A Construction of Fuzzy Inference Network based on Neural Logic Network and its Search Strategy

Mal-rey Lee
Dept. of Multimedia Information & System, School of Multimedia,
Yosu National University,
San 96-1 Dunduckdong, Yosu JunNam, KOREA 550-749
e-mail:mrlee@yosu.yosu.ac.kr
Fax:81-662-659-3440

ABSTRACT

Fuzzy logic ignores some information in the reasoning process. Neural networks are powerful tools for the pattern processing, but, not appropriate for the logical reasoning. To model human knowledge, besides pattern processing capability, the logical reasoning capability is equally important. Another new neural network called neural logic network is able to do the logical reasoning. Because the fuzzy inference is a fuzzy logical reasoning, we construct fuzzy inference network based on the neural logic network, extending the existing rule- inference network. And the traditional propagation rule is modified. For the search strategies to find out the belief value of a conclusion in the fuzzy inference network, we conduct a simulation to evaluate the search costs for searching sequentially and searching by means of search priorities.

Keywords: Neural Logic Network, Propagation Rule, Fuzzy Inference Network

1. Introduction

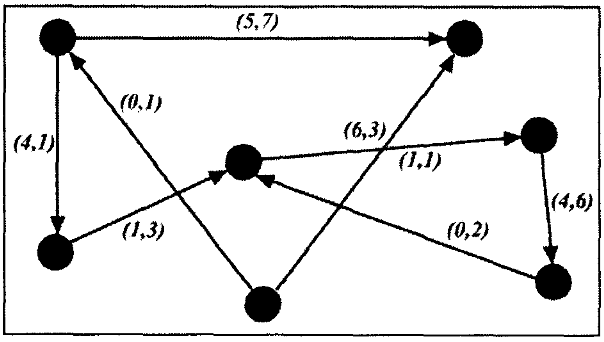
Expert system has been the most successful one among the application systems based on the research results in the artificial intelligence area. The main problem area in the expert system is knowledge acquisition, i.e., it is very time-consuming to acquire experts' knowledge. It is also very difficult to transform all the acquired knowledge to the rule of "IF··· THEN" form. [1,5,7,11] In order to resolve these problems machine learning is widely being studied for developing the self- learning system. Neural network and genetic algorithm are in the category of machine learning. Therefore, a hybrid system, which combines the expert system with genetic algorithm and neural network is considered as the new methodology for the next generation artificial intelligence system [1,2,4,5]. Regarding this Quinlan [13] proposed an automatic rule generation method using decision tree. Gallant established a basis for coupling neural network and expert system [5,11]. According to Gallant, knowledge base is constructed by using neutral network and the knowledge acquisition problem can be solved by learning knowledge base using effective learning algorithms such as pocket algorithm [5]. However, the logical reasoning capability as well as the pattern processing capability is required to model human knowledge. In this reason there has been a research activity which enables the logical reasoning by using reasoning network constructed by neural logic network [1] which is a modified neural network.

On the other hand the expert system which introduces fuzzy logic in order to treat uncertainties in called fuzzy expert system. The fuzzy expert system, however, has a potential problem which may lead to inappropriate results due to the ignorance of some information by applying fuzzy logic in reasoning process in addition to the knowledge acquisition problem. In order to overcome these problems, We construct fuzzy inference network by extending the concept of reasoning network in this paper. In the fuzzy inference network, the propositions which form fuzzy rules are represented by nodes. And these nodes have the truth values representing the belief values of each proposition. The logic operators between propositions of rules are represented by links. And in this paper the propagation rule [2] used in the existing rule-inference network is modified and then applied.

In the fuzzy inference network, to determine the truth values of propositions in the execution part of fuzzy rules, the nodes linked with the propositions are to be searched for . In order to do this , we conduct a simulation to evaluate the search costs for searching sequentially all the nodes along the links and searching by means of priorities given to the links.

2. Neural Logic Network

Neural logic network is a basis for the effective modeling of three-valued boolean logic using the existing neural network and may be extended further to perform probabilistic or fuzzy logic. Whole the existing boolean logic was developed based on the two values, "TRUE" and "FALSE", the three-valued boolean logic additionally includes "UNKNOWN" , [1,2,4]. Neural logic network is represented by finite and directive graphs using nodes and links. Sequence pair (x, y) which corresponds to weighting factor is assigned to every link. Fig.1 shows an example of neural logic network.



[Fig 1] Neural Logic Network

All the nodes have one of the following three activation values

(1,o)	for	" TRUE "
(0,1)	for	" FALSE "
(1,1)	for	" UNKNOWN "

The following propagation rule is used to determine the activation value of P node in Fig.2 [1]. The nodes linked with node P are $\{Q_1, Q_2, \cdots, Q_n\}$, Node value of Q_i is (a_i,b_i) and the weighting value of a link between P and Q_i is (x_i, y_i).

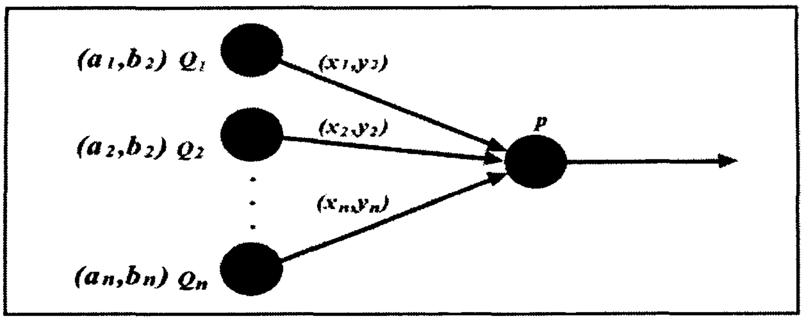
Step 1 : Compute $\alpha = \Sigma a_i,x_i$ and $\beta = \Sigma b_i,y_i$

Step 2 : Activation value of node P is calculated as

(1,0) , for $\alpha - \beta \geq 1$

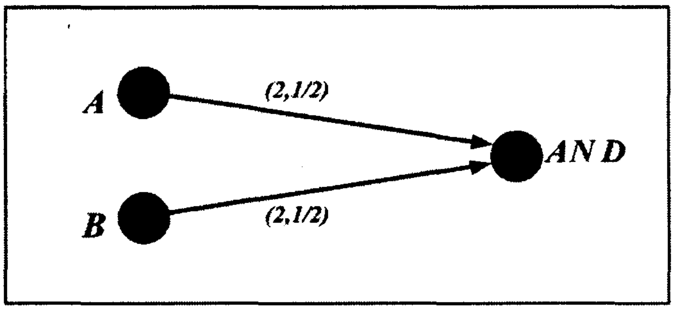
(0,1) , for $\alpha - \beta \leq -1$

(0,0) , otherwise.

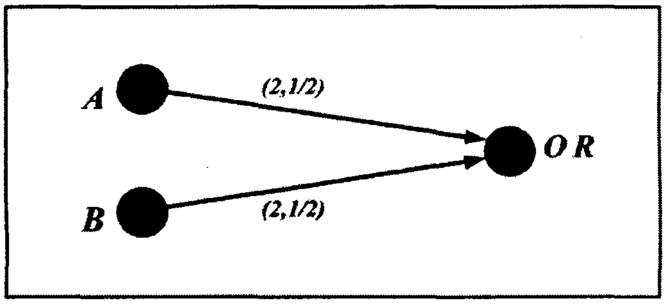


[Fig 2] propagation rule for Neural Logic Network

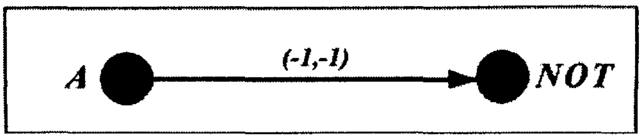
Using the neural logic network , logic operation can be represented. Figs 3 to 6 show "AND", "OR", "NOT", and "IF···THEN" operations for three valued logic.



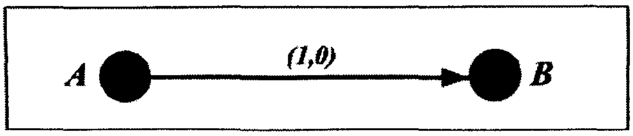
[Fig 3] A AND B



[Fig 4] A OR B

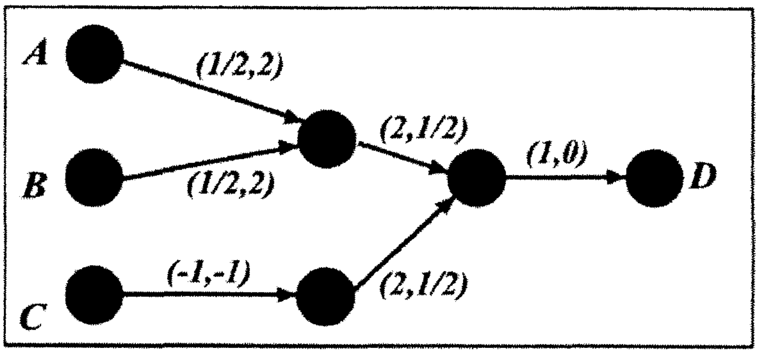


[Fig 5] NOT A



[Fig 6] IF A THEN B

Using the above definitions of logic operations, we can easily represent the rules including arbitrary logic operators by neural logic network. An example is shown below.
IF (A AND B) OR (NOT C) THEN D



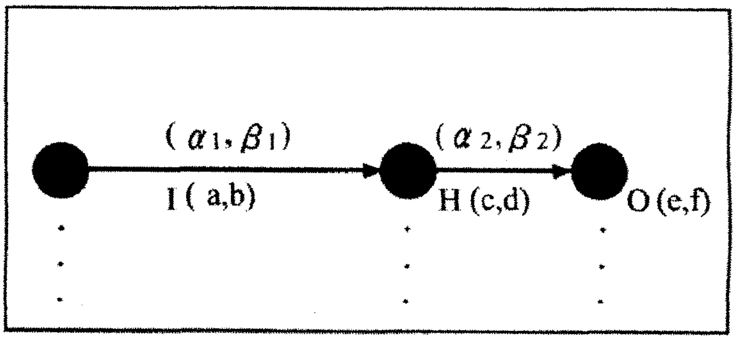
[Fig.7] Representation of a rule by Neural Logic Network

Each node in the network corresponds to a proposition or a logic operator in the rule and the reasoning process can be represented by neural logic network consisting of nodes and links.

3. Construction of Fuzzy Inference Network

3.1 Construction of Nodes

Fuzzy inference network consists of input node (I), hidden node (H) and output node (O) as shown in Fig.8 [2].



[Fig 8] Construction of fuzzy Inference Network

Each input node corresponds to a proposition in the condition part under the fuzzy rule. Node values for the input nodes are given by knowledge engineer in the form of sequence pair (a, b) which consists of non-negative real numbers. "a" and "b" are quantitative expectations for the proposition related to the node to be "TRUE" and "FALSE", respectively. For instance, if 70 and 20 out of 100 experts replied "TRUE" and "FALSE" to a proposition, respectively, and 10 experts did not know the answer, (a, b) becomes (0.7, 0.2).

Hidden node represents a logic operation and its node values are (c, d). These node values are non-negative real numbers and determined by the propagation rule to be discussed in section 3.4.

Each output node represents execution part of a fuzzy rule. The node values (e, f) are non-negative real numbers and determined by input and output node values and weighting factors between nodes. There output node values e and f are the basis to decide whether the corresponding execution part in adopted or not. IF "l-e-f" is large, the decision can not be node.

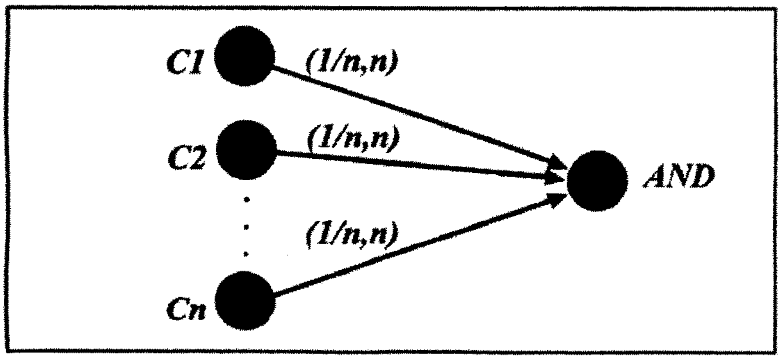
Nodes are connected by links and weighting factors (α, β) are given to links. α and β may be zero, negative or positive real numbers depending on the logic operation and the number of propositions in the conditional part of the fuzzy rule. The method to determine weighting factors in described in the next section.

3.2 Link and weighting factor

In the fuzzy inference network, nodes are connected by links and the weighting factors given to links are determined by the logic operations and the number of input nodes.

(1) "AND" operator

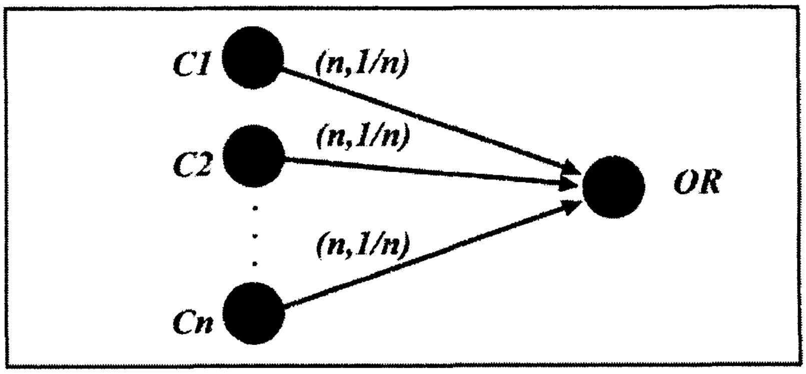
In case that the conditional part of the fuzzy rule is "C_1, AND C_2 AND … AND C_n " , the weighting factor (α , β) is (1/n,n) as shown in Fig.9



[Fig9] "AND" operator and weighting factor

(2) "OR" operator

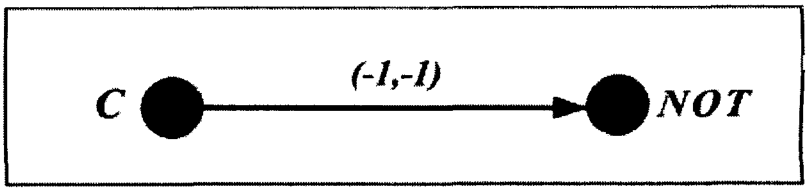
In case that the conditional part of the fuzzy rule is "C_1 OR C_2 OR … OR C_n , the weighting factor (α , β) is (n,1/n,) as shown in Fig.10.



[Fig.10] "OR" operator and weighting factor

(3)"NOT" operator

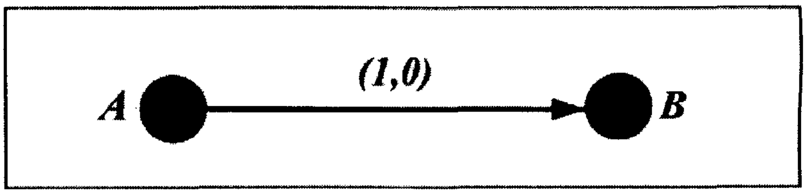
For "NOT" operator, the weighting factor (α , β) is (-1,-1) as shown in Fig.11.



[Fig.11] "NOT" operator and weighting factor

(4)"IF…THEN" operator

For "IF…THEN.." operator, the weighting factor (α , β) is (1,0) as shown in Fig.12.

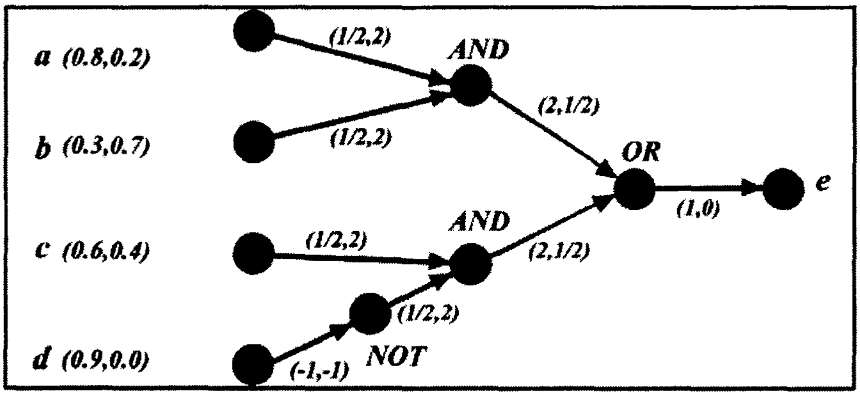


[Fig.12] "IF…THEN" operator and weighting factor

3.3 An Example of Fuzzy Inference Network

Any kind of fuzzy rules in the form of "IF…THEN" can be represented by the method of fuzzy inference network explained in sections 3.1 and 3.2. For example, the fuzzy rules below can be represented by fuzzy inference network in Fig.13.

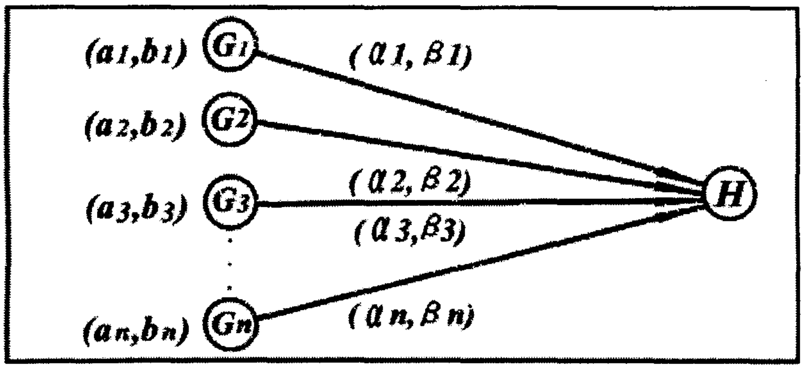
IF a(0.8,0.2) AND b(0.3,0.7) THEN e
IF c(0.6,0.4) AND NOT d(0.9,0.1) THEN e



[Fig.13] An Example of Fuzzy Inference Network

3.4 Propagation Rule

In a fuzzy inference network, propagation rules are needed to determine node values which are propagated from one node to another node. In this paper the propagation rules used in the existing inference networks are modified and then applied to the fuzzy inference network. The propagation rule to determine the node value of an arbitrary node H in Fig.14 is described below.



[Fig.14] Propagation of Node Values in Fuzzy Inference Network

(Step1) For input nodes, unknown , true and false are calculated as follows:

$$\text{unknown} = \sum_{1 \le i \le n} 1 - a_i - b_i$$

$$\text{true} = \sum_{1 \le i \le n} a_i \alpha$$

$$\text{false} = \sum_{1 \le i \le n} b_i \beta$$

(Step2) Node value (a_h, b_h) of node H is calculated as follows :

a_h = true / (true +false –unknown)
b_h= false / (true +false –unknown).

(Step3) Critical value θ_h is determined as follows:

$\theta_h = (I_{max} + I_{min})/2$

where $I_{max} = (\bigvee_{1 \le i \le n} a_i) \vee (\bigvee_{1 \le i \le n} b_i)$

$I_{min} = (\bigwedge_{1 \le i \le n} a_i) \wedge (\bigwedge_{1 \le i \le n} b_i)$

(Step4) IF a_h, - $b_h \ge Q_h$ THEN a_h=1, b_h=0
ELSE a_h and b_h remain the same value in step2.

4. Search Strategy in Fuzzy Inference Network.

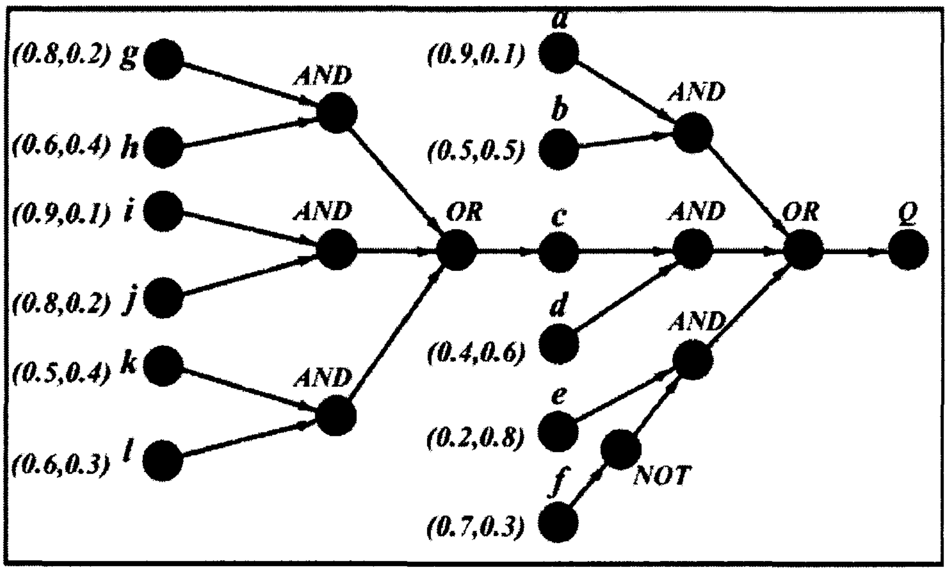
4.1 Sequential Search

By sequential search, in order to determine the belief value of the final proposition in a fuzzy inference network, all nodes connected with the proposition node are searched sequentially. For example, assume that we need to determine the belief value of a proposition Q when the following fuzzy rules are given:

IF a (0.9,0.1) AND b (0.5,0.5) THEN Q
IF c (0.7,0.2) AND d (0.4,0.6) THEN Q
IF e (0.2,0.8) AND NOT f (0.7,0.3) THEN Q
IF g (0.8,0.2) AND h (0.6,0.4) THEN C
IF i (0.9,0.1) AND j (0.8,0.2) THEN C

IF k (0.5,0.4) AND l (0.6,0.3) THEN C

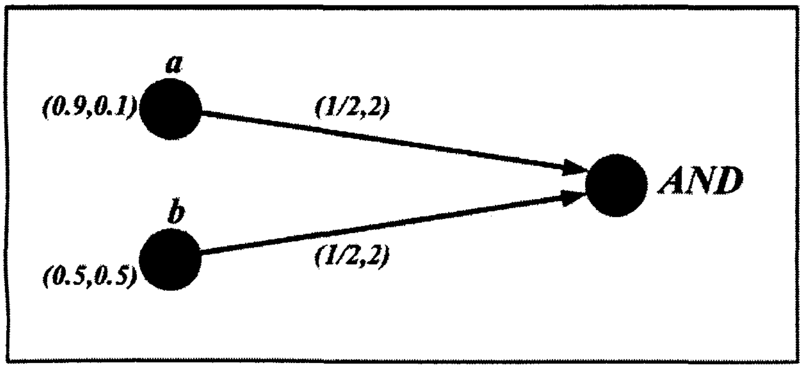
Above fuzzy rules construct the fuzzy inference network in Fig. 15.



[Fig.15] Construction of Fuzzy Inference Network

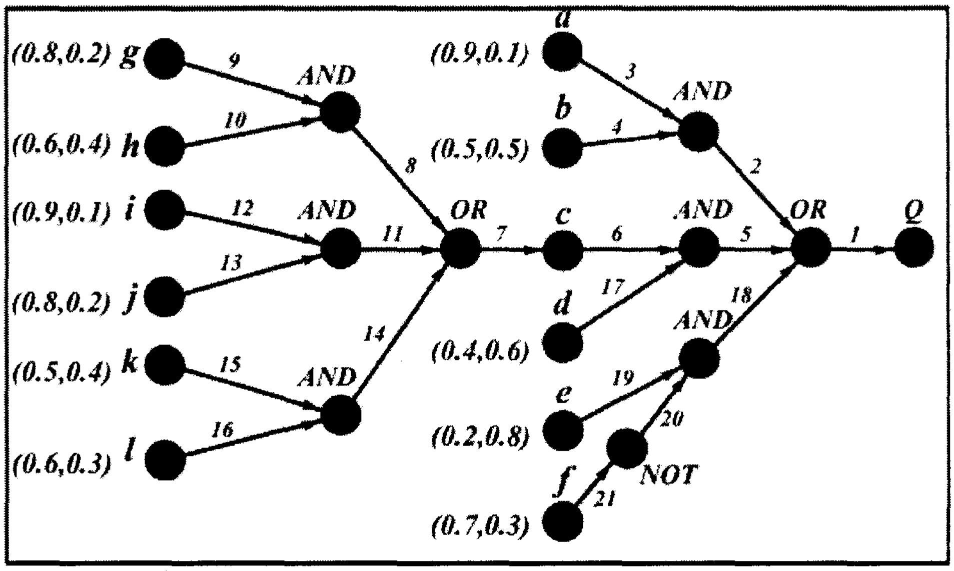
To determine the belief value of node Q, it is necessary to know the belief values of input nodes connected with node Q. The input node of node Q is "OR" operator node. The belief value of the "OR" node is again determined by the input nodes connected with the "OR" node. Three "AND" nodes are connected with the "OR" node and sequential searching is performed. If the belief value of a "AND" node is determined to be true, searching is no more performed. Then the belief values of "OR" and Q nodes become (1,0) which means true.

When the searching is performed for the first "AND" node, this node has two proposition nodes a and b. Using the fuzzy inference network shown Fig.16, the belief value of the first "AND" node can be determined by the propagation rule in section 3.4.



[Fig.16] "AND" node in Fuzzy Inference Network

Following the same procedure , the other two "AND" nodes are searched. When belief values of the three "AND" nodes are determined, belief values of Q node and "OR" node connected with the Q node are determined, and then the searching for the fuzzy inference network is finished.



[Fig. 17] shows the fuzzy inference network with the searching sequence numbers.

4.2 Searching by Priorities

In searching by means of search priorities, search priority numbers are assigned to each link and then the searching is performed by the priorities. The method to decide priorities depends on logic operators of the connected nodes.

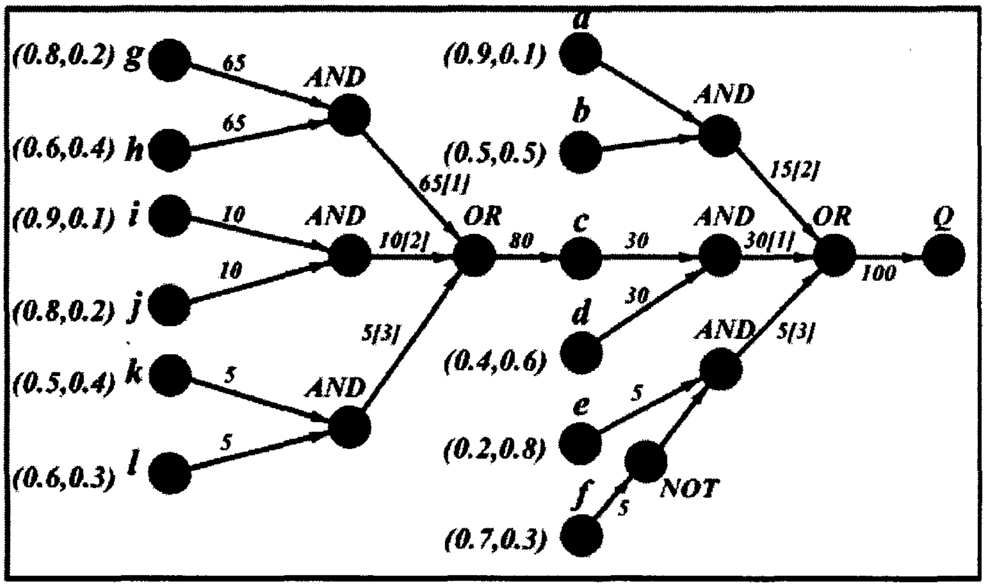
(1) "OR" node
Higher priority is given to the link connected with the node which contributes more to make the belief value of the node true.

(2) "AND" node
High priority is given to the link connected with the node which contributes more to make the belief value of the node false.

The reason of the above "high priority" policy is that, in case of "OR" ["AND"] node, when a node out of the input nodes is true [false],

searching for the other nodes is no more necessary.
For example, assume that searching of a fuzzy inference network is performed 100 times to determine the belief value of node Q.
There are 3 "AND" nodes as input nodes of "OR" node which is connected with node Q. Assume that the belief value of node Q is determined by the belief value of the first [second, third] node 15[80,5] times out of 100 times. Then the second " AND" node will have the highest priority because the probability for second "AND" node to determine the belief value of node Q is the highest.

The first " AND" node will have the next priority. Therefore, the third "AND" node has the lowest priority. Fig.18 shows an example of searching by priorities in which priorities are assigned to "OR" nodes to determine the belief value of node Q.

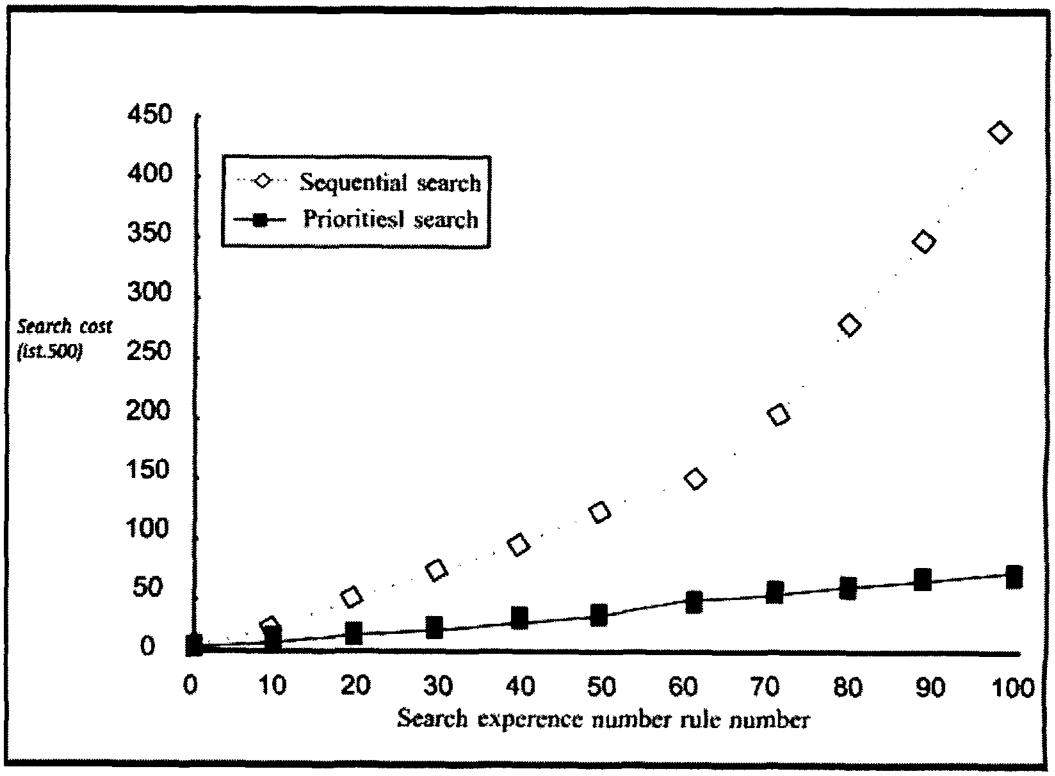


[Fig.18] Searching by priorities in Fuzzy Inference Network

The belief values (α, β) given to proposition nodes in the fuzzy inference network can be varied dynamically by knowledge engineers. However, the priorities given to connection links may be changed as the belief values of proposition nodes vary because the priorities are determined by searching results which depends on the belief values of input nodes. Thus ,the most appropriate searching strategy is taken under a given environment as the searching is performed by dynamically adjustable searching priorities.

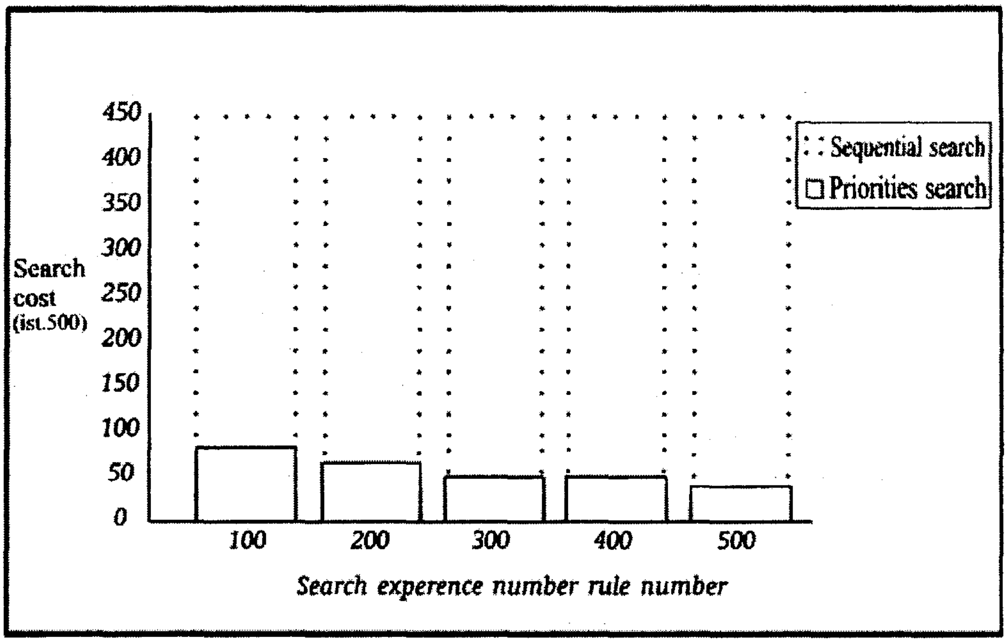
5. Experiments

The criteria for evaluation of experiments in this paper is IST (inference network search time for a node, unit: second) required to determine the belief value of a proposition node in the fuzzy inference network.
Fig. 19 shows IST for sequential searching and searching by priorities as the number of rules in the fuzzy inference network. The difference in IST between the two methods is larger as the number of rules increases. Therefore, searching by priorities is more effective as the size of fuzzy inference network increases.



[Fig.19] comparison of IST versus number of rules.

[Fig.20] compares IST for sequential searching and searching by priorities as the number of searching increases when the number of rules is fixed as 100. While IST for sequential searching is constant, IST for searching by priorities decreases as the number of searching increases.
This means that the efficiency of searching by priorities increases as the number of searching increases.



[Fig.20] Comparison of IST versus number of searching

When the amount of knowledge consisting of a fuzzy inference network is large or when the knowledge database continuously expands through collection of knowledge, the IST may be an important factor for the performance of the entire system.

The experimental results in this chapter show that searching by priorities is more efficient than sequential searching in the fuzzy inference network.

6. Results

The fuzzy inference network is constructed by extending the concept of the existing inference network based on neural logic network. In the fuzzy inference network, the propositions consisting of fuzzy rules are represented using nodes. These nodes have belief value of each proposition.

The logic operators between propositions are represented using links. The traditional propagation rule is modified and applied in this paper.

Experiments are performed to compare search costs by sequential searching and searching by priorities. The experimental results show that the searching by priorities is more efficient than the sequential searching as the size of the fuzzy inference network becomes larger and an the number of searching increases.

However, the searching by priorities depends on the previously-collected searching experiences, therefore, when the fuzzy inference network is modified, it needs to obtain some new searching experiences to find out the most appropriate searching paths. More studies on this feature will be performed in the future.

Reference

[1] B. T. Low, H.C. Lui, A. H. Tan, and H. H. Teh, "*Connectionist Expert System with Adaptive Learning Capability*," IEEE Transaction on Knowledge and Data Engineering, Vol. 3, No. 2, June, pp. 200-207, (1991).

[2] L. S. Hsu, H. H. The, S. C. Chan and K. F. Loe, "*Fuzzy Logic in Connectionist Expert Systems*", IJCNN'90, Vol. 2, pp. 599-602, (1990).

[3] Tatsuki Watanabe, Masayuki Matsumoto and Takahiro Hasegawa, "*A Layered Neural Network Model using Logic Neurons*," in Proc. Of the International Conf. On Fuzzy Logic & Neural Networks, pp. 675-678, (1990).

[4] Wang-Pei Zhuang, Wu Zhi Qiao and The Hoon Heng, "*The Truth-Valued Flow Inference Network*," in Proc. Of the International Conf. On Fuzzy Logic & Neural Networks, pp. 267-281, (1990).

[5] Stephen I. Gallant, "*Connectionist Expert Systems*," Comm. Of the ACM, Vol.31, pp. 152-169, (1992).

[6] Ricardo Jose Machado and Armando Freitas da Rocha, "*Fuzzy connectionist Expert System*", in Proc. of IEEE International Conf. on Neural Networks, Vol. 3, pp. 1571-1576, (1994).

[7] Elie Sanchez, "*Fuzzy Connectionist Expert System*," in Proc. of the International Conf. on Fuzzy Logic & Neural Networks, pp. 31-35, (1990).

[8] Stephen I. Gallant, "*Neural Network Learning and Expert System*," The MIT press, pp. 255-293, (1993).

[9] Shashi Shekhar and Minesh B. Amin, "*Generalization by Neural Networks*," IEEE Transactions on Knowledge and Data Engineering, Vol. 4, No., 2, pp. 28-38, (1992).

[10] Adam Blum, "*Neural Networks in C++*," Wiley, (1992).

[11] K. Hirota and W. Pedrycz, "*Fuzzy Logic Neural Networks: Design and Computations*," IJCNN, pp. 152-157, (1991).

[12] Zadeh L. A., "*The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems*", Fuzzy Sets and Systems, Vol. 11, pp. 199-227, (1983).

[13] J. R. Quilan, "*Generating production rules from decision trees*,", in Proc. IJCAI, pp. 304-307, (1987).