# Algorithms for Increasing of the Effectiveness of the Making Decisions by Intelligent Fuzzy Systems

# Olga Dolinina, Aleksandr Shvarts

Yury Gagarin State Technical University of Saratov, Saratov, Russia

**Keywords:** intelligent systems, fuzzy inference, increasing of the effectiveness of the making decisions, grouping of the rules and variables, building of the rules' dependence.

**Abstract.** There are considered rule-based intelligent systems using fuzzy inference. Comparative analysis of different approaches and algorithms of making decisions on the base of fuzzy logic is given. Using of the parallel calculations can reduce the time of making decision in case of large-scale systems. Effectiveness of parallel calculations depend on the grouping of the rules and variables. Building of the graph of the dependence of the rules and the graph of dependence of the linguistic variables are suggested. On the base of the developed groups of rules and defuzzification of the linguistic variables we suggest to reduce the time of making decision and therefore to increase the effectiveness of the decision making with using of parallel calculations for each group.

## Introduction

Intelligent systems are one of the most rapidly developing branches of science and technologies. In the majority of cases expert is not able to describe knowledge accurately or express the causation between them. The most popular method to design systems with such uncertainties was introduced by Zadeh [1,2]. At the same time large-scale knowledge bases demand special methods of knowledge processing, for example, using parallel calculations [3] effectiveness of which depend on the correct grouping or formation of clusters of rules.

# **Fuzzy Intelligent Systems**

Fuzzy rule base is presented in the form of

with V – set of linguistic variables of the system,

*I* – set of input linguistic variables,  $I \subset V$ ; *G* – set of output linguistic variables,  $G \subset V$ ; *P* – set of intermediate linguistic variables,  $P \subset V = I \cup G \cup F$ 

P - set of intermediate linguistic variables,  $P \subset V$ ,  $I \cup G \cup P = V$ ,  $I \cap G = \emptyset$ ,

 $I \cap P = \emptyset, P \cap G = \emptyset;$ 

R – set of fuzzy rules

$$< r_{j,} u_{j} > \text{IF } v_{j,1} \text{ is } t_{j,1} AND \dots AND \ v_{j,n_{j}-1} \text{ is } t_{j,n-1}$$

$$THEN v_{j,n_{j}} \text{ is } t_{j,n_{j}}$$
(2)

(1)

with  $r_i$  – unique rule name;

 $u_i$  – linguistic certainty degree of the  $r_i$  rule;

 $n_i$  – number of linguistic variables in the  $r_i$  rule;

 $v_{j,1}, \dots, v_{j,n_i-1}$  – premises (input variables) for the  $r_j$  rule;

 $t_{j,1}, \dots t_{j,n_i-1}$  - values of input linguistic variables (terms) for the  $r_j$  rule;

 $v_{j,n_i}$  – output linguistic variable (goal) for the  $r_j$  rule;

 $t_{j,n_i}$  – value of output linguistic variable for the  $r_j$  rule.

Fuzzy set is A is presented in the following form

$$A = \left\{ \left( x, \mu_A(x) \right) \mid x \in X \right\}$$
(3)

with X – universal set;

x – element of X set;

 $\mu_A(x)$  – membership function.

Membership function  $\mu_A(x)$  describes the degree of membership of each  $x \in X$  to the fuzzy set A, with  $\mu: X \to [0,1]$ . Thereby,  $\mu_A(x)$  is 1, if element x is completely included in A and 0, if element x is not included in A. Membership function values on interval (0,1) describe fuzzy included elements.

Diagram of reasoning engine in fuzzy intelligent system (IS) is presented on Fig. 1.

A number of reasoning methods and algorithms have been introduced and discussed since the appearance of fuzzy sets theory. Each of algorithms [4-12] differs from the point of view of sequence and type of operators applied in fuzzyfication, fuzzy inference and defuzzyfication stages. They use also fuzzy rules of different form.



Fig. 1. Diagram of reasoning engine in fuzzy intellectual systems

Mamdani [4], Tsukamoto [5], product-sum-gravity [6] and Larsen [7] methods use fuzzy rules, which form is very close to the way of human thinking. Therefore, it would be easier for experts and knowledge engineers to design the rule base.

Tsukamoto inference method also requires only monotonous membership functions, what is unacceptable in many cases.

Sugeno [8], simplified fuzzy inference [9], singleton [10] and SIRMs [11] methods requires crisp number values or coefficients, which calculation is not a trivial task for some areas. However, those methods involve only simple algebraic operators, that lead to lower computational cost in comparison with the first group of methods.

SIRMs fuzzy systems contains separate groups of rules (modules) and each of the rules must have the only one premise.

Most of the modern intelligent systems using fuzzy inference have rule bases of high dimension (thousands of rules). This causes the need to use special methods of inference management in order to improve efficiency and minimize calculation time. Existing methods of fuzzy inference allows parallel execution on a number of computational nodes, what can reduce time spent on reasoning process. This paper introduces algorithms for improving efficiency of fuzzy inference in rule-based intelligent systems by grouping of the rules.

In fuzzy rule bases there is a wide range of cases, when it is impossible to process one rule without processing some other rules before. This problem appears, if the rule's premises contains variables, which are not received from the input of the whole system, but calculated after different rules are processed.

New algorithms which allow to define the order of the processing of the rules in the knowledge base by preliminary grouping of the rules, have been introduced to solve this problem.

#### **Exploring Dependencies in the Rule Base**

**Rules Dependencies.** The first stage is to generate the matrix of rules dependencies. Following algorithm is used:

**Step 1.** Initialize zero-filled matrix *DR* with dimension of  $|R| \times |R|$  (|R| – cardinal number of *R* set).

**Step 2.** For each  $r_i \in R$  execute Step 3.

**Step 3.** For each  $r_q \in R$ ,  $r_q \neq r_j$  execute Step 4.

Step 4. If  $v_{j,n_i} \in G_q$ , then  $DR_{j,q} = 1$  and  $DR_{q,j} = -1$ .

*DR* is an adjacency matrix of the directed graph of rules dependencies. This graph can be used to define each rule predecessor rules.

Each rule  $r_j \in R$  is mapped to a value  $\rho(r_j)$ , which represents sequence number of a group of processed rules. This sequence number means that rule  $r_{j_1}$  cannot be processed before  $r_{j_2}$ , if

 $\rho(r_{j_2}) < \rho(r_{j_1}).$ 

#### 3.1 Calculating the Sequence Number of a Group of Processed Rules

**Step 1.** Initialize zero-filled set RO, |RO| = |R|. **Step 2.** j = 1. **Step 3**. If  $j \leq |\mathbf{R}|$ , then go to Step 4, else go to Step 10. **Step 4.** *i* = 1. **Step 5.** If  $i \leq |R|$ , then go to Step 6, Step 8. **Step 6.** If  $DR_{i,i} = -1$ , then go to Step 9. **Step 7.** i = i + 1. **Step 8.**  $ro_i = 1$ . **Step 9.** j = j + 1. Go to Step 3. **Step 10.** j = 1, c = 0. **Step 11.** If  $j \le |R|$ , then go to Step 12, else go to Step 20. **Step 12.** i = 1, m = 1. **Step 13.** If  $i \leq |R|$ , then go to Step 14, else go to Step 17. **Step 14.** If  $DR_{i,i} = -1$ , then go to Step 15, else go to Step 16. **Step 15.** If  $ro_i > m$ , to  $m = ro_i$ . **Step 16.** i = i + 1. Go to Step 13. **Step 17.** If  $ro_i < m+1$ , to c = 1. **Step 18.**  $ro_i = m+1$ . **Step 19.** j = j + 1. Go to Step 11. **Step 20.** If c = 0, then go to Step 21, else go to Step 10. **Step 21.** For each  $r_i : \rho(r_i) = ro_i$ . Using calculated values of  $\rho(r_i)$  following sets are formed  $RG_m: r_i \in RG_m \Leftrightarrow \rho(r_i) = m, m = \overline{1, \dots L_R}$ 

(4)

where  $L_R$  – total number of groups of processed rules.

#### **3.1 Linguistic Variables Dependencies**

Fuzzy inference involves not only calculation of membership functions for each rule  $r_j$ , but also defuzzyfication of rules' goals. Sequence of goal variables defuzzyfication is not simply obtained from rules dependencies graph, because, each variable can be used in the right part of many rules

with different value of  $\rho(r_j)$ 

So, it is important to discover dependencies between linguistic variables. Following algorithm is used to generate the matrix of dependencies:

**Step 1.** Initialize zero-filled matrix D with dimension of  $|V| \times |V|$ .

**Step 2.** For each  $v_k \in V$  execute Step 3.

**Step 3.** For each  $v_h \in V$ ,  $v_h \neq v_k$  execute Step 4.

**Step 4.** For each  $r_i \in R$  execute Step 5.

**Step 5.** If  $v_k \in G_j$  and  $v_h = v_{j,n_j}$ , then  $DV_{k,h} = 1$  and  $DV_{h,k} = -1$ .

DV is an adjacency matrix of directed graph of linguistic variables dependencies. This graph can be used to find each variable predecessors.

Each rule  $r_i \in R$  is mapped to a value  $\rho(r_i)$ , which represents sequence number of a group of processed rules. This sequence number means that rule  $r_{i_i}$  cannot be processed before  $r_{i_i}$ , if  $\rho(r_{i_1}) < \rho(r_{i_1}).$ 

Each linguistic variable  $v_k \in V$  is mapped to a value  $\rho(v_k)$ , which represents sequence number of a group of defuzzyfied variables. This sequence number means that variable  $v_{k}$  cannot be defuzzyfied before  $v_{k_2}$ , if  $\rho(v_{k_2}) < \rho(v_{k_1})$ .

## 3.2 Calculating the Sequence Number of a Group of Defuzzyfied Variables

**Step 1.** Initialize zero-filled set *VO*, |VO| = |V|. **Step 2.** k = 1. **Step 3**. If  $k \le |V|$ , then go to Step 4, else then go to Step 10. **Step 4.** h = 1. **Step 5.** If  $h \le |V|$ , then go to Step 6, else then go to Step 8. **Step 6.** If  $DV_{k,h} = -1$ , then go to Step 9. **Step 7.** h = h + 1. **Step 8.**  $vo_k = 1$ . **Step 9.** k = k + 1. Go to Step 3. **Step 10.** k = 1, c = 0. **Step 11.** If  $k \le |V|$ , then go to Step 12, else then go to Step 20. **Step 12.** h=1, m=1. **Step 13.** If  $h \le |V|$ , then go to Step 14, else then go to Step 17. **Step 14.** If  $DV_{k,h} = -1$ , then go to Step 15, else then go to Step 16. **Step 15.** If  $vo_h > m$ , to  $m = vo_h$ . **Step 16.** h = h + 1. Go to Step 13. **Step 17.** If  $vo_k < m+1$ , to c = 1.

**Step 18.**  $vo_k = m+1$ .

**Step 19.** k = k + 1. Go to Step 11.

**Step 20.** If c = 0, then go to Step 21, else then go to Step 10.

**Step 21.** For each  $v_k$ :  $\rho(v_k) = vo_k$ .

Using calculated values of  $\rho(v_k)$  following sets are formed

$$VG_m: v_k \in VG_m \Leftrightarrow \rho(v_k) = m, m = \overline{1, \dots L_V}$$
(5)

where  $L_v$  – total number of groups of defuzzyfied variables.

Any variable  $v_k : \rho(v_k) = 1$  is input variable ( $v_k \in I$ ) and there is no need to defuzzyfy it.

#### Conclusions

The suggested algorithms of formation of the groups of fuzzy rules for processing and defuzzyfication of the knowledge base can be used to reduce the full time of the making solution in fuzzy intelligent systems by applying of parallel calculations for each group of rules. Effective parallel processing is impossible without grouping the rules in advance, because transitive dependencies allow often only sequential processing. Effectiveness of the suggested algorithms have been proved by a set of the experimental tests generated with randomly chosen parameter values of the rules. Three types of the membership functions (Z-type, triangle, S-type) have been analyzed. The number of the rules differed from 1000 to 50000 in the experiment. The result got from the experiment showed that the processing time of the rules with using of the developed algorithms have been implemented successfully for the development of the expert system GAZDETECT [13] in which the knowledge base for detecting of the reasons of the faults consisted of the fuzzy rules. GAZDETECT is used for the faults detection in gas compressor plants for underground storage facilities of GAZPROM company.

## References

- [1] Zadeh, L. A. Fuzzy Sets. Information and Control, vol. 8, pp. 338–353, 1965.
- [2] Zadeh, L. A. The Concept Of A Linguistic Variable And Its Application To Approximate Reasoning. *Information Sciences*, vol. 8, pp. 199–249, 301–357; vol. 9, pp. 43–80, 1975.
- [3] Yingxia Shao; Bin Cui; Lin Ma. PAGE: A Partition Aware Engine for Parallel Graph Computation. *IEEE Transactions on Knowledge and Data Engineering*, vol.27, no.2, pp.518-530, 2015.
- [4] Mamdani E.H., Assilian S. An Experiment In Linguistic Synthesis With Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, vol. 7, No. 1, pp. 1–13, 1975.
- [5] Tsukamoto Y. An Approach To Fuzzy Reasoning Method. In: Gupta M.M., Ragade R.K. and Yager R.R. (eds.) Advances In Fuzzy Set Theory And Applications, pp. 137–149, 1979.
- [6] Seki H., Mizumoto M. On the Equivalence Conditions of Fuzzy Inference Methods Part 1: Basic Concept and Definition. *IEEE Transactions on Fuzzy Systems*, vol. 9, No. 6, pp. 1097– 1106, 2011.
- [7] Larsen P. M. Industrial Applications Of Fuzzy Logic Control. International Journal Of Man-Machine Studies, vol. 12, pp. 3–10, 1980.
- [8] Takagi T. Sugeno M., Fuzzy Identification Of Systems And Its Applications To Modeling And Control. IEEE Transactions On Systems, Man, And Cybernetics, vol. 15, No. 1, pp. 116–132, 1985.

- [9] Mizumoto M. Fuzzy controls under various fuzzy reasoning methods. *Information Science*, vol. 45, pp. 129–151, 1988.
- [10] Sugeno M. On stability of fuzzy systems expressed by fuzzy rules with singleton consequents. *IEEE Transactions on Fuzzy Systems*, vol. 7, No. 2, pp. 201–224, 1999.
- [11] Yubazaki N., Yi J., Hirota K. SIRMs (Single Input Rule Modules) connected fuzzy inference model. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 1, No. 1, pp. 23–30, 1997.
- [12] Jang J.-S. R. ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Transactions on Systems & Cybernetics, vol. 23, pp. 665–685, 1993.
- [13] Dolinina O.N., Antropov P.G., Kuzmin A.K., Shvarts A.Y. Using Intelligent Systems for fault detection in gas compressor plants. *Modern problems of science and education*, No.6, 2013 (in Russian), http://www.science-education.ru/113-11252.