere on input domain. The trained HFDMs have reported to have satisfactory performance in weapon target assignments. However, there exist a large number of much more sophisticated rule extraction methods in the literature. It would be a nice future work to apply state-of-the-art rule extraction methods for the WTA problem.

The procedure given above is flexible enough to construct different hierarchies from deep hierarchies with narrow rule sets to a single standard decision maker with complex rules sets. It should be noted that the memory required to implement a stage increases as the complexity of its rule-base increases. Besides, the response time of hierarchical structure increases as the depth of hierarchy increases. Thus, the size of rule base and the depth of hierarchy determines together the complexity of each stage, and hence the total memory to implement the entire hierarchy and the total computation time to return a decision for the trained hierarchy. Considering the trade of between the memory requirement and the response time, one should carefully determine the initial parameters of the procedure such as desired approximation error and rule count per stage. However, the procedure is lack of optimizing grid centers for the grid partitioning. It would be a nice future work to merge an rule-base optimization algorithm to the procedure.

REFERENCES

R.K. Ahuja, A. Kumar, K. Jha and J.B. Orlin. *Working Paper 4464-03: Exact and Heuristic Methods for the Weapon Target Assignment Problem.* MIT Sloan School of Management, Cambridge, MA, USA, 2003.

R. Bellman. *Adaptive Control Systems.* Princeton University Press, Princeton, NJ, USA, 1961.

R.J.G.B. Campello and W.C. Amaral. Optimization of hierarchical neural fuzzy models. *Proceeding of Internationel Joint Conference on Neural Networks,* Como, Italy, 2000.

F.L. Chung and J.C. Duan. On multistage fuzzy neural network modelling. *IEEE Transactions on Fuzzy Systems,* volume 8, issue 2, pages 125-142, 2000.

R.H. Day. Allocating weapons to target complexes by means of nonlinear programming. *Operations Research,* volume 14, pagse 992-1013, 1966.

A.R. Eckler and S.A. Burr. *Technical Report AD-A953517: Mathematical Models of Target Coverage and Missile Allocation.* Military Operations Research Society, Alexandria, USA, 1972.

T. Fukuda, Y. Hasegawa and K. Shimojima. Hierarchical fuzzy reasoning: Adaptive structure and rule by genetic algorithms. *Proceedings of Evolutionary Computation,* Orlando, FL, USA, 1994.

T. Furuhashi, S. Matsushita, H. Tsutsui and Y. Uchikawa. Knowledge extraction from hierarchical fuzzy model obtained by fuzzy neural networks and genetic algorithm. *Proceedings of International Conferecene on Neural Networks,* Houston, TX, USA, 1997.

S. Guillaume. Designing fuzzy inference Systems from data: An interpretability-oriented review. *IEEE Transactions on Fuzzy Systems,* volume 9, issue 3, pages 426-443, 2001.

M.G. Joo and J.S. Lee. Universal approximation by hierarchical fuzzy system with constraints on the fuzzy rule. *Fuzzy Sets and Systems,* volume 130, pages 175-188, 2002.

Z.J. Lee, S.F. Su and C.Y. Lee. Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. *IEEE Transactions on Systems, Man and Cybernetcis, Part B,* volume 33, issue 1, pages 113-121, 2003.

A.S. Manne. A target assignment problem. *Operations Research,* volume 6, issue 3, pages 346-351, 1958.

R.A. Murphey. Target-based weapon target assignment problems. *Nonlinear Assignment Problems: Algorithms and Applications,* P.M. Pardalos and L.S. Pitsoulis, Ed. Dordrecht: Kluwer Academic Publisher, pages 39-53, 1999.

G.V.S. Raju, J. Zhou and R.A. Kisner. Hierarchial fuzzy control. *International Journal of Control,* volume 54, pages 1201-1216, 1991.

K. Tachibana and T. Furuhashi. A hierarchical fuzzy modeling method using genetic algorithm for identification of concise submodels. *Proceedings of International Conference on Knowledge-Based Intelligent Electronic Systems,* Adelaide, Australia, 1998.

L.X. Wang. Analysis and design of hierarchical fuzzy systems. *IEEE Transactions on Fuzzy Systems,* volume 7, issue 5, pages 617-624, 1999.

W.Y. Wang, H. Li, S.C. Li, M.S. Tsai and S.F. Su. A dynamic hierarchical fuzzy neural network for a general continuous function. *Proceedings of Intertnational Conference on Fuzzy Systems,* Hong Kong, 2008.

X.J. Zeng and J.A. Keane. Approximation Capabilities of Hierarchical Fuzzy Systems. *IEEE Transactions on Fuzzy Systems,* volume 13, issue 5, pages 659-672, 2005.