Int J Adv Manuf Technol (2003) 21:691–698
Ownership and Copyright

The International Journal of
Advanced Manufacturing Technology

An Efficient Expert System for Machine Fault Diagnosis

S. C. Liu[1] and S. Y. Liu[2]

[1]Department of MIS, National Pingtung University of Science and Technology, Pingtung, Taiwan 912; and [2]Department of Industrial Engineering and Management, I-Shou University, Kaoshiung, Taiwan 840

An efficient expert system for machine fault diagnosis is developed. A new search method is proposed in this system to improve the efficiency of the diagnostic process. First of all, a diagnostic tree (a decision tree) is built by domain experts according to the functions of the devices in the machine. Then, the diagnostic priorities of nodes (devices) in the tree are determined based on a fuzzy group multiple attribute decision making method. A meta knowledge base for fault diagnosis is generated automatically based on the determined priorities to guide the diagnostic process. After that, a domain knowledge base that hypothesises possible faults for each device in the tree is generated by domain experts and/or manuals. At last, the inference process starts based on the meta knowledge base and hypothesises which device is the possible cause of failure. To validate the system performance, an illustrative example (VCR troubleshooting) is presented for demonstration purposes.

Keywords: Domain knowledge base; Efficient expert system; Fault diagnosis; Fuzzy group multiple attribute decision making; Meta knowledge base

1. Introduction

A machine fault will cause economic losses and even human casualties. Diagnosis can approach the problem from all angles and find the cause with the least time, energy, and money wasted. Hence, fault diagnosis is necessary when a machine is malfunctioning [1]. Generally, an expert needs time to diagnose the fault and determine its cause. However, there are very few experienced mechanics in the market. Therefore, an expert system with the expertise of experienced mechanics is required [2].

In the past, various approaches have been used to produce better diagnostic expert systems. Among them, the rule-based diagnostic expert system is the most mature and the most promising [1,3,4]. However, many diagnostic rule-based expert systems suffer from the diagnosis inefficiency problem [2]. To guide the diagnosis, expert systems rely on an inference engine to derive the conclusions from the knowledge base. The inference engine must check all heuristic rules in the knowledge base based on backward chaining, forward chaining, or mixed modes of chaining. If the inference engine finds that any premise clause is unknown during the rule-checking process, the inference engine generates a query to ask the user. In order to answer these queries, the user may spend more time collecting related information and responding. Therefore, the diagnosis time increases significantly when the number of queries grows, and the diagnostic process slows down because of the inefficient search of the knowledge base [5–7].

For improving search efficiency, Vranes et al. [8] proposed a best-first search strategy based on fault probability information. The nodes with high fault probability components are generated and checked first, followed by those with lower fault probability. However, fault probability information is not the only criterion to determine the priority of each node. The other criteria such as the difficulty of device fault diagnosis, and the symptoms, should be considered. Liu and Liu [7] proposed a new search strategy to enhance the efficiency of the diagnostic process for air compressor troubleshooting. A diagnostic tree is constructed based on the functions and connectivity of the air compressor devices. A fuzzy multiple attribute decision making method for a single user is used to determine the priority of each node (device) in the tree. Then, the priorities of devices control the diagnostic process. However, the expert system for air compressor troubleshooting lacks generality. In addition, there might be more than one fault diagnosis expert and they may have different opinions about the importance of each criterion. Hence, a fuzzy group multiple attribute decision making method seems more appropriate for this problem [9,10].

The objective of this paper is to design and develop an efficient machine fault diagnosis expert system (EMFDES) via a fuzzy group multiple attribute decision making method to overcome the diagnosis inefficiency problem mentioned above. EMFDES is a hybrid expert system that combines an expert system and a fuzzy group multiple attribute decision making

Correspondence and offprint requests to: Dr S. C. Liu, Department of Management Information Systems, National Pingtung University of Science and Technology, Pingtung, Taiwan 91207. E-mail: sliu@mail.npust.edu.tw
Received 1 February 2002
Accepted 16 April 2002

method. This decision-making method is responsible for the determination of the most efficient inference process.

2. An Efficient Machine Fault Diagnosis Expert System (EMFDES)

To overcome the inefficient fault diagnosis problem, an EMFDES including four modules: diagnostic tree, fuzzy group multiple attribute decision making (fuzzy group MADM), knowledge base, and inference engine, is proposed. The framework of EMFDES is shown in Fig. 1. In the following sections, the system overview is first presented, and the four modules are described and discussed in that order.

2.1 System Overview

For developing EMFDES, users (domain experts) are asked to build a diagnostic tree according to the functional relationship of devices through the user interface. Each node on this tree represents a physical device. According to some useful information such as the fault probability, the diagnosis time, the job complexity, etc., the priority of each device is determined based on a fuzzy group MADM method. Then, a meta knowledge base is generated automatically based on the determined priority to guide the diagnostic process. In the meantime, the domain knowledge base for hypothesising the possible faults for each device is also created by domain experts. Then, the inference engine begins to diagnose the device with the highest priority and ends with the device with the lowest priority. Based on this search, the inference engine usually checks only part of the diagnostic tree and the knowledge base. Hence, the diagnosis efficiency is improved since the machine fault can be found without checking the whole knowledge base.

2.2 Diagnostic Tree

A diagnostic tree derived from information about where each component is located and how the components are connected to one another can provide the fundamental knowledge for fault diagnosis. In a diagnostic tree, the nodes represent the components of the device, and the branch linking any two nodes represents the parent–child relationship between these two nodes. The diagnostic tree divides the device into several

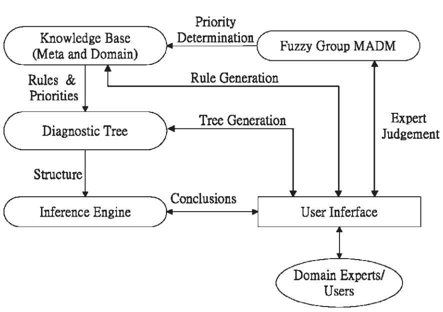

Fig. 1. The framework of EMFDES.

smaller devices, and provides the foundation for rule construction. A device can be divided into several small components based on the function and interrelation of these components, and each component may represent one independent module in the device. Similarly, these smaller components can be decomposed into even smaller components according to their specific functions and design objectives. This decomposition process continues until all the smallest, independent, and repairable components are defined. The diagnostic tree is then completely generated [7]. However, the following situation may occur during the hierarchical breakdown process. If two large components at the same level have a functional relation with the same smaller component at the next level, a "loop" may be formed among these three components. For computation purposes, a simple heuristic method is developed to break the loop and keep the relation among these components. Specifically, the domain expert can just duplicate the same smaller device and put each smaller device under both larger components. Therefore, the nodes in the tree structure do not tangle with one another, and the relation among these components still holds.

2.3 Fuzzy Group Multiple Attribute Decision Making

After the structure is represented by the diagnostic tree mentioned above, the priority of the nodes in the diagnostic tree should be determined. Since there is no certain criterion to evaluate the priority of each node, it cannot be directly evaluated by domain experts. According to the literature review mentioned above, a fuzzy group MADM method seems feasible to solve this problem. Generally, fuzzy group MADM methods refer to selection from several alternatives in the presence of multiple criteria, multiple people and fuzzy environments [11,12]. In the past few years, many fuzzy group MADM methods have been applied in different fields such as business, decision making, and expert systems [9].

For machine fault diagnosis, one criterion seems to dominate other criteria and the trade-off among criteria is non-compensatory (the compensatory model can also be applied in this proposed fuzzy group MADM method). Therefore, a fuzzy group MADM method based on the lexicographic approach (according to the order of importance of the criteria) is developed to help domain experts to determine the priorities of the nodes (devices) in the diagnostic tree. Before our fuzzy group MADM method is discussed, some terminology and formulae should be defined first.

Different types of fuzzy set have been proposed [13]. The trapezoidal fuzzy number is one of the widely used fuzzy sets. Since it has both the computational efficiency and the ease of data acquisition, the trapezoidal fuzzy number is selected for developing our fuzzy group MADM method. In addition, some formulae needed in the proposed fuzzy group MADM method are:

1. If the scale of a trapezoidal fuzzy number M is not within the interval [0,1],

   $$M = (a, b, c, d)$$

   The normalisation process is applied as follows:

(*a*) The attribute of M is a benefit type

$$M' = \left[\frac{a - a^*}{d^* - a^*}, \frac{b - a^*}{d^* - a^*}, \frac{c - a^*}{d^* - a^*}, \frac{d - a^*}{d^* - a^*}\right]$$

(*b*) The attribute of M is a cost type

$$M' = \left[\frac{d^* - d}{d^* - a^*}, \frac{d^* - c}{d^* - a^*}, \frac{d^* - b}{d^* - a^*}, \frac{d^* - a}{d^* - a^*}\right]$$

where a^* is the smallest value for the specific criterion and d^* is the largest value for the specific criterion.

2. The following addition formula is used to combine two fuzzy numbers:

If $M_1 = (a_1, b_1, c_1, d_1)$
$M_2 = (a_2, b_2, c_2, d_2)$

Then $M_1 + M_2 = (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2)$

3. The conversion formula proposed by Chen and Hwang [9] for a fuzzy number is listed below:

If a trapezoidal fuzzy number M,

$$M = (a, b, c, d)$$

the crisp number C for M is

$$C = \frac{1}{2}\left(\frac{d}{1 - c + d} + \frac{b}{b - a + 1}\right)$$

The detailed steps of the proposed fuzzy group MADM method for machine fault diagnosis are as follows:

Step 1. Determine the criteria to judge the priorities of the devices according to the opinions from domain experts.

Step 2. Determine the degree of importance for these criteria and what types (benefit or cost) of criteria they are.

Step 3. Define linguistic terms for each criterion. These linguistic terms (e.g. high, average, low) should be able to distinguish different degrees of importance for that criterion.

Step 4. Define the fuzzy membership function for each linguistic term with respect to each criterion for each domain expert.

Step 5. Perform the normalisation process for any linguistic term of the specific criterion if the scale of the linguistic term for the criterion is not located in the interval [0,1].

Step 6. Begin at the highest level of the diagnostic tree, and each domain expert assigns suitable linguistic terms to each criterion regarding all devices at that level.

Step 7. Combine criteria by adding their corresponding membership function values assigned by all domain experts if several domain experts participate in this evaluation.

Step 8. Compare all devices to consider the criterion with the highest rank first. The membership function values (a fuzzy number) for the criterion are converted into a crisp number. The priorities of devices are determined by their corresponding crisp number. The larger the crisp number, the higher the priority.

Step 9. Combine these criteria by adding their corresponding membership function values and go to step 8 if several criteria possess the same ranking.

Step 10. Obtain the criterion (or criteria) with the next highest rank and compare these tied devices by steps 7, 8, and 9 if there are several devices tied with the same crisp number.

Step 11. Proceed in this manner until the priorities of all devices are determined.

Step 12. Go down to the next level of the diagnostic tree and go back to step 6. This priority determination process continues until the lowest level of the diagnostic tree is reached.

After the proposed fuzzy group MADM method is performed, a prioritised diagnostic tree is generated. The priorities of these devices are in terms of the graphic location. The higher the location of the device, the higher the priority. If the trade-off among criteria is compensatory, some steps must be modified in this procedure. Steps 8~11 combine these criteria by adding their corresponding membership function values and multiplying their relative importance of criteria. The combined membership function values (a fuzzy number) are converted into a crisp number. Compare all devices at the specific level based on the crisp number. If ties occur, determine the priority of the device based on the criterion with the highest rank. The other steps are the same in this procedure.

2.4 Knowledge Base

EMFDES includes two types of knowledge base: a meta knowledge base and a domain knowledge base. The meta knowledge base is generated automatically based on the priorities determined by the proposed fuzzy group MADM method. The purpose of the meta knowledge base is to guide the diagnostic process efficiently. To guide the inference process, a tree search method is helpful to develop the meta knowledge base. Generally, there are two types of tree search method: breadth-first search and depth-first search [14]. For our problem, the depth-first search method seems more feasible and efficient than the breadth-first search method since each node contains several branches as its child nodes and the result is located at the deepest (or lowest) level of the diagnostic tree [7]. Therefore, the depth-first approach is selected for developing our meta knowledge base. The rule generation process will go right until the node is terminated. If a node is terminated, the diagnostic process will back up one level, go down, and then go right until another terminal node is encountered. This process will last until the rightmost and lowest level node is encountered.

The domain knowledge base is derived from experienced experts (shallow knowledge) and/or repair manuals (deep knowledge). The domain knowledge base gathers the associated heuristic rules that can hypothesise the possible causes of machine failures. Each device has its own domain knowledge base, and each domain knowledge base is responsible for hypothesising the possible faults for that device. Figure 2 illustrates the relationship between domain knowledge base and the machine components. The rule expression is as follows:

IF premise 1
AND premise 2
THEN conclusion (confidence factor = 0.7)

The confidence factor is used to represent uncertainty. It reflects a subjective estimate of the confidence for the validity of the rule.

2.5 Inference Engine

The inference engine searches for machine faults according to the meta knowledge base to hypothesise which device is the possible cause of failure and starts at the highest level. Once a device is selected and hypothesised, the corresponding domain knowledge base for that device is used to prove that the hypothesised device is at fault. If the evidence confirms the hypothesis, the inference engine moves down to the next level under that device if the next lower level exists. If there is not enough evidence, the inference engine will hypothesise another device with lower priority at the same level as the hypothesised device and find the associated domain knowledge base to derive evidence for that device. The inference engine will find the smallest responsible device. The detailed inference process of EMFDES is shown in Fig. 3.

In this paper, enough evidence is defined as that in which there are several rules supporting the hypothesis, and their combined confidence factor is larger than a specified threshold value (this confidence factor union method is selected since it is widely employed [15]). The formula is given as:

$$C(cf) = cf_1 + cf_2 - cf_1 \times cf_2 \quad (1)$$

where $C(cf)$ is the final combined confidence factor and cf_1 and cf_2 are the confidence factor of rules 1 and 2, respectively. For example, if a device at fault is hypothesised, EMFDES must check the domain knowledge base for the specific device. If the threshold value is assigned to be 0.75, and rules 1 and

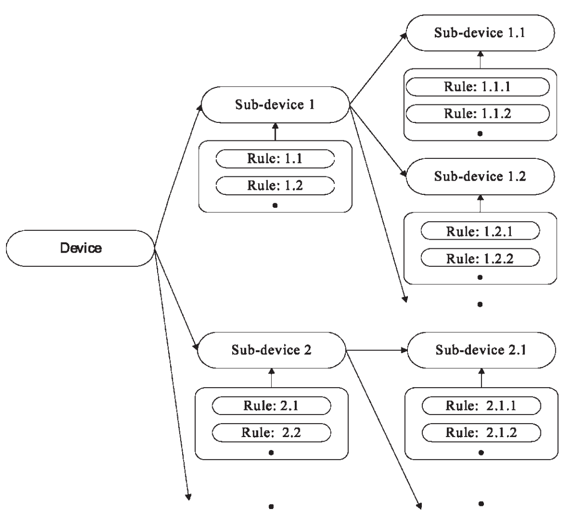

Fig. 2. The domain knowledge base for the diagnostic tree.

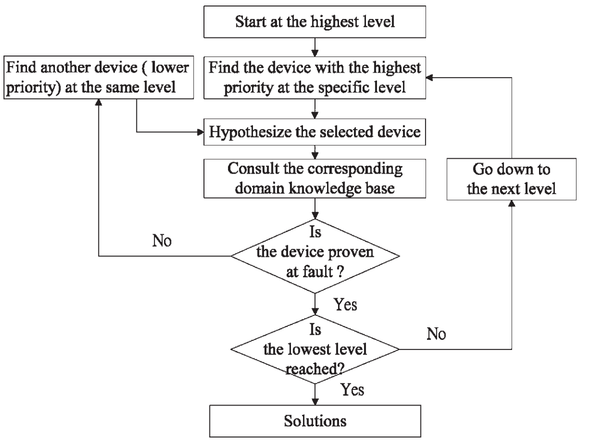

Fig. 3. The process for the inference engine.

2, with confidence factor 0.5 and 0.7, respectively, for supporting the specific device are triggered, the combined confidence factor becomes 0.85 according to the confidence factor union method (i.e. $0.5 + 0.7 - 0.5 \times 0.7 = 0.85$). This value is higher than the threshold value. It concludes that the device is at fault.

3. System Configuration

EMFDES has been successfully implemented in Visual Basic on a PC. It provides a comfortable human–machine interface for domain experts to define the diagnostic tree, determine criteria, develop the knowledge base, and set experts' evaluation (Fig. 4).

EMFDES includes six functions: FILE, TREE, CRITERIA, RULE, EXPERT EVALUATION, and RUN functions. The FILE function allows users to maintain a machine fault diagnosis problem. The TREE function allows users to define a diagnostic tree. The CRITERIA function allows users to determine which model (compensatory or non-compensatory) and criteria should be considered. The RULE function allows users to develop a domain knowledge base. The EXPERT EVALUATION function allows users to evaluate the importance of each device according to each criterion. Furthermore, a meta knowledge base for the machine fault diagnosis is generated automatically. The RUN function allows users to execute the software and find the results.

4. An Illustrative Example

In order to illustrate the system of EMFDES, one sample application description is presented for demonstration purposes. Video cassette recorders (VCRs) are complex electronic and mechanical devices. They rely on special proprietary integrated circuits and zero tolerance mechanical alignment [16]. An expert system for VCR fault diagnosis is developed based on the structure of EMFDES. The detailed description is given in the following section.

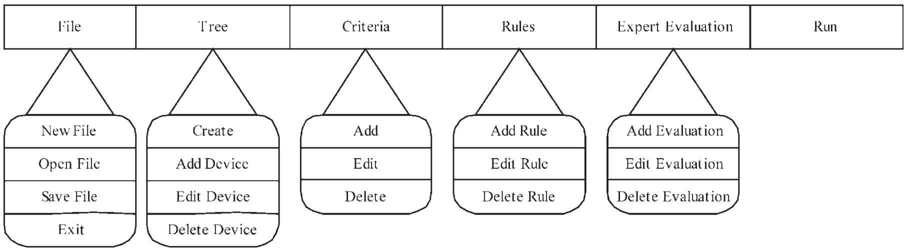

Fig. 4. The main menu of EMFDES.

4.1 Diagnostic Tree

Based on the VCR repair manual and specifications [16–18] the diagnostic tree of the VCR is constructed (Fig. 5). In Fig. 5, a VCR is divided into three major devices: tape control, video board, and track control. Each device is also composed of several smaller devices. For example, the tape control can be broken down into three smaller devices: belt, mechanism, and gear. This breakdown process continues until the smallest repairable units have been defined.

4.2 Fuzzy Group Multiple Attribute Decision Making

Based on the proposed steps mentioned in Section 2.3, the first step is to define the evaluation criteria (or attributes). According to the opinions from domain experts, the following three criteria are selected to determine the relative importance of these components:

1. Fault probability.
2. Time for diagnosis.
3. Complexity of diagnosis.

Once the three criteria are defined, step 2 is to determine the rank and the type of these three criteria. After we consult with domain experts, the relative importance among these criteria is determined. In addition, this model is non-compensatory for criteria. Fault probability is the most important criterion, time for diagnosis is the second most important one, and complexity of diagnosis is the least important. In addition, fault probability is a benefit type of factor. The other two criteria, time for diagnosis and complexity of diagnosis, are cost types of factor.

Step 3 is to define different linguistic terms to distinguish the difference for each criterion according to the opinions from domain experts (Table 1). In Table 1, the linguistic terms for the criterion "time for diagnosis" from the highest degree to the lowest degree are "very large", "large", "medium", "small", and "very small".

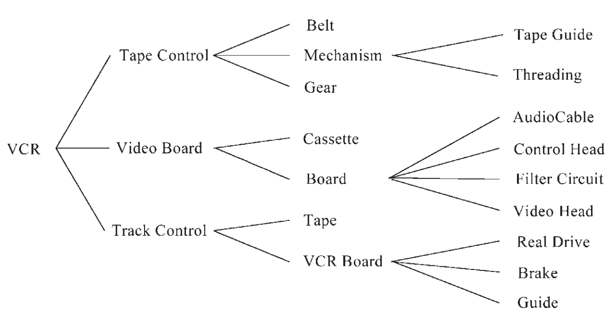

Fig. 5. The diagnostic tree structure of the VCR.

Step 4 is to define the membership functions of these linguistic terms according to the opinions from domain experts. For instance, if the linguistic term "large" of the criterion "time for diagnosis" (Fig. 6) has the most probable value 1 between 20 and 22 min, and this value gradually decreases to 0 within 5 min, the membership function of this linguistic term is written in (15, 20, 22, 27). Similarly, the rest of the linguistic terms for the other criteria can be defined in the same way (Figs 7 and 8).

Step 5 is to perform the normalisation process for any linguistic terms in a specific criterion if it is required. For example, since the scale of the "time for diagnosis" criterion ranges from 0 to 30 and exceeds the interval [0,1], the normalisation process is required. A linguistic term "small" can be used to demonstrate the normalisation process.

"small" = (3, 8, 10, 15).

Since this criterion is the cost type, the normalized fuzzy number for the linguistic term "small" is

$$\text{“small”} = \left[\frac{30-15}{30-0}, \frac{30-10}{30-0}, \frac{30-8}{30-0}, \frac{30-3}{30-0}\right]$$
$$= (0.5, 0.667, 0.733, 0.9)$$

The other linguistic terms of the criterion "time for diagnosis" can be calculated in the same way. Figure 9 shows new membership functions of all linguistic terms after the normalisation process is performed.

According to step 6, the highest level of the diagnostic tree contains three components: "tape control", "video board", and "track control". The appropriate linguistic terms are assigned by a domain expert to each criterion for these three components (Table 2).

According to step 7, since there are two experts in this evaluation, the other expert assigns his weighting to each component according to each criterion (Table 3).

Step 8 is to compare these components. Since "fault probability" is the most important criterion, it is the first one selected for comparison. The linguistic terms of these three components (Tables 2 and 3) are then converted into the crisp values according to the conversion formula mentioned above. The detailed computation is:

"Tape control" = (0.6, 0.75, 0.8, 0.95) + (0.6, 0.75, 0.8, 0.95) = (1.2, 1.5, 1.6, 1.9)

Table 1. The linguistic terms for the criteria.

Criteria	Linguistic terms				
Fault probability	Very high	High	Average	Low	Very low
Time for diagnosis	Very large	Large	Medium	Small	Very small
Complexity of diagnosis	Very high	High	Average	Low	Very low

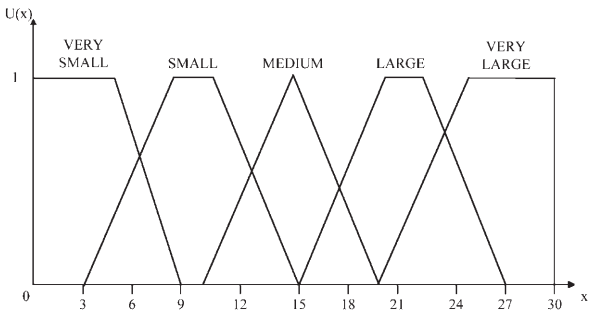

Fig. 6. The membership functions of the linguistic terms for "time for diagnosis".

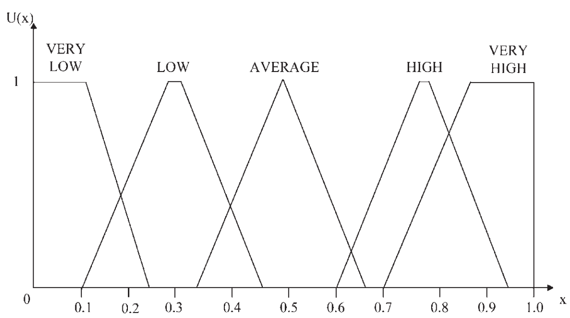

Fig. 7. The membership functions of the linguistic terms for "fault probability".

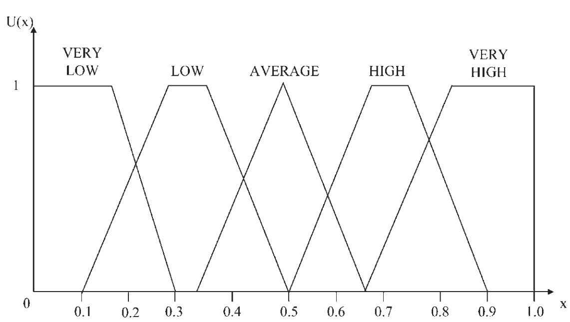

Fig. 8. The membership functions of the linguistic terms for "complexity of diagnosis".

$$= \frac{1}{2}\left(\frac{1.9}{1 - 1.6 + 1.9} + \frac{1.5}{1.5 - 1.2 + 1}\right) = 1.3$$

"Video board" = (0.35, 0.5, 0.5, 0.65) + (0.6, 0.75, 0.8, 0.95) = (0.95, 1.25, 1.3, 1.6)

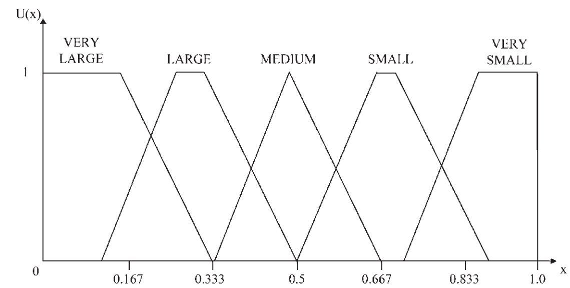

Fig. 9. The normalised fuzzy number for "time for diagnosis".

Table 2. The assigned linguistic terms for these three components (the first expert).

Criteria	Tape control	Video board	Track control
Fault probability	High	Average	High
Time for diagnosis	Medium	Large	Medium
Complexity of diagnosis	High	Average	Average

Table 3. The assigned linguistic terms for these three components (the other expert).

Criteria	Tape control	Video board	Track control
Fault probability	High	High	High
Time for diagnosis	Large	Large	Medium
Complexity of diagnosis	High	Average	Average

$$= \frac{1}{2}\left(\frac{1.6}{1 - 1.3 + 1.6} + \frac{1.25}{1.25 - 0.95 + 1}\right) = 1.1$$

"Track control" = (0.6, 0.75, 0.8, 0.95) + (0.6, 0.75, 0.8, 0.95) = (1.2, 1.5, 1.6, 1.9)

$$= \frac{1}{2}\left(\frac{1.9}{1 - 1.6 + 1.9} + \frac{1.5}{1.5 - 1.2 + 1}\right) = 1.3$$

The result shows that the priorities of the components "tape control" and "track control" are the highest, and the "video board" has the lowest priority. Notice that there is a tie between "tape control" and "track control". To distinguish the priorities between these two components, the next important criterion "time for diagnosis" must be used. Similarly, the crisp value of these two components for the "time for diagnosis" criterion is calculated as follows:

"Tape control" = (0.333, 0.5, 0.5, 0.667) + (0.1, 0.266, 0.333, 0.5) = (0.433, 0.766, 0.833, 1.167) = 0.725

"Track control" = (0.333, 0.5, 0.5, 0.667) + (0.333, 0.5, 0.5, 0.667) = (0.666, 1.0, 1.0, 1.334) = 0.875

The final priority of the components at the first level has now been determined. The priority of these components from high to low is "track control", "tape control", and "video board". By repeating steps 6 to 12, the prioritised diagnostic tree for VCR is built (Fig. 10). The higher the device is located in Fig. 10, the higher priority it has.

4.3 Knowledge Base

There are 19 control rules generated automatically in the meta knowledge base according to the prioritised result mentioned above. The following rules are part of the meta knowledge base:

RULE 1: IF VCR is bad;
THEN check track control.
RULE 2: IF track control is bad;
THEN check VCR board.
RULE 3: IF VCR board is bad;
THEN check brake.
RULE 4: IF brake is good;
THEN check reel drive.
RULE 5: IF reel drive is good;
THEN check guide.
RULE 6: IF VCR board is good;
THEN check tape.
RULE 7: IF track control is good;
THEN check tape control.
•
•

By guiding the diagnostic process, the meta knowledge base allows domain experts to concentrate on hypothesising the possible faults for one device at a given time.

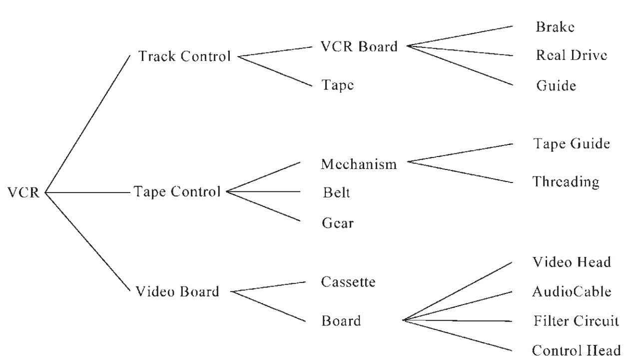

Fig. 10. The prioritised diagnostic tree for VCR.

Now we begin to build the domain knowledge base for each device. The heuristic rules in the domain knowledge base are derived from the repair manual [16–18] and some experts, and will be used to hypothesise the possible faults for that device. For instance, the "brake" component in the knowledge base consists of the following heuristic rules:

RULE: Brake 1
IF the video problem is from VCR itself;
AND the pad in reel brake is worn off;
AND the audio and video connection is in good condition;
THEN jammed take-up brake (confidence factor = 0.5).
RULE: Brake 2
IF the video problem is from VCR itself;
AND the pad in reel brake is worn off;
AND the loading gears run irregularly;
AND some obstruction is found near record head;
THEN jammed take-up brake (confidence factor = 0.7).
•
•

4.4 Inference Engine

Use the previous meta knowledge base and Fig. 3 as an example. Based on rule 1 of the meta knowledge base, the inference engine first examines the device "track control" by consulting the corresponding domain knowledge base. In this case, rule track control 1, rule track control 2, etc., are investigated. If those heuristic rules support enough evidence to prove that the device "track control" is at fault, the diagnostic process will go down to the next level. The inference engine examines the device "VCR board" by consulting rule VCR board 1, rule VCR board 2, etc. If there is not enough evidence to prove that device "VCR board" is at fault, the inference engine will check rule 6 in the meta knowledge base and test the device "tape". However, if the inference engine does not have evidence to support the device "track control" being at fault, the inference engine will check rule 7 in the meta knowledge base and examine the device "tape control" by consulting its domain knowledge base. This examine-and-verify process lasts until the smallest responsible device is found.

In order to validate the system performance of EMFDES, an expert system for VCR troubleshooting, VTS, is developed first. The knowledge base for VTS is directly extracted from manuals [16–18] and two experts with 15 years' experience in VCR repair. There are 156 rules in its knowledge base. The inference engine of VTS checks all heuristic rules in the knowledge base, based on a backward chaining search. VTS has been validated by actual failure cases and performs well (100 actual faults in a maintenance department of a SAMPO company were obtained from 1999 to 2000. The accuracy of VTS is 80%. The accuracy of experts is 76%.).

One hundred faults in a maintenance department of a SAMPO company from 1999 to 2000 were selected for comparing the performance of both systems. These faults involve three major symptoms: not thread tape; not play tape; and snowy picture. (Since these three symptoms happen frequently [16–18], this work will focus on these symptoms.) Two factors

Table 4. The query number required for EMFDES and VTS via 100 cases.

Symptom	Number of cases	EMFDES	VTS	Query number reduced	Improvement (%)
Not thread tape	56	338	374	36	10
Not play tape	24	142	166	24	14
Snowy picture	20	118	136	18	13
Total	100	598	676	78	12

are considered in comparing the performance of both systems. The first factor is the accuracy of the diagnosis result. Since the knowledge base of EMFDES is the same as VTS, the accuracy of the diagnosis should be the same. The later validation experiment also indicates that both systems reach the same conclusions by testing the sample cases. The second considered factor is the efficiency of diagnosis. This can be evaluated by using the number of queries. The more questions a system asks, the more time the user requests. That is, the user may need more time to collect relevant information, to conduct some tests, or to observe some device behaviour for answering these questions. Therefore, the number of queries is used to measure the diagnosis efficiency. Table 4 displays the query number, the reduced query number, and the improvement percentage using 100 sample cases. In this table, the improvement for symptom “not play tape” is the highest of these three symptoms and the improvement is about 14%. In addition, this table shows that EMFDES can normally reduce by 12% the number of queries required for VTS.

5. Conclusions

This paper proposes an efficient machine fault diagnosis expert system via a fuzzy group multiple attribute decision making method. Unlike many existing diagnostic expert systems which must check all rules in the knowledge base, EMFDES conducts an efficient search to find the possible causes. This expert system has been validated by a VCR troubleshooting sample. EMFDES can be implemented in the service shops and factories for troubleshooting machine failures such as PC equipment troubleshooting.

Although this system has been designed and developed successfully, several further research directions can be followed to enhance EMFDES:

1. The system may generate some questions that seem trivial to an experienced user. An algorithm that can detect different degrees of user background is recommended.
2. This system can only detect single faults. Diagnosis of multiple faults may be considered.

References

1. P. Burrell and D. Inman, “An expert system for the analysis of faults in an electricity supply network: problems and achievements”, Computers in Industry, 37, pp. 113–123, 1998.
2. J. Durkin, Expert Systems: Design and Development, Macmillan, 1994.
3. J. H. Bradley and R. D. Hauser, “A framework for expert system implementation”, Expert Systems with Applications, pp. 157–167, 1995.
4. J. Liebowitz, “Expert systems: a short introduction”, Engineering Fracture Mechanics, 50, pp. 601–607, 1995.
5. M. Krishnamurthi and D. T. Phillips, “An expert system framework for machine fault diagnosis”, Computers and Industrial Engineering, 22, pp. 67–84, 1992.
6. T. K. Ma, C. C. Liu, M. S. Tsai and R. Rogers, “Operational experience and maintenance of an on-line expert system for customer restoration and fault testing”, IEEE Transactions on Power Systems, 7, pp. 835–842, 1992.
7. S. C. Liu and S. Y. Liu, “An efficient expert system for air compressor troubleshooting”, Expert Systems, 18, pp. 203–214, 2001.
8. S. Vranes, M. Stanojevic and V. Stevanovic, “BEST-based expert diagnostic system for the aluminum industry”, Computers in Industry, 32, pp. 53–68, 1996.
9. S. J. Chen, and C. L. Hwang, Fuzzy Multiple Attribute Decision Making-Methods and Applications, Springer-Verlag, 1992.
10. C. C. Hon, Y. Y. Guh, K. M. Wang and E. S. Lee, “Fuzzy multiple attribute and multiple hierarchical decision making”, Computers with Mathematical Applications, 32, pp. 109–119, 1996.
11. C. L. Hwang and M. J. Lin, Group Decision Making under Multiple Criteria, Springer-Verlag, 1986.
12. R. A. Ribeiro, “Fuzzy multiple attribute decision making: a review and new preference elicitation techniques”, Fuzzy Sets and Systems, 78, pp. 155–181, 1996.
13. T. J. Ross, Fuzzy Logic with Engineering Applications, McGraw-Hill, 1997.
14. S. J. Russel and P. Norvig, Artificial Intelligence, Prentice-Hall, 1992.
15. H. G. Chong and W. J. Walley, “Rule-based versus probabilistic approaches to the diagnosis of faults in wastewater treatment processes”, Artificial Intelligence in Engineering, 1, pp. 265–273, 1996.
16. G. McComb, Troubleshooting and Repairing VCRS, TAB Books, New York, 1991.
17. R. L. Goodman, Maintaining and Repairing VCRS, TAB Books, Blue Ridge Summit, 1993.
18. J. D. Lend, Complete Guide to Modern VCR Troubleshooting and Repair, Prentice-Hall, 1985.