SYNTHESIS AND ANALYSIS OF DECISION PROCESSES FOR THE AUTOMATION CONTROL SYSTEM WITH UNCERTAINTY. PART II.

Vladimír OLEJ
Jiří KŘUPKA
Department of Computer Science,
Technical University, 031 19 Liptovský Mikuláš, Slovak Republic

In this paper, the synthesis and analysis of decision processes of the automation control system with uncertainty are presented. This synthesis and analysis is realized by means of programming language fuzzy PROLOG and fuzzy neural networks.

1. INTRODUCTION

The synthesis and analysis of the automation control system (ACS) can also be realized on the basis of the programming language fuzzy PROLOG (FPROLOG). On the basis of the determined uncertainty features of means of combat the classifier of decision process of ACS based on the fuzzy neural networks (FNN) is defined. Classical and new modified genetic algorithms may be used for their learning.

2. SYNTHESIS AND ANALYSIS OF DECISION PROCESSES FOR ACS ON THE BASIS OF FPROLOG

The programming language FPROLOG can be used in the synthesis and analysis of decision processes of ACS. Our preliminaries are that decision rules are described with uncertainty.

The FPROLOG enables to process clauses (facts and rules) on the basis of generalized truth values. If the individual clauses are confident (they occur with the truth value 0 or 1), then they are processed by the classical programming language PROLOG. While the basis of the programming language is the resolution principle in the predicate first order logic [1], the basis of the programming language FPROLOG is the fuzzy resolution principle [2,3,4,5] in the FL [2,6,7]. In this section the analysis of the proposed results in programming systems is described. The process of assigning means of combat M_j to some targets T_i is described.

As an example the problem how to assign means of combat M_j to targets T_i is described. Means of combat are determined by given features $d_{i,j}$ and $P_{i,j}$. They are expressed with uncertainty and feature values range from 1 to 7 of kilometer scale. Membership functions of features are in Table 1.

Table 1. Membership functions of features

		Feature values						
		1	2	3	4	5	6	7
Uncertainty values	SM_d	1	0,75	0,25	0	0	0	0
	ME_d	0	0,25	0,75	1	0,75	0,25	0
	LA_d	0	0	0	0	0,25	0,75	1
	ZE_P	1	0,75	0,25	0	0	0	0
	MI_P	0	0,25	0,75	1	0,75	0,25	0
	GR_P	0	0	0	0	0,25	0,75	1

Suppose that it is necessary to assign means of combat M_j to targets T_i, where i=(1,2, 3), from the set of means of combat M_m.

For the first means of combat M_j let feature values be 2 and 4, for the second 7 and 7 and for the third 7 and 1, where SM_d = sm_d_i, ME_d = me_d_i, LA_d = la_d_i, ZE_P = ze_p_i, MI_P = mi_p_i a GR_P = gr_p_i. Then, the problem is how to find the most suitable means of combat M_j for uncertain feature values sm_d_i, me_d_i, la_d_i, ze_p_i, mi_p_i and gr_p_i. The program in the programming language FPROLOG can be designed as follows [2,8,9]:

```
10  choice('distance_parameter',Mj):-p1(Mj,Value1),p2(Mj,Value2),sm_d_1(Value1),ze_p_1(Value2),
    me_d_2(Value1),mi_p_2(Value2),la_d_3 (Value1),gr_p_3(Value2).
15  choice('parameter_distance',Mj):-p1(Mj,Value1),p2(Mj,Value2),ze_p_1(Value1),sm _d_1(Value2),
    mi_p_2(Value1),me_d_2(Value2),gr_p_3(Value1),la_d_3(Value2).
20  choice('distance_parameter',Mj):-p1(Mj,Value1),p2(Mj,Value2),sm_d_1(Value1),ze_p_1(Value2),
    la_d_2(Value1),mi_p_2(Value2),me_d_3 (Value1),gr_p_3(Value2).
25  choice('parameter_distance',Mj):-p1(Mj,Value1),p2(Mj,Value2),ze_p_1(Value1),sm_d_1(Value2),
    mi_p_2(Value1),la_d_2(Value2),gr_p_3 (Value1),me_d_3(Value2).
30  choice('distance_parameter',Mj):-p1(Mj,Value1),p2(Mj,Value2),me_d_1(Value1),ze_p_1(Value2),
    sm_d_2(Value1),mi_p_2(Value2),la_d_3 (Value1),gr_p_3(Value2).
35  choice('parameter_distance',Mj):-p1(Mj,Value1),p2(Mj,Value2),ze_p_1(Value1),me_d_1(Value2),
    mi_p_2(Value1),sm_d_2(Value2),gr_p_3 (Value1),la_d_3(Value2).
40  choice('distance_parameter',Mj):-p1(Mj,Value1),p2(Mj,Value2),la_d_1(Value1),ze_p_1(Value2),
    sm_d_2(Value1),mi_p_2(Value2),me_d_3 (Value1),gr_p_3(Value2).
45  choice('parameter_distance',Mj):-p1(Mj,Value1),p2(Mj,Value2),ze_p_1(Value1),la_d_1(Value2),
    mi_p_2(Value1),sm_d_2(Value2),gr_p_3 (Value1),me_d_3(Value2).
50  choice('distance_parameter',Mj):-p1(Mj,Value1),p2(Mj,Value2),me_d_1(Value1),ze_p_1(Value2),
    la_d_2(Value1),mi_p_2(Value2),sm_d_3 (Value1),gr_p_3(Value2).
```

```
55  choice('parameter_distance',Mj):-p1(Mj,Value1),p2(Mj,Value2),ze_p_1(Value1),me_d_1(Value2),
    mi_p_2(Value1),la_d_2(Value2),gr_p_3 (Value1),sm_d_3(Value2).
60  choice('distance_parameter',Mj):-p1(Mj,Value1),p2(Mj,Value2),la_d_1(Value1),ze_p_1(Value2),
    me_d_2(Value1),mi_p_2(Value2),sm_d_3 (Value1),gr_p_3(Value2).
65  choice('parameter_distance',Mj):-p1(Mj,Value1),p2(Mj,Value2),ze_p_1(Value1),la_d_1(Value2),
    mi_p_2(Value1),me_d_2(Value2),gr_p_3 (Value1),sm_d_3(Value2).
100 p1(Mj,Value)      {1/(1,2),1/(2,7),1/(3,7)}                                    repeat#3.
105 p2(Mj,Value)      {1/(1,4),1/(2,7),1/(3,1)}                                    repeat#3.
110 sm_d_1  (Value)  {1/(1),0.75/(2),0.25/(3),0/(4),0/(5),0/(6), 0/(7)}            repeat#7.
115 me_d_1  (Value)  {0/(1),0.25/(2),0.75/(3),1/(4),0.75/(5),0.25/(6),0/(7)}       repeat#7.
120 la_d_1  (Value)  {0/(1),0/(2),0/(3),0/(4),0.25/(5),0.75/(6),1/(7)}             repeat#7.
125 ze_p_1  (Value)  {1/(1),0.75/(2),0.25/(3),0/(4),0/(5),0/(6), 0/(7)}            repeat#7.
130 mi_p_1  (Value)  {0/(1),0.25/(2),0.75/(3),1/(4),0.75/(5),0.25/(6),0/(7)}       repeat#7.
135 gr_p_1  (Value)  {0/(1),0/(2),0/(3),0/(4),0.25/(5),0.75/(6),1/(7)}             repeat#7.
140 sm_d_2  (Value)  {1/(1),0.75/(2),0.25/(3),0/(4),0/(5),0/(6), 0/(7)}            repeat#7.
145 me_d_2  (Value)  {0/(1),0.25/(2),0.75/(3),1/(4),0.75/(5),0.25/(6),0/(7)}       repeat#7.
150 la_d_2  (Value)  {0/(1),0/(2),0/(3),0/(4),0.25/(5),0.75/(6),1/(7)}             repeat#7.
155 ze_p_2  (Value)  {1/(1),0.75/(2),0.25/(3),0/(4),0/(5),0/(6), 0/(7)}            repeat#7.
160 mi_p_2  (Value)  {0/(1),0.25/(2),0.75/(3),1/(4),0.75/(5),0.25/(6),0/(7)}       repeat#7.
165 gr_p_2  (Value)  {0/(1),0/(2),0/(3),0/(4),0.25/(5),0.75/(6),1/(7)}             repeat#7.
170 sm_d_3  (Value)  {1/(1),0.75/(2),0.25/(3),0/(4),0/(5),0/(6), 0/(7)}            repeat#7.
175 me_d_2  (Value)  {0/(1),0.25/(2),0.75/(3),1/(4),0.75/(5),0.25/(6),0/(7)}       repeat#7.
180 la_d_3  (Value)  {0/(1),0/(2),0/(3),0/(4),0.25/(5),0.75/(6),1/(7)}             repeat#7.
185 ze_p_3  (Value)  {1/(1),0.75/(2),0.25/(3),0/(4),0/(5),0/(6), 0/(7)}            repeat#7.
190 mi_p_3  (Value)  {0/(1),0.25/(2),0.75/(3),1/(4),0.75/(5),0.25/(6),0/(7)}       repeat #7.
195 gr_p_3  (Value)  {0/(1),0/(2),0/(3),0/(4),0.25/(5),0.75/(6),1/(7)}             repeat #7.
```

In the statements:

- 10 to 65 there are double inference rules of choice means of combat for individual combinations of features;
- 100 and 105 there are feature etalons;
- 110 to 195 there are membership function of feature values.

Results of the program:

```
?-choice(K,Mj).
(K='distance_parameter',   Mj=1)  {MT=0.2500,  T=0.0000  with  CONF=0.5000}
(K='distance_parameter',   Mj=2)  {MT=0.0000,  T=0.0000  with  CONF=1.0000}
(K='distance_parameter',   Mj=3)  {MT=0.0000,  T=0.0000  with  CONF=1.0000}
(K='distance_parameter',   Mj=1)  {MT=0.2500,  T=0.0000  with  CONF=0.5000}
(K='distance_parameter',   Mj=2)  {MT=0.0000,  T=0.0000  with  CONF=1.0000}
(K='distance_parameter',   Mj=3)  {MT=0.0000,  T=0.0000  with  CONF=1.0000}
```

Now, feature values are changed to values 7 and 1 for first, 3 and 4 for second, 7 and 1 for third means of combat M_j in statements 100 and 105:

```
100 p1(Mj,Value){1/(1,7),1/(2,3),1/(3,7)}repeat#3.
105 p2(Mj,Value){1/(1,1),1/(2,4),1/(3,1)}repeat#3.
```

The result of this change is as follows:

```
?-choice(K,Mj).
(K='distance_parameter',   Mj=1)  {MT=0.0000,  T=0.0000  with  CONF=1.0000}
(K='distance_parameter',   Mj=2)  {MT=0.2500,  T=0.0000  with  CONF=0.5000}
```

```
(K='distance_parameter',    Mj=3) {MT=0.0000,  T=0.0000  with  CONF=1.0000}
(K='distance_parameter',    Mj=1) {MT=0.0000,  T=0.0000  with  CONF=1.0000}
(K='distance_parameter',    Mj=2) {MT=0.2500,  T=0.0000  with  CONF=0.5000}
(K='distance_parameter',    Mj=3) {MT=0.0000,  T=0.0000  with  CONF=1.0000}
```

Now, feature values are changed to values 7 and 1 for first, 7 and 1 for second, 3 and 4 for third means of combat M_j in the statements 100 and 105:

```
100 p1(Mj,Value){1/(1,7),1/(2,7),1/(3,3)}repeat#3.
105 p2(Mj,Value){1/(1,1),1/(2,1),1/(3,4)}repeat#3.
```

The result of this change is as follows:

```
?-choice(K,Mj).
(K='distance_parameter',    Mj=1) {MT=0.0000,  T=0.0000  with  CONF=1.0000}
(K='distance_parameter',    Mj=2) {MT=0.0000,  T=0.0000  with  CONF=1.0000}
(K='distance_parameter',    Mj=3) {MT=0.2500,  T=0.0000  with  CONF=0.5000}
(K='distance_parameter',    Mj=1) {MT=0.0000,  T=0.0000  with  CONF=1.0000}
(K='distance_parameter',    Mj=2) {MT=0.0000,  T=0.0000  with  CONF=1.0000}
(K='distance_parameter',    Mj=3) {MT=0.2500,  T=0.0000  with  CONF=0.5000}
```

The system chooses means of combat Mj=1, Mj=2 and Mj=3 with values MT=0.25 and CONF=0.5 for the change of feature values as can be seen in this results of the program.

The main subject of this section is the analysis of the posibility of assigning means of combat M_j to targets T_i, with the change of feature values for each means of combat M_j. These resultes of the analysis will be used for creating the classifier of decision process of ACS. This classifier is based on FNN with new method learning by means of genetic algorithms.

3. SYNTHESIS AND ANALYSIS OF DECISION PROCESSES FOR ACS ON THE BASIS OF FNN

The section presents applications of the clasification decision processes of ACS on the basis of FNN. The learning in the FNN is based on classical genetic algorithms (CGA) and new modified genetic algorithms (MGA).

Neural networks (NN) [10,11] and FNN [12,13,14] can be used for pattern recognition, speech analysis and decision processes classification. The important part for using NN and FNN are learning algorithms in these areas. Classical genetic algorithms [15,16,17,18,19,20] and MGA [12,13] represent an efficient technique based on dynamics of the evolution. These algorithms can be used to create the classifier of decision processes of ACS.

Classical genetic algorithms [15,16,17,18,19,20] can be expressed by the following steps:

- Initialization : let t=0 and randomly choose Q elements from the set of all possible genotypes α for creating the population $\beta(0)$;
- Selection : reproduce genotypes $A_i \in \beta(t)$ with respect to the fitness η genotype A_i and create the intermediate population $\beta(t)$;
- Variation : crossing and mutation modify the intermediate population $\beta(t)$ using CGA to produce the new population $\beta(t+1)$;
- Setting : $t = t + 1$, return to step 2.

Let us consider the decision processes classifier of ACS on the basis of NN in Fig. 1, for the means of combat M_j, j=1,2,3. This structure resultes from the analysis of problem of assigning means of combat M_j on the basis of features, which are written in the paper Part I. It consist of the input layer $IN = \{IN_1, IN_2, ..., IN_7\}$, output layer $OUT = \{OUT_1, OUT_2, ..., OUT_7\}$ and the hidden layer $H = \{H_1, H_2, ..., H_{14}\}$. Neurons are connected as shown in Fig. 1. Each synapsis represents one gene. The alleles are the weights of the corresponding synapsis. Initialization randomly generates alleles for Q organisms. The problem is to determine the weights of the NN in Fig. 1. The weights of synapsis between the input neuron IN_i and hidden neuron H_j are v_{ij} and the weights of synapsis between the hidden neuron H_i and the output neuron OUT_j are w_{ij}. The input to the neuron IN_i is a_i, the output of the neuron H_i is b_i and the output of the neuron OUT_i is c_i. Then

$$b_i = \begin{cases} 1 & \text{if } \tilde{b}_i > \Theta \\ 0 & \text{if } \tilde{b}_i \le \Theta \end{cases} \quad \text{where } \tilde{b}_i = \sum_{j=1}^{14} v_{ji} \cdot a_j, \tag{1}$$

$$c_i = \begin{cases} 1 & \text{if } \tilde{c}_i > \Theta \\ 0 & \text{if } \tilde{c}_i \le \Theta \end{cases} \quad \text{where } \tilde{c}_i = \sum_{j=1}^{7} w_{ji} \cdot b_j, \tag{2}$$

and Θ is the common threshold for all neurons in the NN.

Therefore we proposed FNN in Fig. 2 which has the same topology as NN in Fig. 1 but the outputs are computed as fuzzy and not as classical ones in the following way:

$$c_i = \begin{cases} 1 & \text{if } \tilde{c}_i > \Theta \\ 0 & \text{if } \tilde{c}_i \le \Theta \end{cases} \quad \text{where } \tilde{c}_i = \max_j (\min (w_{ji}, b_j)). \tag{3}$$

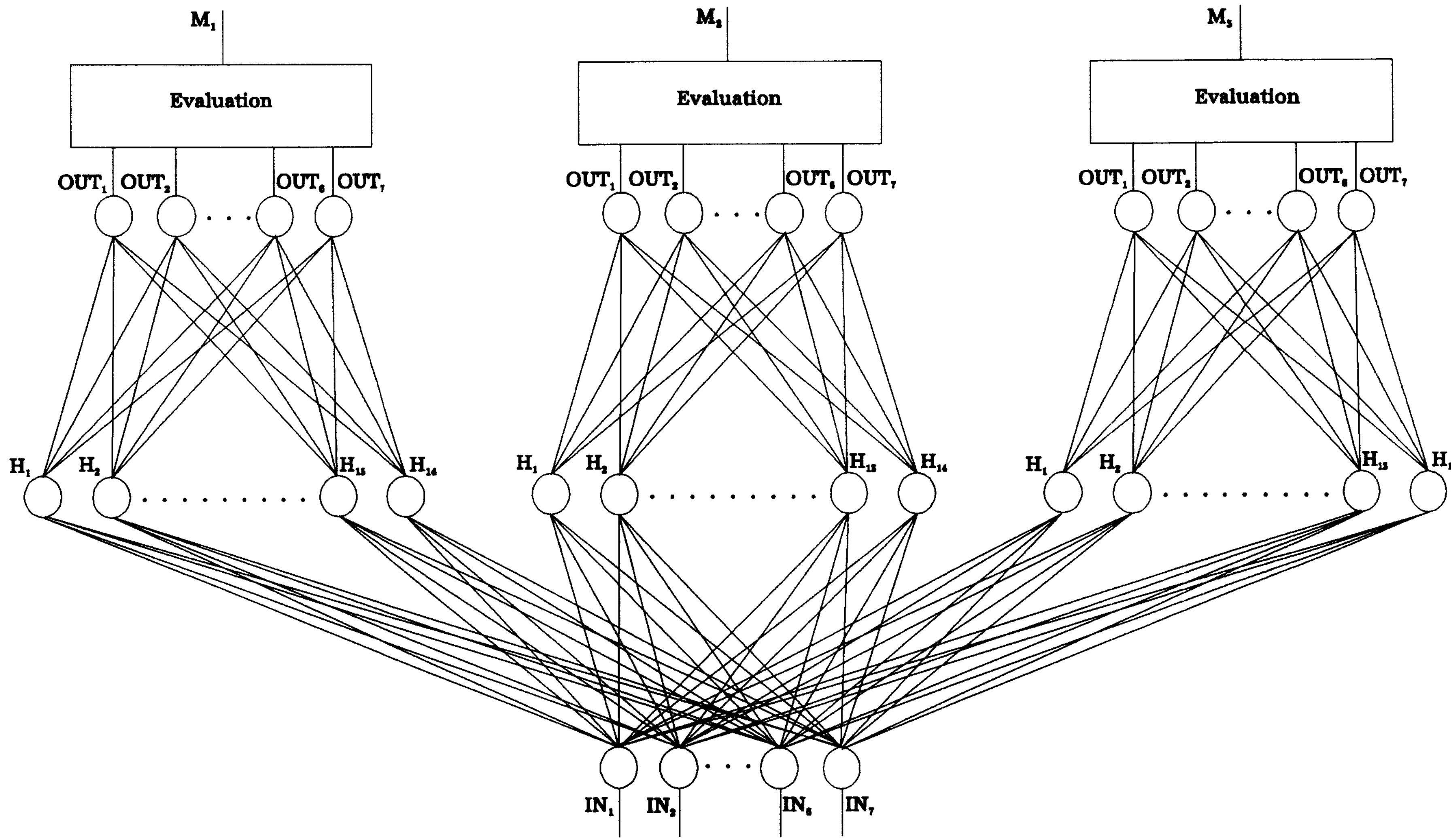

Fig. 1. The classifier of decision processes of ACS by NN

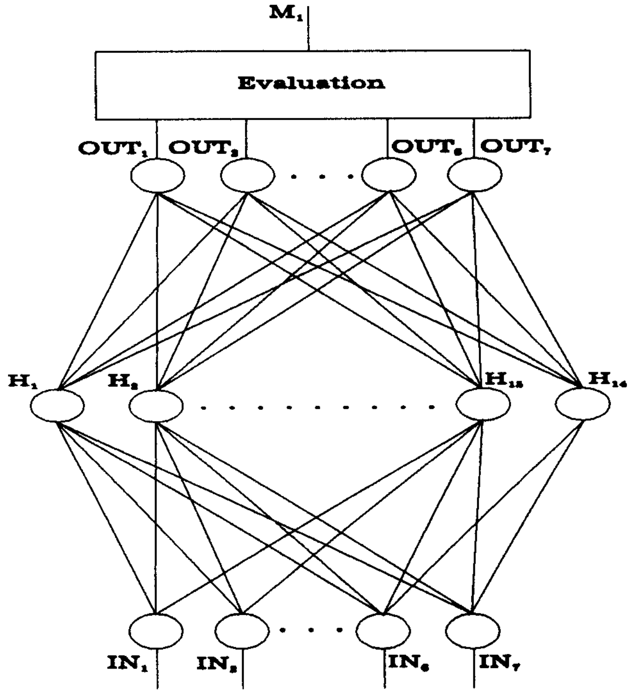

Fig. 2. The classifier of decision processes of ACS on the basis FNN for means of combat M_1

The classifier of decision processes of ACS is realized by means of the evaluation for each feature of means of combat E_1, which are in Table 2.

Table 2. Feature values of equipment M_1

	Feature values						
	1	2	3	4	5	6	7
Features of equipment M_1	1	1	0	0	0	0	0
	1	0	1	0	0	0	0
	1	0	0	1	0	0	0
	1	0	0	0	1	0	0
	1	0	0	0	0	1	0
	1	0	0	0	0	0	1
	0	1	1	0	0	0	0

The fitness η is determined in our experiments as the number of mutually different responses of the FNN between all possible couples of inputting patterns. The maximum is therefore

$$\binom{7}{2} = 21. \tag{4}$$

The Table 3. contains maximum values (max.η) reached in the population and the number of the genotypes having this value.

The input parameters of the FNN and the CGA are the following:

- cardinality of the population $N = 60$;

- probability of the crossing PC = 0.6;
- probabilty of the mutation PM = 0.001;
- threshold TH = 0.5.

The method is convergent until 10th generation as you can see in Table 3.

Table 3. The evolution by CGA for FNN

		Genetic Operators					
		Selection		Crossing		Mutation	
		number	max.η	number	max.η	number	max.η
N	1	15	19	2	20	2	20
	2	14	20	8	20	8	20
	3	6	20	4	20	4	20
	4	13	20	5	20	5	20
	5	6	20	3	20	3	20
	6	10	20	7	20	7	20
	7	12	20	10	20	10	20
	8	13	20	10	20	10	20
	9	19	20	16	20	16	20
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

From the above it results the corresponding weights can be assigned to individual synapsies in the classifier of decision processes of ACS.

The convergence of the CGA is too slow for practical purposes also for FNN. Therefore it seems necessary to modify the CGA [14,15]. The selection in CGA is too random and the syntetical crossing is needed. Therefore a new genetic algorithm was realized, where selection and crossing are put together to one step. In this MGA similarly as in CGA inicialization randomly generates alleles for Q organisms. The selection step chooses the best organisms. These organisms depend on the fitness η for the next evolution [14,15]. The same organisms are randomly crossed in four time steps in one population. The mutation operation is realized as in CGA.

The input parameters of the FNN and the MGA are the following:

- cardinality of the population N = 20;
- probabilty of the mutation PM = 0.01;
- threshold TH = 0.5.

The method is convergent as early as in the third generation as you can see in Table 4. The learning process is finished considerably earlier than in using CGA. Corresponding weights can be assigned to synapsies in the classifier of the decision processes of ACS. This method is convergent quickly, it means that number of generations decrease for more difficult structures of NN and FNN.

Table 4. The evolution by MGA for FNN

		Genetic Operators			
		Crossing		Mutation	
		number	max.η	number	max.η
N	1	1	19	1	19
	2	3	20	4	20

This classifier processes features of means of combat M_j and is realized on the basis of FNN with results of the analysis of FNN learning by means of CGA and MGA. These learning processes enable to state weights of individual synapsies for the definited classifier. The analyzed learning processes show the possibility of method convergence, where evolution of FNN with CGA converges in the 10th generation and evolution with using MGA converges in the 3rd generation. The FNN can process all features of the means of combat with using the paralel way.

4. CONCLUSION

The logic programming language FPROLOG for the synthesis and analysis of decision processes of ACS was used, too. On the basis of the results of program in FPROLOG feature values for means of combat M_j were stated. These features were used according to the recommendation of the classifier of decision processes . This classifier was designed on the basis of FNN for means of combat M_1. Algorithms CGA and MGA were used for FNN learning. All referred methods of the synthesis and analysis of decision processes of ACS meet the requirement of operation in real time.

REFERENCES

[1] ROBINSON, J. A.: A Machine Oriented Logic Based on the Resolution Principle. J. ACM, Vol. 12, No 1, 1965, pp.23-41.

[2] CHMÚRNY, J.- OLEJ, V.- LEHOTSKÝ, M.: Application of Programming Language Fuzzy PROLOG in Decision Processes of Discrete Systems. Electrical Engineering Journal, Vol. 42, No 6, 1991, pp. 294-301.

[3] MUKAIDONO, M.- TAKEDA, A.- SHEN, Z.- DING, L.: On Fuzzy Prolog. Proc. of 3rd Fuzzy System Symposium, Kawasaki, Japan, 1987, pp. 173-178.

[4] SHEN, Z.- DING, L.- MUKAIDONO, M.: A Theoretical Framework on Fuzzy Prolog Machine. Fuzzy Computing, Theory, Hardware, and Applications, North-Holland, Amsterdam, 1988, pp.89-100.
[5] SHEN, Z.- DING, L.- MUKAIDONO, M.: Concept Structure and Its Fuzzy Logical Properties. Proc. of the International Fuzzy Engineering Symposium, IFES '91, Fuzzy Engineering toward Human Friendly Systems, Yokohama, Japan, Vol. 1, 1991, pp.129-140.
[6] MUKAIDONO, M.: Fuzzy Inference of Resolution Style. Fuzzy Set and Possibility Theory, Pergamon Press Inc., New York, 1982, pp.224-231.
[7] ZADEH, L. A.: Fuzzy Sets as a Basis for Theory of Possibility. Fuzzy Sets and Systems, Vol. 1, 1978, pp.3-28.
[8] OLEJ, V.: Choise of a Suitable Etalon in Decision Processes of Discrete Systems with Ambiguity. Bulletin for Studies and Exchanges on Fuzziness and its Applications, BUSEFAL, Université Paul Sabatier, Toulouse, France, No 46, 1991, pp.149-152.
[9] OLEJ, V.: Some Graph and Language Means of Analysis Discrete Systems with Uncertainty. [Sciences Work.], VVTŠ, Liptovský Mikuláš, Slovakia, 1990.
[10] BUCKLEY, J. J.- HAYASHI, Y.: Numerical Relationship between Neural Networks, Continuous Functions, and Fuzzy Systems. Fuzzy Sets and Systems, Vol. 60, 1993, pp.1-8.
[11] LIPPMANN, R. P.: An Introduction to Computing with Neural Nets. IEEE ASSP Magazine, 1987, pp.4-22.
[12] LEHOTSKÝ, M.- OLEJ, V.- CHMÚRNY, J.: Pattern Recognition Based on Neural Networks. In: Zborník zo 7.vedeckej konferencie EF TU, Slovakia, Košice, 1994, 6 p.
[13] LEHOTSKÝ, M.- OLEJ, V.- CHMÚRNY, J.: Pattern Recognition Based on the Fuzzy Neural Networks and Their Learning by Modified Genetic Algorithms. Neural Network World, Neural and Massively-Parallel Computing and Information Systems, Czech Republic, 1995, (in Press).
[14] USHIDA, H.- YAMAGUCHI, T.- GOTO, K.- TAKAGI, T.: Fuzzy-Neuro Control Using Associative Memories, and Its Applications. Control Eng. Practise, Vol. 2, No 1, 1994, pp.129-145.
[15] BUCKLEY, J. J.- HAYASHI, Y.: Fuzzy Genetic Algorithm and Applications. Fuzzy Sets and Systems, Vol. 61, 1994, pp.129-136.
[16] GOLDBERG, E. D.: Sizing Populations for Serial and Parallel Genetic Algorithms. Proc. of the 3rd International Conference on Genetic Algorithms, San Mateo, California, 1989, pp.70-79.
[17] HEISTERMANN, J.: The Application of a Genetic Approach as an Algorithm for Neural Networks. Proc. of 1st Workshop, Parallel Problem Solving from Nature, Dortmund, Germany, 1990, pp.297-301.
[18] KAPSALIS, A.- RAYWARD-SMITH, V. J.- SMITH, G. D.: Solving the Graphical Steiner Tree Problem Using Genetic Algorithms. Journal of the Operational Research Society, Vol. 44, No 4, 1993, pp.397-406.
[19] SPIESSENS, P.: Genetic Algorithms. AI MEMO, Vrije Universiteit, Brussel, No 88-19, 1988.
[20] WHITLEY, D.- STARKWEATHER, T.- BOGART, C.: Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity. Parallel Computing, Vol. 14, 1990, pp.347-361.