Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Evolving a Hierarchical Decision Making Mechanism Using Fuzzy Logic

Ulas Beldek*. Kemal Leblebicioglu**

**Electronic and Communication Engineering Department, Çankaya University, Ankara, Turkey (e-mail: u.beldek@cankaya.edu.tr).*
***Electrical and Electronics Engineering Department, Middle East Technical University, Ankara, Turkey (e-mail: kleb@metu.edu.tr).*

Abstract: In this study, a new hierarchical decision-making and decision-fusion mechanism is introduced for solving decision making problems in a consistent manner. This mechanism is constructed by using a genetic algorithm. The proposed mechanism employs fuzzy logic and a performance index determined based on the performance of decision-making agents at successive hierarchical levels. The mechanism is such that the decisions in previous levels are influential on the current level decisions according to the performance index introduced. This mechanism is tested on an artificial problem, namely finding the amount of faults in a four tank water system.

1. INTRODUCTION

Decision making is an important area where different kinds of decision making and fusion approaches are available: In most of the common applications, fuzzy sets and fuzzy logic are coupled with different machine learning techniques. A decision making problem could be formulated and solved in many different ways. But the main issue is the determination of the method to be used in solving the decision making problem. In (Xu, 2007) intuitionistic preference relations are put forward by intuitionistic fuzzy sets and they are used to develop a method in group decision making process. (Ben-Arieh et al., 2007) deals with reaching a consensus in group decision making among different experts having different amount of influence. (Martinez et al., 2007) is about handling and processing data from different sources of knowledge having different domains and scales. Some problems are carried out using more than one decision making techniques as proposed in (Kahraman et al., 2007). In (Ralescu et al., 2007) applicability of optimal aggregation of fuzzy concepts to introduce a framework for decision making is studied. In (Jin et al., 2007) a genetic algorithm (GA) is used to update fuzzy feature transformations in a decision making about a classification problem of bioinformatics. (Xu et al., 2007) is about multiple attribute group decision making under fuzzy environment. (Xu et al., 2007) is useful especially when attribute weights are partially known. It gives an answer to the question of how decision makers can update their preferences. In (Wang et al., 2006) weights of fuzzy opinions are optimized using two different measures. (Lu et al., 2006) concerns with a general framework for organizing the relations between decision makers at different hierarchical levels such that the leader's decision is influenced not only by its followers but also the followers influence the decisions of each other. In (Tsiporkova et al., 2006) alternative decisions are evaluated by assigning a list of values demonstrating the expert's preference for the alternatives to satisfy a multi-criterion situation; Next, these alternatives are aggregated via some weights which accounts for the relative opinion of the decision maker about the relative importance of the related criteria. (Pei et al., 2006) is about different ways to extract fuzzy decision rules from a fuzzy information system. (O et al., 2006) is a study where a decision-making is handled from the point of view of a machine learning technique. (Lee et al., 2006) is about construction of a genetic fuzzy agent for a scheduling system. (Pasi et al., 2006) is a study about group decision making. It concerns the construction of a majority opinion from individual opinions by two different techniques. The first one uses aggregation of different individual opinions by some aggregation operators. The second one concerns the majority opinion as a fuzzy subset. In (Li, 2005) fuzzy multi-attribute decision making problems with uncertainty are studied. The property of this study is that the proposed method both concerns and in a way demonstrates the subjective judgment and objective information. It also gives insight about determination of membership degrees and weights. The advantage of the proposed hierarchical decision making and decision fusion approach in comparison with the studies refereed is that our hierarchical approach is very flexible. While determining the final decisions, one could provide the basic relations between the hierarchical levels in many different ways: Construction style for performance index, and the style of the decision fusion between hierarchical levels can be selected, the use of different data representation and evaluation techniques with different optimization algorithms could be organized and changed depending on the requirements of the decision making problem in any hierarchical level. The main outcome of our approach is the flexibility it provides in allowing the composition of different problem solving utilities when necessary.

In this study, a hierarchical decision making method is constructed in order to solve decision making problems "as efficiently as desired". In this method, the decision making agents are developed successively at consecutive hierarchical levels by: The decision made in the lower level by a lower level decision maker is carried to upper level and new decision is taken by newly developing decision maker whilst considering the lower level decision taken via a measure of reliability determined by a performance graph demonstrating the partial success of lower level decision. The lower level decision is an integral part of development of upper level decision maker. For the proposed problem the decision makers are made of rules in terms of **IF..THEN** statements. For a given fuzzy membership distribution these rules are updated using genetic algorithm. The fusion of decisions of lower level and newly developing upper level depends on an averaging operation via the use of performance graph obtained for the lower level. This paper is organized as follows: In section 2 the proposed structure of the hierarchical decision making method is given in details. In section 3 the model problem is explained. In section 4 the hierarchical decision making approach is applied to the problem and the obtained results are demonstrated. In the conclusions part the influence of proposed method and future studies in this area are discussed.

2. THE STRUCTURE OF DECISION MAKING METHOD

A decision making problem is set initially. The decision making structure consists of different hierarchical levels. In the first level there exists a decision maker. The decision maker performs its decisions and these decisions are transferred into second level about the decision making problem. Meanwhile the performance of the decisions taken by the first level decision maker is demonstrated by a performance measure. The second level decision maker uses the decisions of first level decision maker and the performance of first level decisions to generate its own decisions: The outputs of second level decision maker is an updated version of first level decisions. The second level decision maker works such that the first level decisions which are possibly consistent are not affected much but the first level decision which may be weak are possible to be improved: This is accomplished using the performance measure indices for the first level decisions. The level by level hierarchical approach continues until Nth level in the same manner similar to the relation between first level and second level. The decision makers in all the hierarchies are developed using machine learning techniques. The main idea in this decision fusion technique is that the higher level decision makers are used in order to operate in the regions where the lower level decision maker is not so successful. The proposed hierarchical decision making method is summarized in Fig. 1.

3. THE EXAMPLE PROBLEM

The hierarchical decision making and decision fusion approach is tested in an example problem: Finding the amount of artificially created faults in a chosen tank in a four tank water system in single and multiple fault scenarios. Determination of fault amounts in this system is designed as a decision making problem. The structure of the system is shown in Fig. 2. The parameters of this system and the variables and the dynamic equations of the system with and without fault are given in (Kılıç, 2005).

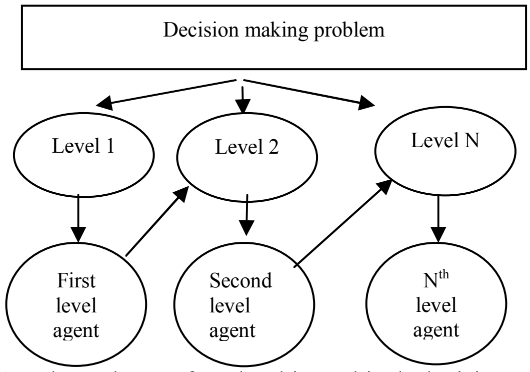

Fig. 1. The scheme for the hierarchical decision making method.

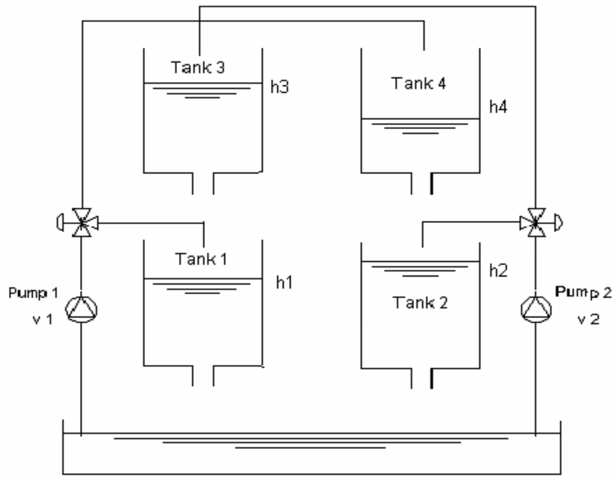

Fig. 2. Four tank water distribution system.

Failures are artificially created at the bottom of the tanks (there are two ways to create the failures: Either the holes at the bottom of the tanks increase or decrease in size) in the simulations. There are seven different level faults created. These levels are negative-big fault (means that the hole at the bottom of the tank is totally closed), negative medium fault (the nominal hole size is decreased 66 % percent), negative small fault (the nominal hole size is decreased 33 %), zero fault, positive small fault (the nominal hole size is increased 33 %), positive medium fault (the nominal hole size is increased 66%), positive big fault (the nominal hole size is doubled). The combination of all fault characteristics of all the four tanks is called as a scenario. The faults created in a scenario after some time instance ‘t’ remain the same after this time instance. 170 different scenarios are created having single and multiple fault structures. For each scenario, the water height levels of tanks due to original system equations and water height levels of tanks due to different scenarios are recorded for each time elapse of 0.1 seconds from time instance 0 to 20 (totally 201 height level data for each scenario).

The input for the decision making mechanism is the water height level differences between the original system and the system with fault for each scenario. The observed (system with fault) tank water height levels are subtracted from the should be observed (system without fault) tank water height levels to obtain the error data which is used as the input for decision making mechanism for each scenario for each time instance. This error data is normalized between the maximum and minimum height values for each tank so that each indices of error data is fixed between 1 and -1. The initial conditions for each tank water height values are taken the same for each scenario. The parameters related to tanks are unchanged through the simulations (ex the potential applied to the tanks). Eventually, for each scenario a normalized error data is obtained. The structure of the normalized error data (briefly error data) is shown in Table 1. In this table 'e_1', 'e_2', 'e_3' and 'e_4' symbolized the difference between the observed and should be observed water levels in each tank respectively, 't' represents the time instance these error values are taken (for two consecutive time instance the time elapse is 0.1 seconds), Scenario_1 represents the first scenario and Scenario_170 represents the last scenario.

Table 1. The configuration of error data (normalized error data)

Error Data	Scenario_1				…	Scenario_170			
t	e_1	e_2	e_3	e_4		e_1	e_2	e_3	e_4
1-									
…									
201-									

4. THE METHOD

4.1 First Level Simulation

A first level agent is developed using GA in the first level simulation. This simulation constitutes the first level of hierarchical decision making procedure. The chromosomes of the GA population is coded to represent the rules of the rule-base to accomplish the decision making task. All the potential rule-bases are created from 30 **IF...THEN** statements. The best chromosome obtained at the end of this simulation is declared as the first level prediction agent. This agent is used in order to predict the fault levels in each scenario at every time instance in the related tank. The agent yields an output for every time instance according to the input it is assigned from the normalized error data. This output is a normalized prediction of the related fault amount for the related scenario and for the corresponding time instance of the rule base using fuzzy logic principles (evaluating the related data using fuzzy evaluation according to the rule base enclosed in the body of the chromosome). There are totally 34170 time instances to be checked (170 scenario × 201 time instances for each scenario). The simple structure of a rule is show below:

IF ($e_{1,t}$='att_1' AND $e_{2,t}$='att_2' AND $e_{3,t}$='att_3' AND $e_{4,t}$= 'att_4') THEN (p_t= 'att_5')

In this rule structure '$e_{1,t}$', '$e_{2,t}$', '$e_{3,t}$' and '$e_{4,t}$' are input variables of the rule that take input from the normalized error data at time instance 't', 'p_t' is the output variable of the rule. It shows the prediction (decision) determined by that rule for the time instance 't'. In the simulation the input variables '$e_{1,t}$', '$e_{2,t}$', '$e_{3,t}$' and '$e_{4,t}$' may be assigned 8 different attribute values. These attribute values are 'negative-big', 'negative-medium', 'negative-small', 'zero', 'positive-small', 'positive-medium', 'positive-big' and 'not important'. Different from the input variables the output variable may be assigned 7 different attribute values excluding 'not important'. The distribution of membership functions for the input variable $e_{t,1}$ is shown in Fig. 3. The distribution of membership functions for other variables (both input and output) is the same as the distribution of membership functions of $e_{t,1}$.

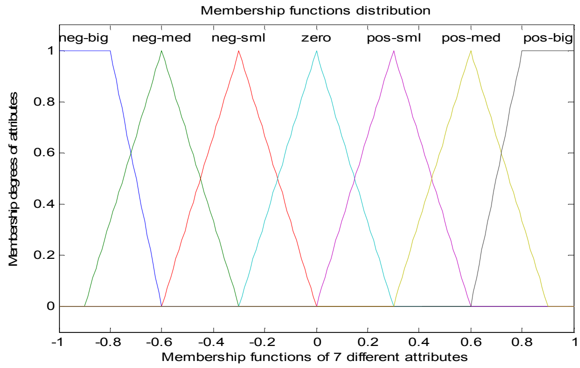

Fig. 3. Membership function distribution of each attribute for the variable $e_{t,1}$.

In the explained rule structure, the 'AND' operation works as taking the minimum of the membership values assigned to the inputs according to the associated membership function distribution. A simple example for 'AND' operation and its usage is shown in Fig. 4. If an input variable is assigned 'not important', this means whatever input the variable takes that variable is inactive and the membership value for that variable is assigned as 1 through the evaluation of the output of the rule. In a rule structures there exists 5 variables and there are totally 30 rules in a rule base enclosed in a chromosome. So there are totally 150 genes (30×5) in a simple chromosome. The aggregation of rules and determination of the output surface for the input using a rule base for a time instance is summarized in the example shown in Table 2: First the normalized error data corresponding to a time instance 't' is assigned as the input of the rule base enclosed in the chromosome. The rule base determines an output attribute and a membership degree for the output attribute for each of its rules using the same procedure briefly explained in Fig 4. For the rules which yields the same output attribute the maximum values of the membership degrees is assigned as the membership degree of the related output attribute. After finding the maximum membership values of the output attributes (there are totally 7 output attributes): according to the membership function distribution for these 7 attributes and the obtained maximum membership values for each attribute a maximum surface is determined. The centre of area of the maximum surface is declared as the normalized fault prediction for the considered time instance 't'. The centre of area defuzzification is summarized in (1).

$$pre_t = \frac{\int_{-1}^{1} x f(x) dx}{\int_{-1}^{1} f(x) dx} \quad (1)$$

In (1) f(x) symbolizes the surface equation of the obtained surface by membership functions and 'pre' symbolized the normalized prediction for the time instance 't'.

Input= [0.1 -0.7 0.8 -0.5]

↓

Rule-1: If ($e_{1,t}$='zero', $e_{2,t}$='neg-big' $e_{3,t}$='pos-big' $e_{4,t}$='pos-med') ise ($p_{1,t}$= 'pos-sml')

↓

Membership values for the input variables are calculated
$\mu(e_{1,t})=0.5$, $\mu(e_{2,t})=0.5$, $\mu(e_{3,t})=1$ $\mu(e_{4,t})=0.25$.
(μ symbolizes the membership value)
The minimum of the membership values is assigned as membership value of output variable
$\mu(p_t)=(0.5 \text{ MIN } 0.5 \text{ MIN } 1 \text{ MIN } 0.25)$
p_t= 'pos-sml' with membership value $\mu(p_t)=0.25$.

Fig. 4. Example: The membership value assignment procedure for the output variable of a rule due to an input.

Table 2. Example: Determination of membership values of all the 7 attributes (pos-big…neg-big) for a rule base having 7 rules.

Rules of the rule base		
Rule no:	**Output variable attribute**	**Membership degree of output variable**
1	Negative-big	0.35
2	Positive-small	0.17
3	Negative-big	0.05
4	Positive-small	0.13
5	Zero	0.11
6	Positive-small	0.05
7	Zero	0.14

Assigned membership values for output attributes	
Output attribute	**Membership degree**
Negative-big	max(0.35,0.05)=0.35
Negative-medium	0 (inactive)
Negative-small	0 (inactive)
Zero	max(0.11,0.14)=0.14
Positive-small	max(0.17,0.13,0.05)=0.17
Positive-medium	0 (inactive)
Positive-big	0 (inactive)

The defuzzified output is declared as the predicted normalized fault amount in the analyzed tank in the simulation. The output is a value between -1 and 1 and it shows how much normalized change has occurred in the tank. To give example if the output is 0 this means there is no change in the nominal hole size (0% change), if it is 1 this means the hole size is doubled (100% positive change in hole size)or if it is -1 this means the hole is totally closed (100% negative change in hole size). The other output values are obtained by a linear relation between these two extreme values 1 and -1.

For a chromosome the cost function is obtained by taking the sum of absolute differences between the predicted normalized fault amounts and real normalized fault amounts as in (2).

$$Cost = \sum_{k=1}^{170} \sum_{t=1}^{201} \left| pre_t(k) - real\_nor\_fault_t(k) \right| \quad (2)$$

In (2) 'Cost' represents the cost obtained for a chromosome $pre_t(k)$ represents the normalized prediction for the time instance 't' for k^{th} scenario, $real\_nor\_fault_t(k)$ represents the normalized real fault value for time instance 't' for k^{th} scenario. The fitness of a chromosome is obtained by taking the reciprocal of the cost function. The parameters and the obtained results for the first level GA simulation for Tank 1 are as below:

The number of chromosomes in a population: 40
The number of genes in a chromosome: 150.
Crossover operation: At each 25 gene one point crossover.
Crossover ratio: 90%
Reproduction: 10 % (with elitist method)
Gene mutation rate: 2% (Only applied to chromosomes which are crossovered. If the best fitness of generation does not exceed previous level best chromosome for 8 consecutive generations the crossover rate is increased to 10 % until an improvement in the best chromosome in the population is improved. Mutation rate is returned to its nominal value after the improvement.)

The fitness value of best chromosome at each generation is shown in Fig.5.

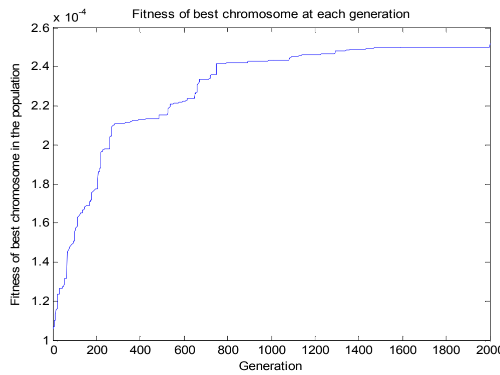

Fig. 5. Fitness of best chromosome in the population at each generation.

From the results it can be concluded that the cost function is minimized in a moderate amount. In the second level simulations this cost function is minimized much further using the hierarchical decision making technique and the decision fusion approach explained in the previous part.

4.2 Second Level Simulation and Decision Fusion in the Hierarchy:

In order to obtain the second level decisions first level decisions obtained for every time instances are used. In this part the main aim is to fuse the decisions obtained at the first level simulations with the decisions of a newly being developed rule base. The fusion technique in this level has a very simple structure. The first level decisions are transferred to second level simulations with an activation amount. The second level simulation agent used the decisions of first level agent, the activation values obtained for the decisions of the first level agent and the decision of the rule base and fuse them through the activation value of the first level decision and second level decisions are obtained. The second level agent is also as a result of an optimization. The fusion of previous level decisions and the decisions of newly developing rule base is a simple averaging operation according to the activation value o the pervious level decisions. For second level decisions a new rule base is proposed: This rule base could be assigned input from the current and one previous time instance of the normalized error data. The rule base is used to make decision for the current time instance using the fuzzy evaluation principle explained in first level simulation. The rules in the rule base could be briefly summarized as follows:

IF ($e_{1,t}$='att_1' AND $e_{2,t}$='att_2' AN D $e_{3,t}$='att_3' AND $e_{4,t}$='att_4' AND $e_{1,t-1}$='att_5' AND $e_{2,t-1}$='att_6' AND $e_{3,t-1}$='att_7' AND $e_{4,t-1}$='att_8') THEN p_t='att_9'

In this rule structure '$e_{1,t}$', '$e_{2,t}$', '$e_{3,t}$' and '$e_{4,t}$' represent input variables which take input from the current time instance of normalized error data, '$e_{1,t-1}$', '$e_{2,t-1}$', '$e_{3,t-1}$' and '$e_{4,t-1}$' represent the input variables which take input from one previous time instance or normalized error data and p_t stands for the output variable representing the output for the considered time instance. The rule base consisting of these new rules uses the same membership value assigning principles as the first level simulation and the distribution of the membership functions for all the variables are the same as first level simulation. In order to find the output of the rule base for any time instance 't' same principles are used as in first level simulation.

In order to fuse the decision of first level agent and the decision obtained using this new rule base to obtain the second level decisions, an approximate performance graph is used showing the capability of the first level agent at different situations (different places of prediction). This performance graph is obtained as follows: First, the total sum of absolute differences between the real normalized fault amount and the predicted fault amount are calculated for each data corresponding to its class attribute (negative-big, .., positive-big). These sums are divided by the number of data in the corresponding class that shows the error amount for the corresponding class. If these error amounts are subtracted from 1, approximate success rates of the first level agent for different classes are obtained. By a spline interpolation the success rates obtained for different classes are combined and an approximate performance graph is obtained. This performance graph shows approximate consistency of any decision made. In Fig. 6 the performance graph for the first level decisions made is shown.

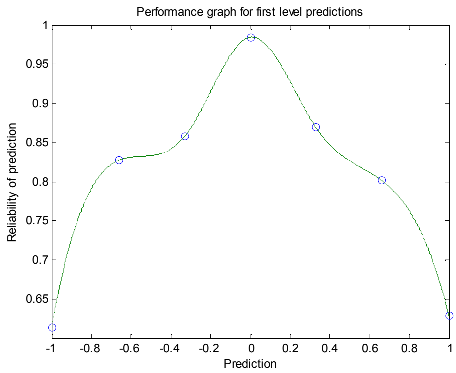

Fig. 6. The performance graph showing the reliability of the a first level prediction.

It could be observed from the performance graph that a prediction of '0' (the prediction is that there is no fault in the tank) has a confidence between 0.95 and 1. For example a prediction of 0.2 has a confidence between 0.9 and 0.95. For every prediction performed its confidence value is the activation value of that prediction.

The fusion of the first level prediction with the prediction of the new rule base to yield the second order prediction is given in (4)

$$slp = (flp \quad act\_flp) + (nrbp \quad (1 - act\_flp)) \qquad (4)$$

In (4), 'slp' is the second level prediction, 'flp' is the first level prediction, 'act_flp' activation value of the first level prediction, 'nrbp' is the prediction of new rule base. Second level prediction operation is an integral part of optimization procedure. Optimization includes both the prediction of the new rule base and the fusion between the first order predictions and the predictions of new rule base. The cost function for the second level simulation is shown in (5).

$$Cost = \sum_{k=1}^{170} \sum_{t=1}^{201} \left| slp_t(k) - real\_nor\_fault_t(k) \right| \qquad (5)$$

In (5) '$slp_t(k)$' represents second level prediction performed for the time instance 't' for the 'k^{th}' scenario, '$real\_nor\_fault_t(k)$' represents the real normalized fault value. The Fitness function is the reciprocal of the cost function.

The new rule structure has 8 input variables and 1 output variable. The rule base is made of 10 rules. This means a chromosome enclosing a rule base should have 90 genes (10× (8+1)). The second level simulation has the same GA parameters as the first level simulation. The obtained results for Tank1 are as follows:

Fitness of best chromosome at the end of simulation: 2.621×10^{-4}.

Cost of the best chromosome at the end of simulation: 3815
Amount of fault per data: 3815/34170 = 0.1116.
The improvement in cost with respect to first level simulation: (|3974-3815|/3974) ×100 = % 4

The fitness value of best chromosome at each generation is shown in Fig.7.

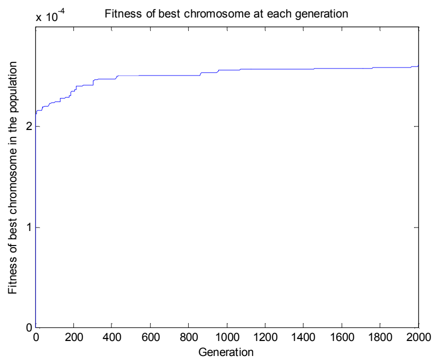

Fig. 7. The average fitness of population at each generation.

It is observed that there is an improvement in the fitness values when second level simulation is compared with the first level simulation. This is an expected since the decisions performed at the first level simulations are transferred to the second level simulation with activation amounts showing their approximate performance to perform good decisions. The second level predictions become more successful due to optimization. It is possible to increase the success rate using new decision fusion techniques and rule base structures.

5. CONCLUSIONS

In this study, a hierarchical decision making technique is proposed. The interaction between the levels is provided by a decision fusion technique depending on performance criteria of the previous level decisions. The technique is tested over an example problem and its efficiency is demonstrated: There is an improvement in the decisions made from the first level to the second level. It is planned to vary the structure of the rule bases and investigate the performance of new decision fusion techniques in forthcoming studies.

REFERENCES

Ben-Arieh, D. and T. Easton (2007). Multi-criteria group concensus under linear cost opinion elasticity, *Decision Support Systems,* **43,** 713-721.

Jin, B., Y.C. Tang and Y-Q. Zhang (2007). Support vector machines with genetic fuzzy feature transformation for biomedical data classification, *Information Sciences,* **177,** 476-489.

Kahraman, C., G. Büyüközkan, N.Y. Ateş (2007). A two phase multi-attribute decision-making approach for new product introduction, *Information Sciences,* **177,** 1567-1582.

Kılıç, E. (2005). *Fault Detection and Diagnosis in Nonlinear Dynamical Systems- Ph. D Thesis*, Middle East Technical University, Ankara.

Lee, C-S., C-C. Jiang and T-C. Hsieh (2006). A genetic fuzzy agent using ontology model for meeting scheduling system, *Information Sciences,* **176,** 1131-1155.

Li, D-F. (2005). An approach to fuzzy multiattribute decision making under uncertainty, *Information Sciences,* **169,** 97-112.

Lu, J., C. Shi and G. Zhang (2006).On bilevel multi-follower decision making: General framework and solutions, *Information Sciences,* **176,** 1607-1627.

Martinez, L., J. Liu, D. Ruan and J-B. Yang (2007). Dealing with heterogeneous information in engineering evaluation processes. *Information Sciences,* **177,** 1533-1542.

O, J., J. Lee, J. W. Lee and B-T Zhang (2006). Adaptive stock trading with dynamic asset allocation using reinforcement learning, *Information Sciences,* **176,** 2121-2147.

Pasi, G. and R.R. Yager (2006). Modelling the concept of majority opinion in group decision making, *Information Sciences,* **176,** 390-414.

Pei, Z., G. Resconi, A.J.V.D. Wal, K. Qin and Y. Xu (2006).Interpreting and extracting fuzzy decision rules from fuzzy information systems and their inference, *Information Sciences,* **176,** 1869-1897.

Ralescu, A.L., D.A. Ralescu and Y. Yamakata (2007). Inference by aggregation of evidence with applications to fuzzy probabilities, *Information Sciences,* **177,** 378-387.

Tsiporkova, E. and V. Boeva (2006).Multi-step ranking of alternatives in a multi-criteria and multi-expert decision making environment, *Information Sciences,* **176,** 2673-2697.

Wang, Y-M. And C. Parkan (2006). Two new approaches for assessing the weights of fuzzy opinions in group decision analysis, *Information Sciences,* **176,** 3538-3555.

Xu, Z. (2007). Intuitionistic preference relations and their application in group decision making. *Information Sciences,* **177,** 2363–2379.

Xu, Z-S. and J. Chen (2007). An interactive method for multiple attribute group decision making, *Information Sciences,* **177,** 248-263.