Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Rule-Based Weapon Target Assignment On the Battlefield

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 (WTA) problem is the problem of assigning weapons to threats with the objective of minimizing the expected survival value of threats. On a real battlefield, the problem must be solved in quite short time, i.e., real-time. Considering combinatorial complexity of the problem and the strict time constraint, a fuzzy decision maker (FDM) is proposed as an alternative in helping commanders to decide on proper WTA. The idea is to build a FDM for a given data set using an extended grid partitioning based on sensitivity analysis of the input variables. The proposed decision maker is implemented and then tested with a number of randomly sampled WTA instances on realistic scenarios. The proposed system has a satisfactory performance.

1. INTRODUCTION

The Weapon Target Assignment (WTA) problem is a fundamental problem of Operations Research (OR) to model the defense of assets in a military conflict. The WTA refers to the reactive assignment of asset's weapons to engage or counter identified threats. On a real battlefield, weapons are allocated in stages such that weapons are assigned to threats with the outcomes (i.e., survival or destruction of each threat) of the previous stage are observed. Since there is quite limited time to observe, orient, decide and act during a real battle, the problem must be solved in quite shot time, i.e., real-time. Besides, there exists imprecision and uncertainty on the observations in the battlefield. Considering (a) the combinatorial complexity of the problem, (b) the strict time constraint on the battlefield, (c) the imprecision and the uncertainty on the observations, we propose to apply the approximate reasoning theory by Zadeh (1973) to construct a rule-based solution as an alternative decision-aid system in helping commanders to decide on proper WTA on the battlefield. The challenging task here is generating assignment rules to constitute a rule-base.

Early, expert knowledge was directly used to build rule-bases. Unfortunately, rule-bases build on the expert knowledge show poor performances especially for complex systems. Now, the rule generation is restricted to supervised learning and it is mainly based on automatic learning from a training data set. The first kind of rule generation method is the grid partitioning by Guillaume (2001) where input domain is divided into a number of fuzzy grids and all possible combinations form the fuzzy rules. Grid partitioning is also performed as decision tree induction. Interested readers may refer to Rokach and Maimon (2005) for a comprehensive survey of the literature on the decision tree inducers. The association rule mining algorithms are also very common examples of the grid partitioning. The second kind of rule induction method is the fuzzy clustering. In fuzzy clustering, the space is partitioned to clusters on the data set and a rule is associated to each cluster. The most widely used fuzzy clustering method is the Fuzzy C-means, see Dunn (1973).

We claim that a fuzzy decision maker can be used as a decision-aid system to help commanders in small scale WTA where both weapon and threat counts are small. In this work, we apply the grid partitioning to train a fuzzy decision maker. While applying the grid partitioning, we propose to partition the input domain in such a way that regions having a large effect on outputs are partitioned into smaller grids. Note that, this work is the first application of approximate reasoning theory to the WTA problem. The simulation results showed the superiority of the proposed model.

The paper is organized as follows: Section 2 introduces a formal formulation of the Weapon Target Assignment (WTA) problem. Section 3 presents the Fuzzy Decision Maker for the WTA. Section 4 discusses how to choose the grid counts in grid partitioning. Section 5 presents the simulation results. Several existing solutions were also employed for the comparison. The proposed decision aid system outperforms its competitors on all test problems. Finally, Section 6 concludes the paper.

2. PROBLEM FORMULATION: WEAPON TARGET ASSIGNMENT (WTA)

The WTA problem is formulated as a nonlinear integer programming problem with the objective of minimizing the survival value of threats. The problem variables are a kill probability for each weapon-threat pair and a survival value for each threat. 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 following nonlinear integer programming problem:

minimize
$$\sum_{t=1}^{t_x} v_t \prod_{w=1}^{w_x} (1 - p_{tw})^{d_{tw}} \tag{1a}$$
subject to
$$\sum_{t=1}^{t_x} d_{tw} \le 1, \quad \forall w = 1, ..., w_x \tag{1b}$$
$$d_{tw} \in \{0, 1\} \tag{1c}$$
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.

3. A FUZZY DECISION MAKER (FDM) FOR WTA

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., $O = \{(y_1, y_2, ..., y_m) | y_i \in \{0, 1\}, \forall i = 1, ..., m\}$. 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 $\bar{\mathbf{y}}$. In the following, we discuss how to build a Fuzzy Decision Maker (FDM) for WTA on a given training set, i.e., S.

We propose a cascaded FDM structure as depicted in the Fig. 1. The decision maker consists of two fuzzy rule-based systems: The Fuzzy Transformation Unit (FTU) and Fuzzy Classifier Unit (FCU). The first unit, FTU, approximates the data set to an intermediate output domain where assignment decisions is not necessarily integers. Next, the second unit, FCU, decides on valid weapon assignments for given FTU outputs. The FTU actually performs a nonlinear transformation of original input pattern to an intermediate pattern. The transformation is expected to increase the discrimination performance of the FCU layer leading to more accurate decision making. Moreover, the FCU takes a subset of intermediate variables instead of entire input pattern without worrying about information loss. Hence, the FCU can be build on less complex rule-bases. In the following, we proceed with details of each unit.

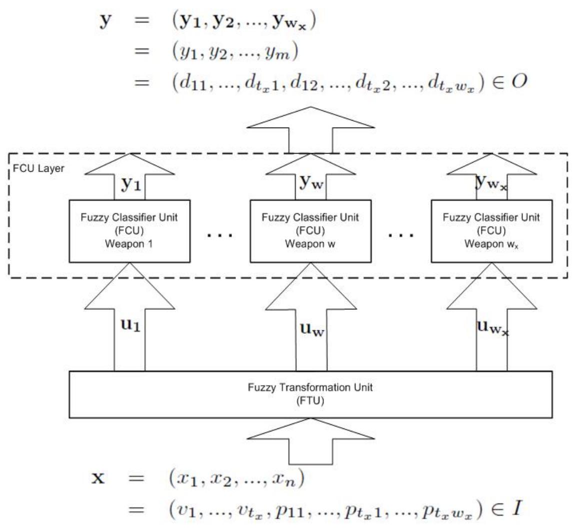
$\mathbf{y} = (\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_{w_x}})$
$= (y_1, y_2, ..., y_m)$
$= (d_{11}, ..., d_{t_x 1}, d_{12}, ..., d_{t_x 2}, ..., d_{t_x w_x}) \in O$
FCU Layer
$\mathbf{y_1}$
$\mathbf{y_w}$
$\mathbf{y_{w_x}}$
Fuzzy Classifier Unit (FCU) Weapon 1
Fuzzy Classifier Unit (FCU) Weapon w
Fuzzy Classifier Unit (FCU) Weapon w_x
$\mathbf{u_1}$
$\mathbf{u_w}$
$\mathbf{u_{w_x}}$
Fuzzy Transformation Unit (FTU)
$\mathbf{x} = (x_1, x_2, ..., x_n)$
$= (v_1, ..., v_{t_x}, p_{11}, ..., p_{t_x 1}, ..., p_{t_x w_x}) \in I$

Fig. 1. Structure of the Fuzzy Decision Maker (FDM).

3.1 Fuzzy Transformation Unit (FTU)

As depicted in the Fig 1, the FTU takes the input pattern (2). The goal is to approximate the input pattern to an intermediate pattern. The output is denoted by the vector $\mathbf{u} = (u_1, u_2, ..., u_m) \in U$ where U is the intermediate set where any FTU output, i.e., u_i, does not necessarily constitute a valid decision. Note that, $U = \{(u_1, u_2, ..., u_m) | u_i \in [0, 1]; \forall i = 1, ..., m\}$.

We apply the grid partitioning to extract the rule-base from the training data set S (4): Suppose that, the domain interval of each input variable is partitioned into h_i fuzzy sets: $A^i_{j_i}$, where $i = 1, ..., n$ and $j_i = 1, ..., h_i$. Let $\mu_{A^i_{j_i}}(.)$ be the membership function corresponds to fuzzy set $A^i_{j_i}$. Then, the FTU is characterized by the following rule set:

$$\text{RULE}_{j_1, j_2, ..., j_n} : \text{IF } \mathbf{x} \in A_{j_1, j_2, ..., j_n} \quad \text{THEN } \mathbf{u} \text{ is } \mathbf{B_{j_1, j_2, ..., j_n}} \tag{5}$$

where $A_{j_1, j_2, ..., j_n}$ is the fuzzy set defining the rule's premise, i.e., $A_{j_1, j_2, ..., j_n} = A^1_{j_1} \times A^2_{j_2} \times \ldots \times A^n_{j_n}$, and $\mathbf{B_{j_1, j_2, ..., j_n}} \in U$ is the rule's output. Note that, the output is evaluated by

$$\mathbf{B_{j_1, j_2, ..., j_n}} = \frac{\sum_{\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S} \mu_{A_{j_1, j_2, ..., j_n}}(\bar{\mathbf{x}}) \bar{\mathbf{y}}}{\sum_{\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle \in S} \mu_{A_{j_1, j_2, ..., j_n}}(\bar{\mathbf{x}})} \tag{6}$$

where $\mu_{A_{j_1, j_2, ..., j_n}}(\mathbf{x}) = \prod_{i=1}^{n} \mu_{A^i_{j_i}}(x_i)$.

Once the rule-base is formed, we represent the FTU as a Takagi-Sugeno (TS) model by Takagi and Sugeno (1973):

$$\mathbf{u} = \frac{\sum_{j_1,j_2,\ldots,j_n \in H} \mu_{A_{j_1,j_2,\ldots,j_n}}(\mathbf{x})\mathbf{B_{j_1,j_2,\ldots,j_n}}}{\sum_{j_1,j_2,\ldots,j_n \in H} \mu_{A_{j_1,j_2,\ldots,j_n}}(\mathbf{x})} \tag{7}$$

where $H = \{j_1, j_2, .., j_n | j_1 = 1, 2.., h_1;\ j_2 = 1, 2.., h_2;\ ...;\ j_n = 1, 2.., h_n\}$. Note that, a TS type fuzzy system does not require a separate defuzzifier.

3.2 Fuzzy Classifier Unit (FCU)

As depicted in the Fig. 1, there exist w_x number FCU in the FCU layer. Each is a fuzzy classifier and takes a part of the FTU output. The goal is to return assignment decision vector of the corresponding weapon. Each FCU, i.e, for the weapon w, is fed with the intermediate variables corresponding the weapon's benefit levels, i.e., $\left(u_{(w-1)t_x+1}, u_{(w-1)t_x+2}, ..., u_{wt_x}\right)$. Let O_w be the set of the possible assignment's decision vectors for the weapon w: $O_w = \{class_0^w, ..., class_t^w, ..., class_{t_x}^w\}$, where $class_0^w = (0, 0, .., 0)_{1xt_x}$ corresponds no assignment, $class_t^w = (0, 0, ..., 1, ...0)_{1xt_x}$ corresponds that the weapon w is assigned to the threat t. Having the input variables and the set of decision vectors, we define the classification problem of any weapon w as assigning an input pattern, i.e., $\mathbf{u_w} \in U_w = \{\left(u_{(w-1)t_x+1}, ..., u_{wt_x}\right) | u_i \in [0,1]; \forall i = 1, ..., m\} \subset U$, to one of the given set of decision vectors, i.e., $\mathbf{y_w} \in O_w = \{class_0^w, ..., class_t^w, ..., class_{t_x}^w\} \subset O$. Note that, the total number of input variables of a FCU is only t_x.

Similar to the FTU, we apply grid partitioning to extract the rule-base. To do so, we form another training data set S' from the data set S (4) as follows:

$$S' = \{\langle \bar{\mathbf{u}}, \bar{\mathbf{y}}\rangle \mid \langle \bar{\mathbf{x}}, \bar{\mathbf{y}}\rangle \in S, \bar{\mathbf{u}} = FTU(\bar{\mathbf{x}})\} \tag{8}$$

Having the training data set S', the domain interval of each input variable u_i, i.e., each output variable of the FTU, is partitioned into g_i fuzzy sets: $C_{j_i}^i$, where $i = 1, ..., m$ and $j_i = 1, ..., g_i$. Thus, the FCU is characterized by the following rule set:

$$\begin{aligned} \text{RULE}_{j_1,j_2,\ldots,j_{t_x}}^w \ :\ & \text{IF } \mathbf{u_w} \in C_{j_1,j_2,\ldots,j_{t_x}}^w \\ & \text{THEN } \mathbf{y_w} \text{ is } \mathbf{D^w_{j_1,j_2,\ldots,j_{t_x}}} \end{aligned} \tag{9}$$

where $C_{j_1,j_2,\ldots,j_{t_x}}^w$ is the fuzzy set defining the rule's premise, i.e., $C_{j_1,j_2,\ldots,j_{t_x}}^w = C_{j_{(w-1)t_x+1}}^{(w-1)t_x+1} \times C_{j_{(w-1)t_x+2}}^{(w-1)t_x+2} \times ... \times C_{j_{wt_x}}^{wt_x}$, and $\mathbf{D^w_{j_1,j_2,\ldots,j_{t_x}}}$ is the output. Note that, $\mathbf{D^w_{j_1,j_2,\ldots,j_{t_x}}}$ is the common decision encountered in the set

$$\begin{aligned} S_{j_1,\ldots,j_{t_x}}^w = \{&(\bar{y}_{(w-1)t_x+1}, \bar{y}_{(w-1)t_x+2}, ..., \bar{y}_{wt_x}) \\ &\mid \langle \bar{\mathbf{u}}, \bar{\mathbf{y}}\rangle \in S', \bar{\mathbf{u}}_\mathbf{w} \in C_{j_1,\ldots,j_{t_x}}^w\} \end{aligned} \tag{10}$$

Once the rule-base is formed, it is easy to obtain the corresponding fuzzy classifier. Let $\mu_{C_{j_i}^i}(.)$ be the membership function corresponding the fuzzy set $C_{j_i}^i$. Then, the firing strength of each rule is computed by the following equation:

$$\phi_{j_1,j_2,\ldots,j_{t_x}}^w(\mathbf{u_w}) = \frac{\mu_{C_{j_1,j_2,\ldots,j_{t_x}}^w}(\mathbf{u_w})}{\sum_{j_1,j_2,\ldots,j_{t_x} \in G_w} \mu_{C_{j_1,j_2,\ldots,j_{t_x}}^w}(\mathbf{u_w})} \tag{11}$$

where $\mu_{C_{j_1,j_2,\ldots,j_{t_x}}^w}(\mathbf{u_w}) = \prod_{i=1}^{t_x} \mu_{C_{j_{(w-1)t_x+i}}^{(w-1)t_x+i}}(u_{(w-1)t_x+i})$ and $G_w = \{\left(j_{(w-1)t_x+1}, j_{(w-1)t_x+2}, ..., j_{wt_x}\right) | j_{(w-1)t_x+1} = 1, 2.., g_{(w-1)t_x+1}; ...; j_{wt_x} = 1, 2.., g_{wt_x}\}$.

Having the firing strengths, the FCU output is evaluated by following inference mechanism:

$$\mathbf{y_w} = \mathbf{D^w_{j_1^*,j_2^*,\ldots,j_{t_x}^*}}, \tag{12a}$$

$$j_1^*, j_2^*, ..., j_{t_x}^* = \underset{j_1,\ldots,j_{t_x} \in G_w}{\arg\max}\ (\phi_{j_1,j_2,\ldots,j_{t_x}}^w(\mathbf{u_w})) \tag{12b}$$

4. TRAINING THE FDM MODEL

To apply the grid partitioning while training the FTU and the FCU models, we need to properly set the grid counts, i.e., h_i and g_j. Obviously, the input variables having high sensitivity on the output should be partitioned with larger grid counts. Based on this rationale, a rule-base can achieve satisfactory performance even with small upper bound on rule count. In this section, we propose a methodology that uses a sensitivity level for each input variable to set grid counts while applying grid partitioning.

4.1 Sensitivity Analysis of Input Variables

In statistics, non-linear regression method suggests to approximate a non-linear function by a linear one. See Seber and Wild (1989). Given a multi-input multi-output non-linear function $[y_1, y_2, ..., y_m] = f(x_1, x_2, ..., x_n)$, we may approximate on a set of q data derived from the function, i.e., f, as follows:

$$[\Delta Y] = [\Delta X][K] \tag{13}$$

where

$$[\Delta Y] = \begin{bmatrix} \bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_2 \\ \bar{\mathbf{y}}_1 - \bar{\mathbf{y}}_3 \\ \vdots \\ \bar{\mathbf{y}}_i - \bar{\mathbf{y}}_j \\ \vdots \\ \bar{\mathbf{y}}_{q-1} - \bar{\mathbf{y}}_q \end{bmatrix} \tag{14}$$

and

$$[\Delta X] = \begin{bmatrix} \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2 \\ \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_3 \\ \vdots \\ \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j \\ \vdots \\ \bar{\mathbf{x}}_{q-1} - \bar{\mathbf{x}}_q \end{bmatrix} \tag{15}$$

Each element of $[K]$ in (13) implies an approximate rate of change in output w.r.t rate of change in input. Thus, we may evaluate sensitivity level for any input variable x_i as follows:

$$l_i = \frac{\| k_{i1}\ k_{i2}\ \ldots\ k_{im} \|}{\sum_{j=1}^{n} \| k_{j1}\ k_{j2}\ \ldots\ k_{jm} \|} \tag{16}$$

where $[K] = ([\Delta X]^T[\Delta X])^{-1}[\Delta X]^T[\Delta Y]$.

4.2 Rule Extraction Based on Sensitivity Analysis

First, consider the FTU model: Based on the sensitivity analysis just described, a sensitivity level for each input variable can be obtained, i.e., l_i for each x_i where $i = 1, ..., n$. To do so, first, we form the input variation matrix, i.e., $[\Delta X]$, and the output variation matrix, i.e., $[\Delta Y]$, on the training data set S (4) as in (14) and (15). Having the input variation matrix $[\Delta X]$ and the output variation matrix $[\Delta Y]$, we can easily evaluate sensitivity level of each input variable by (16).

Suppose that we have an upper bound on the rule count of the FTU model, i.e., rc_1. Note that, in grid partitioning, rule count is equal to multiplication of grid counts: $rc = \prod_{i=1}^{n} h_i$. Using the result of the sensitivity analysis, we can identify the grid count h_i as follows:

$$h_i = \left\lfloor \sqrt[n]{\frac{rc_1}{\prod_{j=1}^{n} l_j}}\, l_i \right\rfloor \tag{17}$$

for each input variable, i.e. x_i where $i = 1, ..., n$.

Now, consider the FCU layer: We form the input variation matrix, i.e., $[\Delta U]$, for the training data set S' (8) as follows:

$$[\Delta U] = \begin{bmatrix} \mathbf{\bar{u}_1} - \mathbf{\bar{u}_2} \\ \mathbf{\bar{u}_1} - \mathbf{\bar{u}_3} \\ \vdots \\ \mathbf{\bar{u}_i} - \mathbf{\bar{u}_j} \\ \vdots \\ \mathbf{\bar{u}_{q-1}} - \mathbf{\bar{u}_q} \end{bmatrix} \tag{18}$$

Note that, the output variation matrix $[\Delta Y]$ is already formed.

Similar to the method to identify the grid counts of the FTU, we identify grid count g_i of each input variable u_i of the FCU as follows:

$$g_i = \left\lfloor \sqrt[\bar{t}]{\frac{rc_2}{\sum_{w=1}^{w_x} \left[\prod_{j=(w-1)t_x+1}^{wt_x} l_j \right]}}\, l_i \right\rfloor \tag{19}$$

where rc_2 is the upper bound on total rule count for the FCU layer.

By adopting the methodology just described, we can set grid counts, i.e., implies grid sizes, properly for a given upper bound on total rule count. Moreover, rule-bases will be more specific for the input variables having significant affect on the output and general enough to cover variations for all input variables. Thus, we may implement more accurate fuzzy decision makers even with limited hardware, i.e, implies a bound on the rule count, compared to standard grid partitioning. Note that, the idea is to approximate the underlying function to a continuous and linear function. Obviously, the approximation error will increase as the underlying function deviates from being continues and linear.

5. SIMULATIONS AND RESULTS

In this section, we analyze and discuss the performance of the FDM model and the effectiveness of the proposed grid partitioning methodology. To do so, we consider two different illustrative examples. In order to test how the FDMs perform in practice, we apply K-fold cross validation by Kohavi (1995) in evaluating performance of the trained FDM models. We implement models and perform all our tests with MATLAB 7.0 on a 2.4 GHz P4 processor with 3 GB RAM PC.

As the first illustrative example, we define a simple scenario where weapon count is 2 and target count is at most 3, i.e., $w_x = 2$ and $t_x = 3$. We choose an illustrative function to evaluate the value of each threat as follows

$$v_t = 1 - r_t/r_x \tag{20}$$

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, i.e., w, for any threat, i.e., t, as follows

$$p_{tw} = a_t e^{-\frac{(r_t - b_t)^2}{c_t^2}} \tag{21}$$

where a_t is the maximum kill probability, b_t is the range where weapon is with the maximum kill probability and c_t is the width of the function's bell indicating the standard deviation in kill probability. As to the example, we set the maximum kill probabilities as $a_1 = 0.7$ and $a_2 = 0.4$, the ranges as $b_1 = 5000$ and $b_2 = 20000$, the standard deviations as $c_1 = 5000$ and $c_2 = 20000$.

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 (20) and (21). Next, we calculate the exact decisions d_{tw} using a branch and bound algorithm for the particular v_t and p_{tw}. Finally, we form each data tuple $\langle \bar{\mathbf{x}}, \bar{\mathbf{y}} \rangle$ by (2) and (3).

Having the data set, we apply 10-fold cross validation in testing approximation performance of the FDM. As a first test, we analyze training parameters on the performance of the trained models. In this respect, we vary rule counts for both the standard grid partitioning and the sensitivity based grid partitioning. Note that, the fuzzy sets are all normal, complete and consistent. We compare the FDM output, i.e., $\mathbf{y}$, with the optimal assignment, i.e., $\mathbf{y}^*$ to compute approximation and optimality error. As the approximation error calculation, we take norm of difference between optimal assignment with the FTU output, i.e., $\|\mathbf{y}^* - \mathbf{u}\|$ where $\mathbf{u} = FTU(\mathbf{x})$. As the optimality error calculation, we take the difference between cost of optimal assignment with the cost of the FDM output and normalize the difference by the cost of optimal assignment, i.e., $(COST(\mathbf{y}) - COST(\mathbf{y}^*))\,/COST(\mathbf{y}^*)$ where $\mathbf{y} = FCU(FTU(\mathbf{x}))$. Note that, $COST(.)$ is the cost function associated with the minimization problem (1). We report results in the Table 1 in the "Approximation Error" and the "Optimality Error" columns of the "Average of Simulations" column. Besides, we also compare the approximation performance of the FDM with the an

Table 1. Analysis results of the FDM models for the scenario 1

Solver	Approximation Error	Optimality Error	Computation CPU Time (sec)
Fuzzy Decision Maker (FDM) Standard Grid Partitioning FTUM Rule Count: 512 FCUM Rule Count: $27 + 27$	0.75	0.035	< 0.01
Fuzzy Decision Maker (FDM) Standard Grid Partitioning FTUM Rule Count: 19683 FCUM Rule Count: $343 + 343$	0.54	0.0015	0.028
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level FTUM Rule Count: 125 FCUM Rule Count: $12 + 36$	0.62	0.046	< 0.01
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level FTUM Rule Count: 2352 FCUM Rule Count: $27 + 64$	0.47	0.0013	< 0.01
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level FTUM Rule Count: 108864 FCUM Rule Count: $4 + 4$	< 0.01	0.0	0.078
Branch and Bound Algorithm (BBA) Branching: Maximum Marginal Return Bounding: Maximum Marginal Return Search: Breadth-First	-	0.0	0.46
Genetic Algorithm (GA) Population Size: 3 Generation Count: 300 Crossover Operator: One-Cut Point Crossover Probability: 0.8 Mutation Operator: Inverse Mutation Mutation Probability: 0.4	-	0.03	0.16

Table 2. Analysis results of the FDM models for the scenario 2

Solver	Approximation Error	Optimality Error	Computation CPU Time (sec)
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level FTUM Rule Count: 1260 FCUM Rule Count: $8 + 6 + 192 + 144$	1.68	0.255	< 0.01
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level FTUM Rule Count: 110592 FCUM Rule Count: $18 + 10 + 384 + 320$	1.35	0.031	0.11
Fuzzy Decision Maker (FDM) Grid Partitioning on Sensitivity Level FTUM Rule Count: 663552 FCUM Rule Count: $7 + 7 + 7 + 7$	< 0.01	0.001	0.48
Branch and Bound Algorithm (BBA) Branching: Maximum Marginal Return Bounding: Maximum Marginal Return Search: Breadth-First	-	0.0	13.56
Genetic Algorithm (GA) Population Size: 6 Generation Count: 6000 Crossover Operator: One-Cut Point Crossover Probability: 0.8 Mutation Operator: Inverse Mutation Mutation Probability: 0.4	-	0.08	1.71

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.

As the second illustrative example, we define another scenario where weapon count is 4 and target count is at most 6, i.e., $w_x = 4$ and $t_x = 6$. As the previous example, we characterize the threat values and the weapon kill probabilities by (20) and (21). We set the maximum kill probabilities as $a_1 = 0.7$, $a_2 = 0.8$, $a_3 = 0.8$ and $a_4 = 0.99$; the ranges as $b_1 = 50000$, $b_2 = 15000$, $b_3 = 15000$ and $b_4 = 5000$; the standard deviations as $c_1 = 3000$, $c_2 = 5000$, $c_3 = 5000$ and $c_4 = 1000$. Defining the scenario, we form a data set with 10^6 data tuples as we form in previous example.

Having the data set, similar to the previous example, we apply 10-fold cross validation in testing approximation performance of the FDM. We analyze training parameters on the performance of the trained models. In this respect, we vary rule counts for the sensitivity based grid partitioning. Note that, the fuzzy sets are all normal, complete and consistent. We report results in the Table 2.

In Tables 1 and 2, the simulation results show that the FDMs with moderate or large rule-bases have satisfactory approximation performance as compared to the exact and the suboptimal solvers, i.e, the BBA and the GA. Furthermore, the FDMs are far beyond efficient in the sense of computation time compared to other solves as seen from the "Computation CPU Time" column. Note that, the solvers are all implemented in MATLAB on a PC. It is quite possible to decrease the computation time considerably by implementing the solvers on more sophisticated platforms as well as with more sophisticated programming tools. Nevertheless, the FDMs still would be far beyond efficient in the sense of computation time due to the concurrent processing capability of the fuzzy systems. Moreover, the BBA and the GA would have worse computation time as the cost calculation function gets more complex. Note that, the complexity of the cost function does not any effect on the performance of the FDM. Therefore, it is quite better to check the computation time comparatively.

From the Table 1, it is evident that the FDMs trained by the sensitivity based grid partitioning could achieve desired performance with smaller rule-bases compared to the FDMs trained by the standard grid partitioning. Therefore, we would require less memory to implement the FDM in the hardware for a satisfactory accuracy. In addition, we notice that the choice of membership functions has small effect on the approximation and optimization errors. Thus, an optimization on membership functions could slightly increase the performance of the FDMs.

6. CONCLUSIONS

In this work, we proposed to use a FDM as a decision-aid system for the WTA problems. In this work, we apply the grid partitioning approach in extracting the rule-base. In the grid partitioning, we propose to choose grid counts considering sensitivity level of each input variable. The trained FDMs 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. Note that, the sensitivity level is computed on a given data set by approximating the underlying function to a linear continuous function. We believe that one can extend such idea further by choosing other functions to which the underlying function to be approximated.

There is a serious limitation in grid partitioning for the complex fuzzy systems with many variables, which is usually termed as the curse of dimensionality problem. See Raju et al. (1991). Suppose that, the WTA problem is to be solved for w_x weapons against a maximum t_x targets. When we divide each input variable into h fuzzy grids, a complete rule-base should have h^n fuzzy rules where $n = t_x + w_x t_x$. Obviously, it is infeasible to design a fuzzy decision maker for weapon target assignment problem with large number of input variables. Hierarchial fuzzy systems, proposed by Raju et al. (1991), provide a way to deal with the curse of dimensionality of standard expert systems. We believe that a hierarchial fuzzy decision maker for the weapon target assignment problem could be a nice future work.

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.

J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics,* volume 3, issue 3, pages 3257, 1973.

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.

R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceeding of Internationel Joint Conferences on Artificial Intelligence.* Montreal, Canada, 1995, pp. 11371143.

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 Cybernetics, Part B,* volume 33, issue 1, pages 113-121, 2003.

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

L. Rokach and O. Maimon. Top-down induction of decision trees classifiers - a survey. *IEEE Transactions on Systems, Man and Cybernetic, Part C,* volume 35, issue 4, pages 476487, 2005.

G.A.F. Seber and C.J. Wild. *Nonlinear Regression.* Wiley, New York, USA, 1989.

T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, Cybernetics,* volume 15, pages 116132, 1985.

L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics,* volume 3, issue 1, pages 28-44, 1973.