VOL. 10, NO. 6, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

EFFICIENCY OF GENETIC ALGORITHMS IN INTELLIGENT HYBRID CONTROL SYSTEMS

Beloglazov Denis A.[1], Finaev Valery I.[1], Zargarjan Jury A.[1], Soloviev Victor V.[1], Kosenko Evgeny Y[1]. and Kobersy Iskandar S.[2]
[1]Southern Federal University, Rostov-on-Don, Russia
[2]Don state technical University, Taganrog, Russia
E-Mail: iskobersi@gmail.com

ABSTRACT

The paper is devoted to features of genetic algorithms application in intelligent hybrid control systems. We demonstrate a general view over the model of hybrid adaptive control system. Also we considered interaction and tuning of elements of the hybrid adaptive control system. Tuning of elements is carried out with the use of genetic algorithms. We designed ten genetic algorithms for research. Researches are carried out when optimization of a multivariable function by genetic algorithms on the example of learning neural network emulator and neuro-fuzzy controller. In the end we made some conclusions about efficiency of genetic algorithms.

Keywords: genetic algorithms, hybrid control system, artificial intelligence, control, learning neural network emulator, tuning of controller, neuro-fuzzy network.

1. INTRODUCTION

The methods of artificial intelligence allow synthesis the hybrid controllers with neural networks and fuzzy logic [1-7]. Such systems are called «neuro-fuzzy». Application of the neuro-fuzzy systems allows successful solution the task of controller parameters adaptation .This task is typical for the conventional controllers designed on the basis of classical control theory [8]. Practical application of hybrid approach requires the use of effective algorithms for the correction of neuro-fuzzy system parameters, for example, algorithms of evolutionary computation [6, 8 - 12, etc.].

Results presented in this paper are continuation of researches dedicated to application of genetic algorithms in intelligent adaptive hybrid control system (AHCS) [13].

Structural model of the AHCS. The model of interaction between elements of adaptive system we present like the sets <*E*, *C*>, where *E* is the set of elements; *C* is the set of links. The set *E* is defined as

$$E=<NFN_1, NFN_2, BAAP, BE, BFTS, BC, BPNFN_2, BAP_1, BAP_2>, \quad (1)$$

where NFN_1 is the first neuro-fuzzy network with constant parameters; NFN_2 is the second neuro-fuzzy network with variable parameters; *BE* is the block of estimation of NFN_1, NFN_2 control signals quality; *BFTS* is the block of real time generation of learning sample for *BE*; *BC* is the coordination block for output signals of controllers NFN_1, NFN_2; *BAAP* is the block of NFN_2 parameters adaptation acceleration; $BPNFN_2$ is the base of NFN_2 parameters; BAP_1 is the block of NFN_1, NFN_2, *BAAP* parameters adaptation; BAP_2 is the block of parameters adaptation of neural network emulator.

The set *C* contains links of elements from E and is defined as

$$C=<\overline{X}(t), N_1(t), N_2(t), N_3(t), K_1(t), K_2(t), \Xi(t), U(t), P_1(t), P_2(t)>, \quad (2)$$

where $\overline{X}(t)$ is the vector of the input controlled parameters; $N_1(t)$, $N_2(t)$ is output control signals of NFN_1, NFN_2; $N_3(t)$ is the class of NFN_2 parameters suitable to the operating mode of control object (CO); $K_1(t)$, $K_2(t)$ are the coefficients of errors of NFN_1, NFN_2 output signals; $\Xi(t)$ is the training sample; $U(t)$ is the integrated control action; $P_1(t)$, $P_2(t)$ are the output signals of *BAAP* containing the set of NFN_2 parameters and the signal of necessity to begin training for BAP_1.

We showed the structural model of the AHCS elements interaction in Figure-1.

We conditionally separated the model of interaction into two parts. The first part is destined to improve the quality of controller operation at the change of the CO operational modes and contains elements capable to adapt their parameters. The second part is destined for providing the uninterrupted functioning of adaptive system when tuning of the first part elements and contains elements with the parameters determined one time before a start and not changing in future.

Application of the AHCS is related to the necessity of preliminary parametric tuning of elements from the second part. Tuning consists of two stages. There is a forming of testing samples for NFN_1 and *BAAP* on the first stage. Experience of experts and data of parameters measuring during the CO functioning is used for this operation.

ARPN Journal of Engineering and Applied Sciences

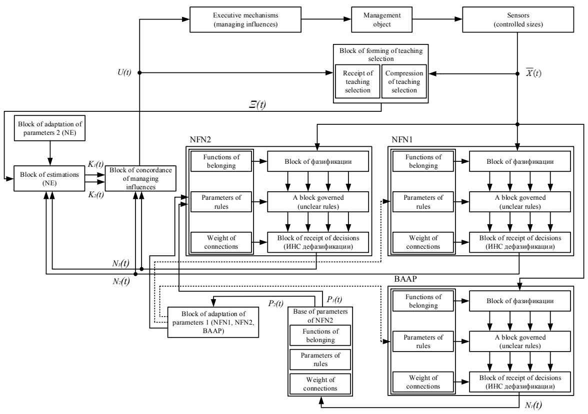


Figure-1. Structural model of AHCS elements interaction.

There is a training of NFN_1 and *BAAP* on the second stage. The elements of NFN_1 and *BAAP* have the parameters unalterable during operation and can be treated as independent controller. We denoted the training of NFN_1 and *BAAP* blocks by dashed line in the structural model of the elements interaction (Figure-1). The training of NFN_1 and *BAAP* occurs before the start of adaptive system operation.

2. PROBLEM STATEMENT

2.1. Research of genetic algorithms during optimization of multivariate functions.

We carried out the research of genetic algorithms (GA) in three stages.

Stage-1. Comparison of different genetic algorithms. We considered as analogues the following algorithms: simple genetic algorithm (GA_1); genetic algorithm with reinicialization as a mutation operator (GA_2); genetic algorithm with the combined operators of random changes (GA_3); fuzzy adaptive algorithm with the combined operators [14] of random changes (GA_4); parallel genetic algorithm with the combined operators of random changes (GA_5); genetic algorithm with the integrated population and combined operators of random changes (GA_6); the first adaptive genetic algorithm (GA_7) [15]; the second adaptive genetic algorithm (GA_8) [16]; fuzzy adaptive parallel algorithms with the combined operators of random changes (GA_9); fuzzy adaptive algorithm with the integrated population and combined operators of random changes (GA_{10}).

Stage-2. Testing of GA operation. We tested GA on the basis of next tasks: optimization of multivariable mathematical functions; training of neural network emulator of the CO; training of neuro-fuzzy classifier; training of neuro-fuzzy controller.

Stage-3. Analysis of GA testing results. Analysis is related to the achievement of information for most effective GA detection.

2.2. Comparison of genetic algorithms during the optimization of multivariable functions.

The task of decision what function should be used for comparison of GA was examined before in papers [17, 18]. We listed types of functions in Table-1.

Table-1. Mathematical functions used for verification of GA efficiency.

Function label	Function	Number of variables
F1	$F1 = 0{,}002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}$	100
F2	$F2 = 0{,}5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0{,}5}{[1{,}0 + 0{,}001(x_1^2 + x_2^2)]^2}$	2
F3	$F3 = 0{,}5 + (x_1^2 + x_2^2)^{0{,}25} \times$ $\times [\sin^2 (50(x_1^2 + x_2^2)^{0{,}1}) + 1{,}0]$	2
F4	$F4 = \sum_{i=1}^{P} \left\| \frac{\sin(\pi k x_i)}{(\pi k x_i)} \right\|$	10
F5	$F5 = (x_2 - (\frac{5}{4\pi^2}) x_1^2 +$ $+ (\frac{5}{\pi}) x_1 - 6)^2 +$ $+ 10(1 - (\frac{1}{8\pi})) \cos(x_1) + 10$	2
F6	$F6 = x_1^2 + 2x_2^2 -$ $- 0{,}3\cos(3\pi x_1) -$ $- 0{,}4\cos(4\pi x_2) + 0{,}7$	2

ARPN Journal of Engineering and Applied Sciences

F7	$F7 = -\cos(x_1)\cos(x_2) \times \times \exp(-((x_1 - \pi)^2 + +(x_2 - \pi)^2))$	2
F8	$F8 = [1 + (x_1 + x_2 + 1)^2 \times \times (19 - 14x_1 + 13x_1^2 - 14x_2 + + 6x_1x_2 + 3x_2^2)] \times \times [30 + (2x_1 - 3x_2)^2 \times \times (18 - 32x_1 + 12x_1^2 - 48x_2 - - 36x_1x_2 + 27x_2^2)]$	2
F9	$F9 = \left\{\sum_{j=1}^{5} j\cos[(j+1)x_1 + j]\right\} \left\{\sum_{j=1}^{5} j\cos[(j+1)x_2 + j]\right\}$	5
F10	$F10 = x_1^2 + x_2^2 + x_3^2$	3
F11	$F10 = \sum_{j=1}^{n-1} [100(x_j^2 - x_{j+1})^2 + (x_j - 1)^2]$	2

The parameters of algorithms GA_1 -GA_{10} are shown for the research of functions F1 - F11 in Table-2.

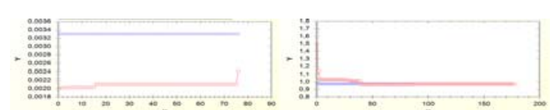
Table-2. Parameters of researches.

		Allowable error	Number of trials	Max. number of iterat-ions	Initializing parameters
Function	F_1	0,0005	1000	1500	random values
	F_2	0,0005			
	F_3	0,0005			
	F_4	0,0005			
	F_5	0,0005			
	F_6	0,0005			
	F_7	0,0005			
	F_8	0,005			
	F_9	0,005			
	F_{10}	0,0005			
	F_{11}	0,0005			

We carried out the researches with the use of developed software application [13]. As an example, we showed the form of achieved results in Figures 2 and 3.

The difference from papers [17, 18] consists in the experimental procedure related to the character of further GA use.

The analysis of researches showed: functions F8 and F9 have a large number of local extremums that complicates optimization. Therefore the value of permissible error of results for F8 and F9 is increased on an order and equals to 0, 005. It allows improvement the performance of genetic algorithms for F8 and F9 without significant loss of accuracy.



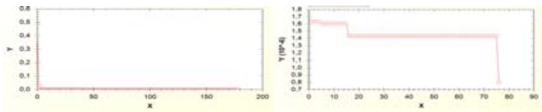
a) F1 b) F2

(- real values, - desired values)

Figure-2. Result of GA_1 operation for F1 and F2.

VOL. 10, NO. 6, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com



a) F1 b) F2

a), b) - mean-squared error of training for F1 and F2

Figure-3. Results of GA_1 operation for F1 and F2.

Initial values of optimized functions parameters are equal to random values in the interval from 0 to 1. We showed summary information about the work of GA_1 - GA_{10} algorithms in the task of functions F1 - F11 optimization in Tables 3 and 4. The results of GA_1 - GA_{10} operation for F1 - F11 functions allow the conclusion about greater efficiency of GA_9 and GA_{10} as compared to other. The number of iterations required for the search of solution with a desired accuracy and the number of algorithm stops in a local optimum appeared to be less than in the compared analogues.

We gave the diagrams showing the number of iterations and stops of genetic algorithms in a local optimum for F1 - F11 functions. The analysis of findings reveals that GA4, GA5, GA6 GA10 algorithms are more effective in the number of iterations and stops of algorithms at a local optimum comparing with GA_1 - GA_3.

Figure-4. Number of GA iterations and stops for F1 - F11.

The superiority of GA_9 is obvious in comparison with GA_1 - GA_{10} algorithms. GA_9 has the parallel properties, the combined operators of random changes, and the possibility of dynamic parameters adaptation. We made true decisions for the acceleration of GA_9, GA_{10} convergence process, prevention of their stacking in local areas due to the application of fuzzy comptroller for the adaptation of parameters and combined operators of casual changes.

Further researches are directed to on an estimatio nefficiency of genetic algorithms on the example of educat ingof the modules of adaptive intellectual hybrid control s ystem.

One of the perspective approaches to the synthesis CO emulators is the use of artificial neural networks (ANN) [19-21]. The CO emulator is called a neuro emulator (NE). As a basis for NE we apply multi-layered perceptron. The task of NE is identification with a desired accuracy of internal dependences and external connections of the CO, i.e. the development of CO identification model [21].

2.3. Comparison of developed GA efficiency on the example of NE training.

We proved the efficiency of GA_1 - GA_{10} for the example of NE training. The number of neurons in the hidden layer of NE is 30. For a representation of ANN we used the mathematical function sin(x/3) +0, 25. In Table-5 (see Appendix) the parameters of GA efficiency research are given for the example of ANN training. The number of GA_1 - GA_{10} iterations and stops at the ANN training is shown in Tables 6 and 7. In Figure-5 we represented the diagram of iterations number and GA stops at a local optimum.

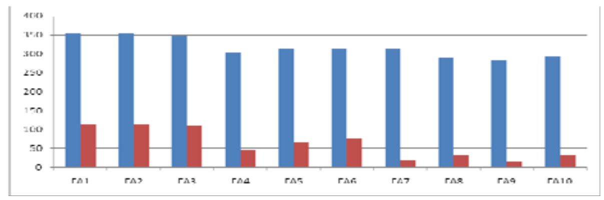


Figure-5. Number of GA iterations and stops at local optimum during ANN education.

We claim on the basis of Tables 6, 7, and interpretation of the data in Figure-5, that the use of GA_4 - GA_{10} has advantages comparing with GA_1 - GA_3. Efficiency of GA_4 - GA_{10} is caused by different factors, for example, by ability to dynamically adapt their parameters, availability of parallelism, etc. GA_9 algorithm has the least number of iterations and stops at the local optimum. Application of GA_9 for the training of CO NE is more appropriate.

2.4. Comparison of GA efficiency on the example of neuro-fuzzy controller training

We applied algorithms GA_1 - GA_{10} for NFN_1 and NFN_2 training. The number of neurons in the hidden layers of neuro-fuzzy controllers is equal to 30. The parameters of GA efficiency researches on the example of NFN_1, NFN_2 is presented in Tables 8 - 10. We cited data from the process of NFN_1, NFN_2 in Tables 8 and 9. This data is the average values on the number of iterations and stops for each investigated algorithm. Data is clustered to 5 groups depending on the number of used training examples.

We showed the diagram of the number of GA iterations and stops at the local optimum at NFN_1, NFN_2 training in Figure-6.

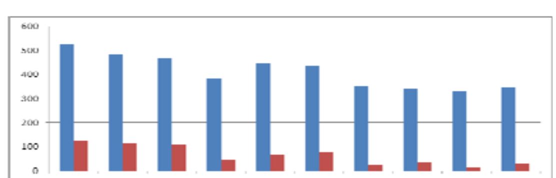


Figure-6. Number of GA iterations and stops at local optimum during NFN_1, NFN_2 training

Analysis of GA_1 - GA_{10} efficiency on the basis of data from Tables 9, 10, and Figure-6 allows talking about the greater efficiency of GA_9 compared to the considered analogues. GA_9 has the best ratio of GA iterations number for the solutions of desired accuracy and stops in the local optimums.

Large number of tuned NFN_1, NFN_2 parameters complicates the system operation. The increasing of generalized examples number affects efficiency of GA application. Large number of training examples causes the increasing of decision search time.

Conclusions: We have idea about the efficiency of GA_1 - GA_{10} algorithms due to the results of presented researches. We carried out analysis on the basis of absolute iteration number values, GA stops at the local optimum in all experiments. In Figure-7 the diagram of number of GA general iterations and stops at the local optimum is shown.

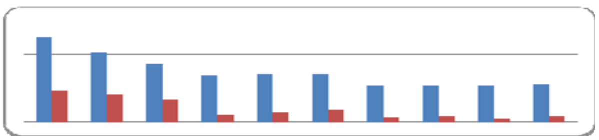


Figure-7. Number of GA general iterations, stops at local optimum.

3. CONCLUSIONS

Analysis of data on Figure-7 allows conclusion that GA_9 (fuzzy adaptive parallel GA with the complex operators of random changes) has the best performance compared to the researched analogues. Researches of GA_1 - GA_{10} proved validity of the suggestion that the use of complex operators [14] of random changes is able to improve the process of GA convergence.

Algorithms GA_5, GA_6 showed good results, but absence of parametric adaptation of operators decreases their efficiency. Application of the mutation operator of reinitialisation positively effects on the algorithm speed of search for decisions, but the substantial rise of efficiency is not observed. The results of GA_2 researches proved that. The efficiency of GA_9 increased in 1, 06 times (6%) on the number of iterations, in 1, 12 times (12%) on the number of stops at the local optimum As compared to GA_7. The efficiency of GA_{10} increased in 1,005 times (0, 005%) on the number of iterations and decreased on 28% on the number of stops at the local optimum as compared to GA_7. The efficiency of GA9 increased in 1,045 times (4, 5%) on the number of iterations, the number of stops in 1, 57 times (57%) in comparing with GA_8. The efficiency of GA_{10} decreased in 1, 03 times (3%) on the number of iterations, increased in 1, 02 times (2%) on the number of stops as compared to GA_8.

ACKNOWLEDGMENT:

This work was supported by the grant of Southern Federal University №213.01-07-2014 / 02 ПЧВГ.

REFERENCES

[1] Пупков К.А., Егупов Н.Д. и др. 2001. Методы робастного, нейро-нечеткого и адаптивного управления/Под. ред. Н.Д. Егупова- М.: МГТУ им. Н.Э. Баумана. - 744 с.

[2] Li-Xin Wang. 1993. Stable Adaptive Fuzzy Control of Nonlinear Systems//IEEE Transactions on Fuzzy Systems. 1(2).

[3] Рутковская Д., Пилиньский М., Рутковский Л. 2006. Нейронные сети, генетические алгоритмы и нечеткие системы/Пер. с польского. И.Д. Рудинского. - М.: Горячая линия - Телеком. - 452 с. - ISBN 5-93517-103-1.

[4] Jeffrey T. SpoonerandKevinM. Passino. Stable Adaptive Control Using Fuzzy Svstems and Neural Networks//IEEE Transactions on Fuzzy Systems. 4(3): 339.

[5] Leszek Rutkowskii. 2004. Flexible neuro-fuzzy systems. Structures, learning and performance evaluation. - Technical University of Czestochowa.

[6] Abdul, Hafez, Ahmed Alrabie and Arun Agarwal. 2004. Design of Neuro-fuzzy Controller Based on Dynamic Weights Updating. - Springer - Verlag. pp. 58-67.

[7] Соловьев В.В. 2010. Синтез гибридных регуляторов-наблюдателей для нелинейных систем Известия ЮФУ. Технические науки. Тематический выпуск. «Методы и средства адаптивного управления в энергетике». - Таганрог: Изд-во ТТИ ЮФУ. № 1 (102). - 262 с.

[8] Kobersy Iskandar S., Finaev Valery I. and others. 2014. Model of the controller for output stream concentration in the mixer of a steam unit. ARPN Journal of Engineering and Applied Sciences. VOL. 10, NO. 4. pp. 1637- 1641.

[9] Pejman T. 2012. Ardeshir Hezarkhani. A hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation. Computers and Geosciences. 42: 18-27.

[10] Mehrabi M., M. Sharifpur and J.P. Meyer. 2012. Application of the FCM-based neuro-fuzzy inference system and genetic algorithm-polynomial neural network approaches to modelling the thermal conductivity of alumina-water nanofluids. International Communications in Heat and Mass Transfer. 39(7): 971-977.

[11] Tayfur G. Soft computing in water resources engineering. Artifical neural networks, fuzzy logic and genetic algorithm. Ashurt Lodge, Ashurst, Southamption. ISBN 978-1-84564-363-3.

[12] Sivanandam S.N. and S.N. Deepa. 2010. Inriduction to genetic algorithm. Springer-Verlag, Berlin Heidelberg. ISBN 978-3-642-09224-4.

ARPN Journal of Engineering and Applied Sciences

[13] Beloglazov D.A. and V.I. Finaev. 2013. Development of Information Support Aiming at Investigation into Trainability of Hybrid Adaptive Neuro-Fuzzy Regulators//World Applied Sciences Journal. 23(11): 1433-1440.

[14] Белоглазов Д.А. 2011. Разработка адаптивного генетического алгоритма с комплексными операторами случайных изменений для обучения нейро-нечеткого регулятора. Технологии Microsoft в теории и практике программирования: труды VIII-ой Всероссийской конференции студентов, аспирантов и молодых ученых. Южныйрегион. - Таганрог: Изд-воТТИЮФУ. С. 44-53.

[15] Srinivas M. and L.M. Patnaik. 1994. Adaptive probabilities of crossover and mutation in genetic algorithms//IEEE Transaction on systems, man and cybernetics. 24(4): 656-667.

[16] Wang L. and Dun-bing Tang. 2011. An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem//Expert Systems with Applications. 38(6): 7243-7250.

[17] Fayad C and Sanja Petrovic. 2005. A fuzzy genetic algorithm for real-world job shop scheduling//Proceeding IEA/AIE' 2005 Proceedings of the 18th international conference on Innovations in Applied Artificial Intelligence Springer - Verlag London, UK.

[18] Chelouah R. and P. Siarry. 2000. A continuous genetic algorithm designed for the global optimization of multimodal functions//Journal of Heuristics. 6: 191-213.

[19] Singn M., Smriti Srivastava and M. Handmandlu. 2007. Identification of a Nonlinear System using Neural Networks by Extracting the System Dynamics. IETE Journal of Research. 53(1): 43-50.

[20] Al Seyab R. K. and Y. Cao. 2008. Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation //Journal of Process Control. 18(6): 568-581.

[21] Kobersy I.S., Ignatev V.V. and others. 2014. An intelligent navigator with the control of the car technical condition. ARPN Journal of Engineering and Applied Sciences VOL. 9, NO. 7.

VOL. 10, NO. 6, APRIL 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

APPENDIX

Table – 3. Number of GA_1 - GA_{10} iterations for F1 - F11 functions.

Function	Length of chromosomes	Number of iterations									
		GA_1	GA_2	GA_3	GA_4	GA_5	GA_6	GA_7	GA_8	GA_9	GA_{10}
F1	2300	78	75	82	70	85	94	69	60	36	53
F2	46	178	132	180	95	90	109	75	88	106	94
F3	46	453	375	346	235	215	240	195	181	187	203
F4	230	325	190	150	110	147	152	80	82	75	78
F5	46	270	231	84	73	76	81	73	54	49	52
F6	46	323	249	139	117	153	125	62	61	63	55
F7	46	380	285	141	100	70	65	55	79	59	68
F8	46	549	354	341	303	301	275	168	193	167	189
F9	115	215	198	170	143	142	150	72	85	89	92
F10	69	185	170	171	98	80	83	75	50	78	64
F11	46	604	513	348	315	307	311	251	275	235	242

Table – 4. Number of GA_1 - GA_{10} stops for F_1 - F_{11} functions.

Function	Number GA stops in a local optimum									
	GA_1	GA_2	GA_3	GA_4	GA_5	GA_6	GA_7	GA_8	GA_9	GA_{10}
F1	88	76	61	2	15	19	0	0	0	0
F2	210	185	141	38	45	51	27	36	25	33
F3	93	86	69	17	21	29	9	17	6	18
F4	107	99	78	47	49	53	33	45	33	51
F5	65	59	28	0	5	8	0	0	0	2
F6	52	48	21	1	7	9	0	0	1	1
F7	73	70	59	10	18	21	3	5	1	7
F8	278	264	241	54	97	115	57	72	54	49
F9	47	35	17	0	4	11	0	0	0	0
F10	189	135	95	23	32	53	0	0	9	19
F11	289	241	197	25	84	121	27	36	19	28

Table – 5. Parameters of algorithms researches.

GA	Allowable error of generalization for one training example	Number of trials	Max. number of iterations	Initialization of NE parameters
GA_1 - GA_{10}	0,075	1000	1500	random values

Table – 6. Number of GA_1 – GA_{10} iterations for ANN training.

Number of examples	Length of chromosom	Number of GA iterations									
		GA_1	GA_2	GA_3	GA_4	GA_5	GA_6	GA_7	GA_8	GA_9	GA_{10}
5	22080	33	32	36	25	32	33	35	32	29	30
10		38	40	37	31	35	34	45	35	31	31
25		70	70	64	56	61	63	56	57	53	59
50		72	72	72	65	64	65	64	58	60	62
100		142	140	138	126	123	117	114	109	110	111

Table – 7. Number of GA_1 – GA_{10} stops for ANN training.

Number of examples	Amount of stops of GA in a localoptimum									
	GA_1	GA_2	GA_3	GA_4	GA_5	GA_6	GA_7	GA_8	GA_9	GA_{10}
5	15	16	17	3	13	13	0	0	0	0
10	19	21	18	5	9	10	1	3	0	0
25	21	22	20	9	13	15	3	8	3	8
50	27	25	23	13	14	17	5	9	5	11
100	33	30	31	17	18	21	10	12	9	13

Table – 8. Parameters if algorithms researches.

GA	Allowable error of generalization for one training example	Number of trials	Max. number of iterations	Initialization of NFN_1, NFN_2 parameters
GA_1 - GA_{10}	0,075	1000	1500	random values

Table – 9. Number of GA_1 - GA_{10} iterations for training of NFN_1, NFN_2

Number of examples	Length of chromosome	Number of GA iterations									
		GA_1	GA_2	GA_3	GA_4	GA_5	GA_6	GA_7	GA_8	GA_9	GA_{10}
5	41160	68	60	57	50	54	52	46	45	44	47
10		87	85	81	68	76	75	62	61	59	61
25		114	105	100	77	95	94	68	69	67	71
50		125	110	109	85	107	103	84	77	74	75
100		135	125	120	103	118	114	90	90	85	95

Table – 10. Numbed of GA_1 - GA_{10} stops during NFN_1, NFN_2 training.

Number of training examples	Number GA stops in a local optimum									
	GA_1	GA_2	GA_3	GA_4	GA_5	GA_6	GA_7	GA_8	GA_9	GA_{10}
5	15	14	15	1	14	14	0	0	0	0
10	17	16	17	3	11	13	2	2	0	0
25	26	22	21	5	12	15	6	7	2	5
50	29	25	25	17	14	18	7	12	4	8
100	37	35	31	20	19	19	13	15	9	16