

RIGA TECHNICAL UNIVERSITY

Arita TAKAHASHI

GENETIC ALGORITHMS IN COMPLEX  
OPTIMISATION TASKS

Summary of Doctor's Work

Riga-2004

## GENERAL DESCRIPTION OF WORK

**Topicality of problem.** Nowadays optimisation is a very important branch of research. Its particular tasks are multicriterial optimisation, search for the Pareto -optimal set, optimisation of multimodal functions, and optimisation of noisy or dynamically changing functions. At the same time the speed of computing operations increases and so does the size of the computer memory which gives the opportunity for effective implementation of resources demanding and time demanding algorithms for finding the solution with sufficient speed.

Genetic algorithms perform the search process not at one point of the parameter space, but at several points at once. Genetic algorithms use operators which are probabilistic. It means that although initial populations can be identical, all other populations in every trial would be different. The chance of individual (object) survival till the next generation depends on its fitness compared to the average fitness of the population.

The random features of genetic algorithms make them adaptive to an objective junction if its value changes in time dynamically. Though classical genetic algorithm has certain disadvantages, which cause extremely bad impact on optimisation process in certain tasks. One of such drawbacks is premature convergence (rapid convergence to local optimum). It is caused by assimilation process when moderate and good solutions, which are represented in population in majority, assimilate the best solutions which are very few due to the peculiarities of objective function. The main aim of this research paper is to study this particular problem and its impact on optimisation process.

**Goals and tasks of the research.** The employable aim of this study is to create a genetic algorithm which is applicable to optimisation of multimodal functions. The indirect aim of the research is to develop a sub-class of genetic algorithms which are specially intended for optimisation of multimodal functions.

The **optimisation task** of this paper is formulated as follows:

To maximize a function  $f(x_1, x_2, \dots, x_n)$ , where every  $x_i$  is in boundaries:

$$a_i \leq x_i \leq b_i$$

for some restrictions  $a_i$  and  $b_i$ .

To create such genetic algorithms, one has to solve the following problems:

1. To examine genetic algorithms as generators of new solutions. To create a program for such a generator.
2. To explore an analogy between binary genetic algorithms and real genetic algorithms. To explore an analogy of real crossover operator and binary crossover operator. To compare real and binary mutation operators. To investigate the usage of binary code and Gray code in genetic algorithms.
3. To investigate various crossover operators of genetic algorithms. To observe and compare the impact of various crossover strategies on optimisation process and premature convergence.
4. To investigate premature convergence and its impact on maximum search for different functions with one global maximum. To observe the difference in

population behaviour in cases when the maximum lies in the centre of search space and in cases when it does not.

5. To explore the performance of genetic algorithms in task of search for maximum of multimodal function. To create a genetic algorithm which does not show premature convergence like a classical genetic algorithm does.
6. To investigate optimisation problems in cases when objective function contains global maximum domains. To create algorithms applicable to optimisation of such functions.
7. To create a genetic algorithm which is applicable to tasks of multicriterial optimisation. To compare multicriterial optimisation, using maxi-min and maxi-max\* strategies and not using any decision strategies at all.
8. To formulate a new sub-class of genetic algorithms as a model of artificial selection. To find out possible exploitation of both natural selection models and artificial selection models.
9. To elaborate software for real genetic algorithms and their application to optimisation of real functions.
10. To create software for real genetic algorithms and their application to searching of Pareto optimal solutions in case of multicriterial optimisation.

**Object and subject of the research.** The object of the research is evolutionary algorithms. The subject of the research is optimisation of functions using real genetic algorithms.

#### **Research hypotheses**

1. Classical genetic algorithm is not able to find a global maximum of multimodal function
2. In task of optimisation of multimodal function new genetic operators (such as coupling) and new strategies must be used.
3. It is possible to create a genetic algorithm which finds diverse solutions in Pareto set or multi-maximum domain so that they are distributed in all the set uniformly.

**Methods of the research.** In the thesis, the following research methods were used: algebra and calculus, discrete mathematics, set theory, statistics, genetic algorithms, decision making methods, methods of artificial intelligence, software for statistics SPSS, and software for statistics MathCad.

**Scientific novelty of work.** The thesis contains an analysis of the behaviour of genetic algorithms, which is based on both experimental results and theoretical conclusions. It can be used to create new genetic algorithms.

In this research a "multilayer genetic algorithm" has been created, which is more effective for tasks of optimisation of multimodal function than a classical genetic algorithm. The coupling operator, which is based on arranging of solutions according to their fitness value, is an original idea for reduction of premature convergence.

In this research a "restricted genetic algorithm" has been created, which provides certain diversity of solutions in case of several (finite number of) solutions or in case of infinite number of solutions. The idea of arranging solutions according to their parameter values and including them into certain subpopulations can be regarded as a model of artificial selection.

**Theoretical value.** In the research a new approach to the behaviour analysis of genetic algorithms is offered that uses statistical package SPSS. Also, two-parameter test functions, which allow comparison of premature convergence of genetic algorithms, are offered.

The exploration of the behaviour of the modified algorithms leads to the further investigation of algorithms as the suggested genetic algorithms are based on a new theoretical approach. The new idea relates to the strategy of crossover: in order to prevent premature convergence and to provide diversity of solutions, objects with similar parameter values must be crossed over. Techniques for coupling (those of similar parameters) can be various, and dividing the population into several subpopulations depending on the value of their parameter values is one of the techniques, while coupling according to the fitness value is other one.

#### **Practical value**

1. Software program "Rgenalg" was developed. It will be used for further experiments. Also, it will be used to prepare illustrations for students. The program parameters are the genetic algorithm parameters, and its output product is its database with experimental results. The data shows all the information of the population in certain (fixed) generation, i.e. moment of time. The program cannot be regarded as the one, which should serve for commercial goals and be sold in the market.
2. The results of the research were presented to students, and some of the experiments performed have been used as an illustrative material for lecture course "Evolutionary genetic algorithms"
3. The description files of test functions in the MathCad package can serve as a basis for creation of new functions.

#### **The approbation of work**

The results of the research were presented to the audience of scientists at the following conferences:

1. 40. RTU studentu zinātniska un tehniska konference, 1999-gada 26 - 30. aprīlis (Rīga, Latvija)
2. Computational Intelligence: Theory and Applications; International Conference; Proceedings/ 6th Fuzzy Days Dortmund, Germany, May 25-28, 1999 (Dortmund, Germany)
3. "The Second International Conference on Soft and Intelligent Computing. Budapest, Hungary May 25-28, 1999" (Budapest, Hungary)
4. "The 2nd International Conference Environment. Technology. Resources, June 25-27, 1999." (Rēzekne, Latvia)
5. Starptautiska konference "Baltijas Reģiona Valstu Integrācijas Problēmas Ēs uz Eiropas Savienību", Rēzekne, Latvija 2000.gada 2.-3.marts (Rēzekne, Latvia)

6. "KES'2000" /Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies, University of Brighton, UK, 30, 31 August, 1 September 2000 (Brighton, U.K.)
7. The Second International conference "Simulation, Gaming, Training and Business Process Reengineering in Operations RTU September 8-9, 2000" (Riga, Latvia)
8. ES2000, the Twentieth SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence, Cambridge, December 11th-13th, 2000 (Cambridge, U.K.)
9. "International Conference on Computational Intelligence. Theory and Applications: 7 Fuzzy Days, Dortmund, Germany, October 1-3, 2001" (Dortmund, Germany)
10. RTU 43 Starptautiska Zinatniska konference, Riga, 2002.gada 10.oktobris - 14.oktobris, apaklsekija "Informācijas tehnoloģija un vadības zinātne" (Riga, Latvija)

**Publications.** The results of the research are represented in ten published papers, performed by the author independently and in co-authorship. Seven of the papers are published abroad.

**Personal contribution.** All the results of the research described in the dissertation are obtained by the author via unaffiliated research.

**The structure and volume of dissertation.** The dissertation consists of five Chapters and conclusion, Table of Contents, Bibliography and an appendix. The main text of the research is presented on 218 pages and illustrated with 204 pictures and 73 tables. The list of references includes 21 entry.

## BRIEF CONTENT OF WORK

*Chapter 1* describes the basic principles of the genetic algorithms. The genetic algorithms are stochastic iterative algorithms and they can be described by three basic steps, common to these methods (see "New Ideas in Optimisation" by David Come, 2000):

Step 1 Begin: generate and evaluate initial a set of solutions  $S$ , consisting of candidate solutions

Step 2 Operate: create a set of new solutions  $S'$  by making randomised changes to some selected members of  $S$

Step 3 Renew: randomly replace some members of  $S$  with members of  $S'$  and then (unless some terminating condition is met) return to step 2.

*Chapter 1* tells that existing stochastic iterative methods can be divided into two large classes: local search and population-based search. In case of local search one current solution and its neighbour solutions are considered. In case of population-based search a whole population of diverse solutions is maintained. Such methods are known as "evolutionary algorithms".

*Chapter 1* also describes how the canonical genetic algorithm appeared as the model of the evolutionary theory of Darwin. An assumption is made that any real solution (phenotype) is encoded within some particular combination of genes (called genotype) which was represented as a string of symbols. The heredity was imagined

as maintenance of certain sub-strings of symbols for the future generations. The variation of beings was explained by random changes (mutations) of the genes caused by accidental impacts.

There is a short description of canonical genetic algorithm and its block diagram in *Chapter 1*.

Further, genetic algorithm as a generator of new solutions is discussed. The problem of human limited creativeness is explained and some experiments to solve this problem by means of genetic algorithms are described.

*Chanter 2* proposes region-based crossover which generates offspring into a region delimited by parameter values of their parents (see Fig. 1). The disadvantages of binary encoding of real parameters are discussed (Goldberg, 1989). The idea of a real genetic algorithm as analogy of the binary one (Wright, 1991) is shown.

*Chapter 2* shows how the binary code and the Gray code can be used in genetic algorithms.

*Chapter 2* interprets real crossover as an analogue of crossover of binary strings (Wright, 1991). This real crossover is called combinatorial crossover in the paper, because two sub-strings of real parameters are combined in a new string, describing a new solution. The description of a random change of a randomly chosen parameter is shown as an analogue of mutation in a binary string (Wright, 1991) afterwards.

Then next the description of region-based crossover follows. The author shows an illustration of two-parameter space where one of the parents has parameter values (2.5; 0.15), but the other one has parameter values {3.7; 0.4}. In Fig. 1 we can see that the children of both parents would be generated randomly in the region delimited by parameter values of their parents. The author considers such a crossover strategy as a model of crossover of living beings, for example, baby hair colour if the baby is of a blond mother and a dark-hair father can be in various colourings from blond till dark. The author also explains that region-based crossover does not need encoding and decoding, because the real parameters can be used directly.

The author sees the parameter values of a child solution as parameter values of one of its parents, perturbed in the direction of the other parent.

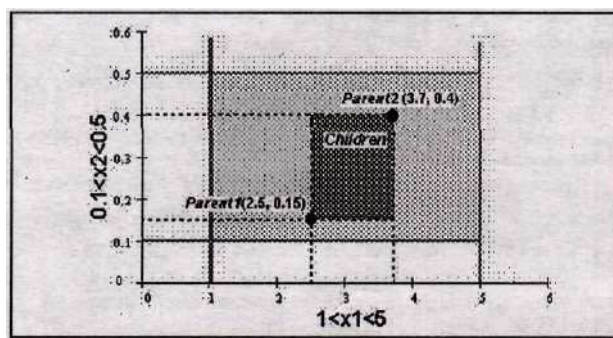


Fig.1. The region of the children to be randomly generated in the area delimited by parameter values of their parents

The main advantages of the region-based crossover are:

1. The space is searched not at discrete points which are  $n$  sixteenth ( $n/16$ ) or  $n$  sixty-fourth ( $n/64$ ) of the space size in the direction of a certain parameter axis, but in continuous space with the assigned precision;
2. The algorithm gets simpler (it isn't precision-dependent any more);
3. The economy of resources is achieved, because the binary coding and decoding is not needed.
4. The region-based crossover does not create infeasible or illegal children.

The region-based crossover is compared with combinatorial crossover through both single trials (randomly chosen) and series of 18 trials (the mean of population fitness and the mean of the best fitness is compared). The mutation in both cases means creating a new parameter value for one of parameters which is generated as a random number in the interval  $[X_{n,j_n}; x_{n\max}]$ . Mutation is only applied if both parents are copies of the same solution, i.e. mother=father. Then one of the parents is left unchanged while the other one and both their children go through the mutation of one parameter.

In *Chapter 2* the diagonal crossover (when the children are generated at the random point of the line connecting both parents) is discussed. It is compared with the arithmetic crossover (when children are generated at the points, dividing the same line in proportion 1:3 and 2:3).- The trial with arithmetic crossover approves the previous studies (Wright, 1991), which demonstrates that the arithmetic crossover is more effective than the combinatorial crossover in many optimisation problems.

In *Chapter 3* the author proposes a multilayer genetic algorithm to search for the maximum of multimodal functions.

At the beginning of the chapter the drawbacks of classical genetic algorithm are mentioned:

1. The genetic algorithm can find a local optimum instead of global one.
2. The possibility of successful optimisation lies in the characteristics of the objective function as well as in the operators of the genetic algorithm and its parameters.
3. If the objective function has more than one global optimum, the genetic algorithm will either find only one of them or show premature convergence.
4. If the objective function has a region of global optimums (infinite number of optimums), then the classical genetic algorithm will find only one solution belonging to the region (instead of finding many diverse solutions, all of them being global). Or possibly, the classical genetic algorithm may end the search process in premature convergence.

Next in *Chapter 3* the classical genetic algorithm with region-based crossover is described.

After that the behaviour of the classical genetic algorithm in the task of optimisation of multimodal function is explored. The premature convergence is observed.

Then the coupling operator is proposed to diminish premature convergence of classical genetic algorithm.

The phenomenon of premature convergence is illustrated in Fig. 2(b), where two cross-sections of the function depicted in Fig. 2(a) are shown. One can see a region of fine fitness level of  $y=1.5866$  on the cross section with  $X_2=0.63$  when the parameter  $X_i$  values are in range  $[-0.2; 0.2]$ . Although the first population always contains solutions

near global maximums with  $y=1.874$ , predominance of the solutions in the local maximum always causes the assimilation of the first ones.

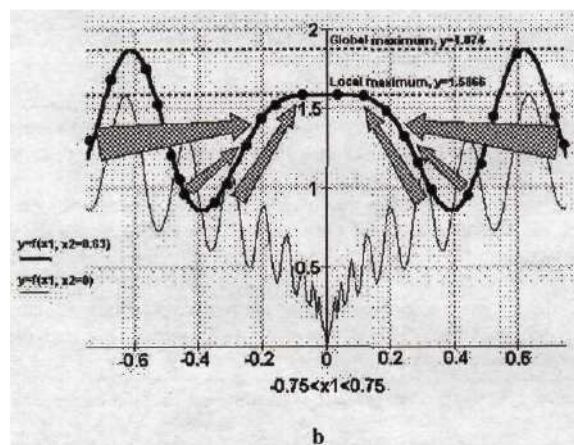
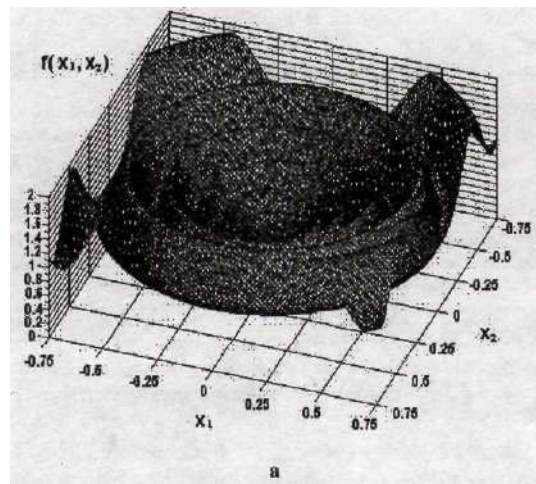


Fig.2 Objective function (a) and Premature convergence of classical genetic algorithm to the areas of fine fitness which can be explained by predominance of fine solutions, which causes assimilation of the best ones (b)

The hypothesis about premature convergence is stated as follows: The premature convergence diminishes if free crossover is infringed by ranging the solutions according to their fitness value (in its descending direction). Then the first best solution is coupled for crossover with the second one, the third is coupled



with the fourth one, etc. That strategy can only be used if similar objective function fitness values are mapped to similar parameter values.

Further in *Chapter 3* the multilayer algorithm is described. The algorithm can be brought to the following steps:

- 1) generate the initial population randomly (consisting, say, of 200 solutions);
- 2) calculate the fitness value of every solution  $Y$ ;
- 3) save the results;

4) arrange solutions in the descending sequence of  $Y$ . To assign couple numbers so that the first two (best) solutions have a couple number equal to 1, the next two solutions have a couple number equal to 2 and so on;

5) cross-over (the region-based crossover, see Fig. 1) parents (solutions with the same couple number) twice, to obtain two children. In case of both parents being copies of the same solution, mutate the one of the parents and both children. The mutation is performed by re-generating one of the parameter values (chosen randomly) according to uniform distribution in the whole range of the solution space;

6) calculate fitness value  $Y$  for every solution;

7) calculate survival probability  $SELPROB_i$  (with which it will be selected for

the next generation) for every solution as 
$$\frac{Y_i}{\sum_j Y_j} ;$$

8) calculate cumulative probability  $CUMPROB_i$  for  $i$ -th solution as  $CUMPROB_{i-1} + SELPROB_i$ ;

9) perform roulette wheel selection (200 times);

10) check if the final condition is met (for example, if 100 iterations has been performed). If it is not, then go back to the step 3, otherwise end the procedure;

Further in *Chapter 3*, an example with the multilayer genetic algorithm is shown. The task is to find the maximum of the "stool leg" function (see Fig.3, case a and equations (1), (2) and (3)) with the population size = 8.

The parameters of the experiment are the following: *Population\_Size*=8, the terminating condition is *Loop\_Count*=7, crossover probability is 1 and children number for one couple is 2 (every couple has two children), crossover type is "region-based crossover", function type is "stool leg" function, type of the algorithm is "multilayer genetic", the parameter values of the solution space:

$$-0.5 \leq x_1 \leq 0.5 \quad -0.5 \leq x_2 \leq 0.5$$

Further in *Chapter 3* the initial population of the experiment is shown. Its every parameter value is a randomly generated number belonging to the interval  $[-0.5; 0.5]$ . The initial population is depicted in Table 1.

X1	X2	Y	GENERATION
0,11	0,20	3,05	1
-0,37	0,21	1,64	1
0,17	-0,39	1,75	1
0,22	-0,07	3,08	1
0,34	0,13	2,18	1
-0,07	0,09	3,76	1
-0,18	-0,40	1,66	1
0,13	-0,35	2,16	1

The column title "X1" denotes parameter  $x_1$ , column title "X2" denotes parameter  $x_2$ , column title "Y" denotes function value, but "GENERATION" denotes the number of the current generation.

Further in this example let's assume that two iterations of the algorithm have been performed and we will follow the execution of the third iteration of multilayer genetic algorithm. All the data were documented from a real experiment.

In Table 2 the third population is shown. It differs from the initial population in limited diversity of solutions, because the solutions of this generation are copies of the few solutions with the highest fitness value.

The third population

Table 2

X1	X2	Y	GENERATION
0,15	0,05	3,54	3
0,05	0,05	3,90	3
0,05	0,05	3,90	3
0,11	0,01	3,77	3
0,11	0,20	3,05	3
0,11	0,20	3,05	3
-0,07	0,09	3,76	3
0,15	0,18	3,01	3

Next in *Chapter 3* "step 4" of the multilayer genetic algorithm is shown. The solutions are arranged in the descending sequence of Y. For every two neighbour solutions (starting from the top) the same number of couple is assigned. The roles "mother" and "father" do not show any particular meaning for the algorithm, and they are added just to make it easier to figure out the parents of every "family".

The third population, already arranged, is shown in Table 3.

The third population, arranged

Table 3

X1	X2	Y	MEMBER	COUPLE	GENERATION
0,05	0,05	3,90	Mother	1	3
0,05	0,05	3,90	Father	1	3
0,11	0,01	3,77	Mother	2	3
-0,07	0,09	3,76	Father	2	3
0,15	0,05	3,54	Mother	3	3
0,11	0,20	3,05	Father	3	3
0,11	0,20	3,05	Mother	4	3
0,15	0,18	3,01	Father	4	3

Further in *Chapter 3* the 5th step of multilayer genetic algorithm is shown. After crossing the first two parents, children as two their copies are obtained, because both "Mother" and "Father" are copies of the same solution (see Table 4). That leads to mutation of one of "Mother's" parameters. As random number  $Rand\_num < 0.5$  .therefore the parameter  $x_1$  of "Mother" is going to be mutated. It is generated as a random number in range  $]-0.5; 0.5]$  and  $x_1 = -0.37$  is obtained. Then the Y value of the "Mother" is recalculated, and  $y = 2.32$  is obtained (see Table 4). The same mutation operator is applied to both children of Couple 1.

The children of the second couple are generated in the following way: The parameters  $x_1$  of both parents delimit range  $]-0.07; 0.11]$ , and the  $x_1$  value of the child is generated as a random number in the range. In similar way, the  $x_2$  value of the child is generated as a random number in the range  $]0.01; 0.09]$ . For the first child the random numbers are  $x_1=0.03$  and  $x_2=0.08$ , but for the second child they are  $x_1 = -0.07$  and  $x_2 = 0.03$  (see table 4). The values of  $Y$  are calculated (step 6 of the algorithm) according to equations (1), (2) and (3).

Formation of the children of the third generation in the region crossover Table4

COUPLE	MEMBER	X1	X2	Y
1	Child1	0,05 -0,25	0,05	3,90 2,97
1	Child2	0,05 0,05 -0,14	0,05	3,90 3,59
1	Mother	0,05 -0,37	0,05	3,90 2,32
1	Father	0,05	0,05	3,90
2	Child1	0,03	0,08	3,86
2	Child2	-0,07	0,03	3,90
2	Mother	0,11	0,01	3,77
2	Father	-0,07	0,09	3,76
3	Child1	0,14	0,11	3,39
3	Child2	0,14	0,11	3,41
3	Mother	0,15	0,05	3,54
3	Father	0,11	0,20	3,05
4	Child1	0,15	0,20	2,90
4	Child2	0,12	0,20	3,04
4	Mother	0,11	0,20	3,05
4	Father	0,15	0,18	3,01

Next the survival probability for every solution should be calculated (step 7). To do that, the sum of all the function values has to be calculated:

$$\sum_{j=1}^{16} Y_j = 53.46$$

After that the survival probability or selection probability  $SELPROB$  is calculated. For the solution in the third row of Table 4 we have:

$$SELPROB_3 = \frac{Y_3}{\sum_{j=1}^{16} Y_j} = \frac{2.32}{53.46} = 0.0434$$

Now, the cumulative probability  $CUMPROB$  can be calculated (step 8). For the third solution:

$$CUMPROB_3 = CUMPSOB_2 + SELPROB_3 = 0.1227 + 0.0434 = 0.1661$$

The calculated  $SELPROM_i$  and  $CUMPROB_i$  values are shown in Table 5.

Next in *Chapter 3* the roulette wheel selection (step 9) is described. It is applied to both parent and children solutions to randomly select 8 solutions for the next generation, therefore for this intermediate population of 16 solutions the selection operator would be performed only 8 times. It means that 8 times a random number *RAND* (in number interval ]0; 1]) is generated. Once *RAND*=0,1633, we can figure out that it points to the roulette wheel segment denoting the third solution (see Table 5). The table field *SELECTED* shows the number of times the solution has been selected for the next generation. If the *RAND* points to the selection which once has been already selected, then its value is saved in the field *RAND1* (or *RAND2*, *RAND3* etc.). The tenth solution is selected twice - at first with *RAND*=0,6090, but then with *RAND1* =0.6009. As none of solutions were selected more than twice, fields *RAND2*, *RAND3* ... *RAND7* are not shown in Table 5.

The selection of the fifth generation from parents and children

Table 5

NR	COUPLE	MEMBER	X1	X2	Y	SEL PROB	CUM PROB	RAND	RAND1	SELECTED
1	1	Child1	-0,25	0,05	2,97	0,0556	0,0556			0
2	1	Child2	0,05	-0,14	3,59	0,0672	0,1227			0
3	1	Mother	-0,37	0,05	2,32	0,0434	0,1661	0,1633		1
4	1	Father	0,05	0,05	3,90	0,0729	0,2391			0
5	2	Child1	0,03	0,08	3,86	0,0721	0,3113			0
6	2	Child2	-0,07	0,03	3,90	0,0729	0,3842	0,3556		1
7	2	Mother	0,11	0,01	3,77	0,0706	0,4548	0,4429		1
8	2	Father	-0,07	0,09	3,76	0,0703	0,5251	0,5186		1
9	3	Child1	0,14	0,11	3,39	0,0634	0,5885	0,5457		1
10	3	Child2	0,14	0,11	3,41	0,0638	0,6523	0,6090	0,6009	2
11	3	Mother	0,15	0,05	3,54	0,0662	0,7185			0
12	3	Father	0,11	0,20	3,05	0,0570	0,7755			0
13	4	Child1	0,15	0,20	2,90	0,0543	0,8298			0
14	4	Child2	0,12	0,20	3,04	0,0568	0,8867	0,8819		1
15	4	Mother	0,11	0,20	3,05	0,0570	0,9437			0
16	4	Father	0,15	0,18	3,01	0,0563	1,0000			0
			$\Sigma=$		53,46					

From the selected solutions the fourth generation is formed. It is shown in Table 6.

The fourth population

Table 6

X1	X2	Y	GENERATION
-0,07	0,03	3,90	4
0,14	0,11	3,39	4
0,14	0,11	3,41	4
-0,07	0,09	3,76	4
0,12	0,20	3,04	4
0,11	0,01	3,77	4
-0,37	0,05	2,32	4
0,14	0,11	3,41	4

Further in *Chapter 3* a series of experiments is described. "Stool leg" function, "waved bowl" function and "broken slope" function (see Fig. 3) were used as objective functions to search for the global maximum. The "stool leg" function can be described "with equations (I), (2) and (3):

$$f(x_1, x_2) = (2 + f_1(x_1, x_2)) \cdot (5 + 4 \cdot \Phi(f_1(x_1, x_2) - 2 + 0.1)) \quad (1)$$

$$\text{, where } \Phi(x) = \begin{cases} -1 & \forall x < 0 \\ 0 & x = 0 \\ 1 & \forall x > 0 \end{cases} \quad (2)$$

$$\text{and } f_1(x_1, x_2) = \cos(x_1 \cdot \frac{\pi}{3}) \cdot \cos(2 \cdot \pi \cdot x_1) + \cos(x_2 \cdot \frac{\pi}{3}) \cdot \cos(2 \cdot \pi \cdot x_2) \quad (3)$$

The parameters' ranges are:

$$-0.5 \leq x_1 \leq 0.5 \quad -0.5 \leq x_2 \leq 0.5$$

The "stool leg" function has a global maximum at point (0; 0) with function value 36. The global maximum is much higher than functions local maximums with the function value 3.5.

The "waved bowl" function is described with equation (4), but the parameter range is defined with inequalities (5):

$$f(x_1, x_2) = (x_1^2 + x_2^2)^{0.25} \cdot \left( \sin^2 \left( 50 \cdot (x_1^2 + x_2^2)^{0.1} \right) + 1 \right) \quad (4)$$

$$-0.75 \leq x_1 \leq 0.25 \quad -0.25 \leq x_2 \leq 0.75 \quad (5)$$

The "waved bowl" function has many local maximums and one region of global maximums. The Y value of global maximums is 1.874 , while in local maximums it is 1.586 and 1.325 respectively. Other local maximums have even lower function values.

The " broken slope" function which is depicted in Fig.3 case (c), can be obtained using expressions (6) and (7). The inequalities (8) describe the solution space:

$$f(x_1, x_2) = 9 + \left( 1 - \frac{x_1 + x_2}{2} \right) + (1 - (x_1 + x_2)) \cdot (-1 + \Phi(1 - (2 - (x_1 + x_2)))) \text{ , where } (6)$$

$$\Phi(x) = \begin{cases} -1 & \forall x < 0 \\ 0 & x = 0 \\ 1 & \forall x > 0 \end{cases} \quad (7)$$

$$-2.5 \leq x_1 \leq 1 \quad -2.5 \leq x_2 \leq 1 \quad (8)$$

The "broken slope" function global maximums have function value 9.4995 and they are located on the line where the "slope" is "broken". The peculiarity that is not easy noticeable from the picture of the function is that the incline of the one side of the slope is not the same as that of the other one.

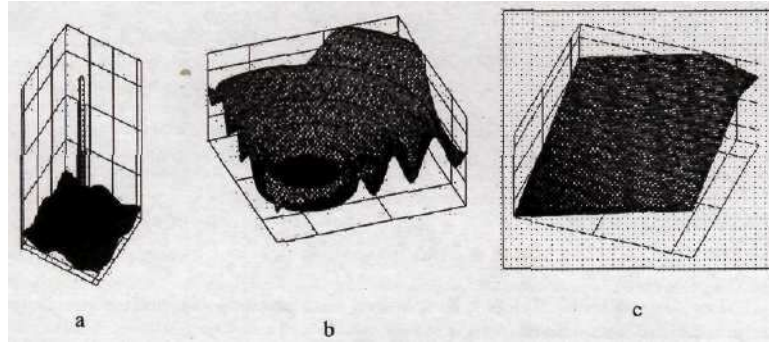


Fig. 3 Multimodal functions, which were used as objective functions for the experiment series

A series of experiments was performed with the following parameters: *Population\_Size* => 50, the terminating condition is *Loop\_Count* = 200, crossover probability is 1 and children number for one couple is 2 (every couple has two children), crossover type is "region-based crossover" in some series, but "diagonal crossover" in others, function type is "stool leg" ("waved bowl" or "broken slope" in other series) function, type of the algorithm is "multilayer genetic" (or "classical genetic" in other series). A series of 18 identical experiments (with different random numbers) was performed. Visual pictures from single experiments (of population displacement) and convergence curves were obtained. Also some statistical data (convergence curves showing mean and maximal fitness in 18 experiments) were obtained. Data were collected in FoxPro database tables.

Further in Chapter 3 the behaviour of the multilayer genetic algorithm and classical genetic algorithm in task of searching for the global maximum of "waved bowl" function are compared on the basis of a randomly chosen experiment. The population displacement was fixed by package SPSS (see Fig. 4).



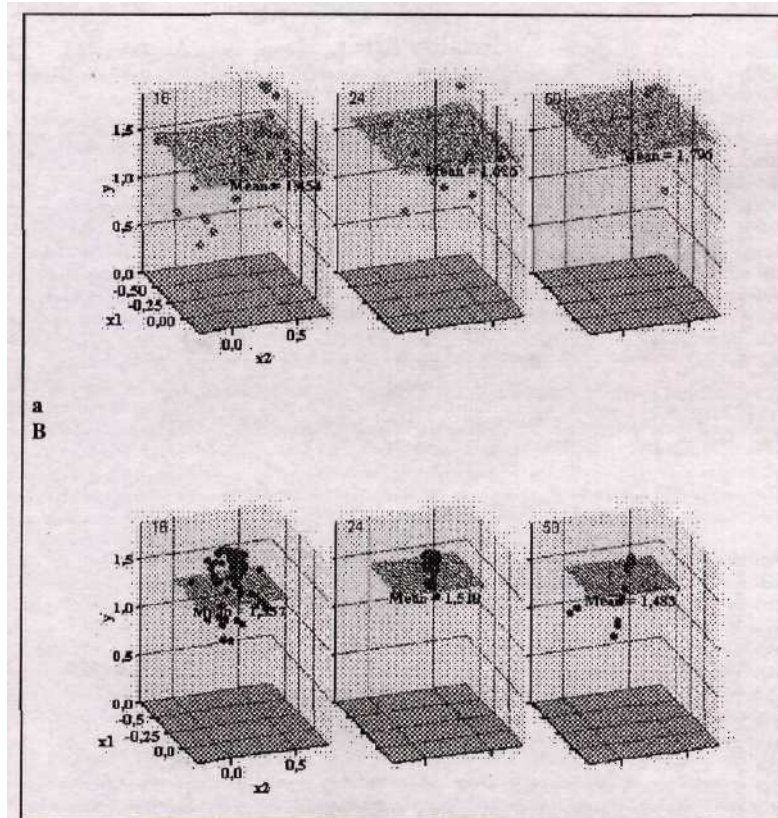


Fig. 4 The disposition of populations in 16th, 24th and 50th generation of multilayer genetic algorithm (a) and classical genetic algorithm (b) while searching for the maximum of "waved bowl" function which is depicted in Fig.3 (b). The level plane shows the mean of population fitness.

At the end of *Chapter 3* the conclusion is drawn that the multilayer genetic algorithm is more adaptive for optimisation of multimodal functions. The relative error (percentage of global maximum) while searching for the maximum of "waved bowl" function is shown in Fig.5 (a), but the case with "broken slope" function is depicted in Fig.5 (b).

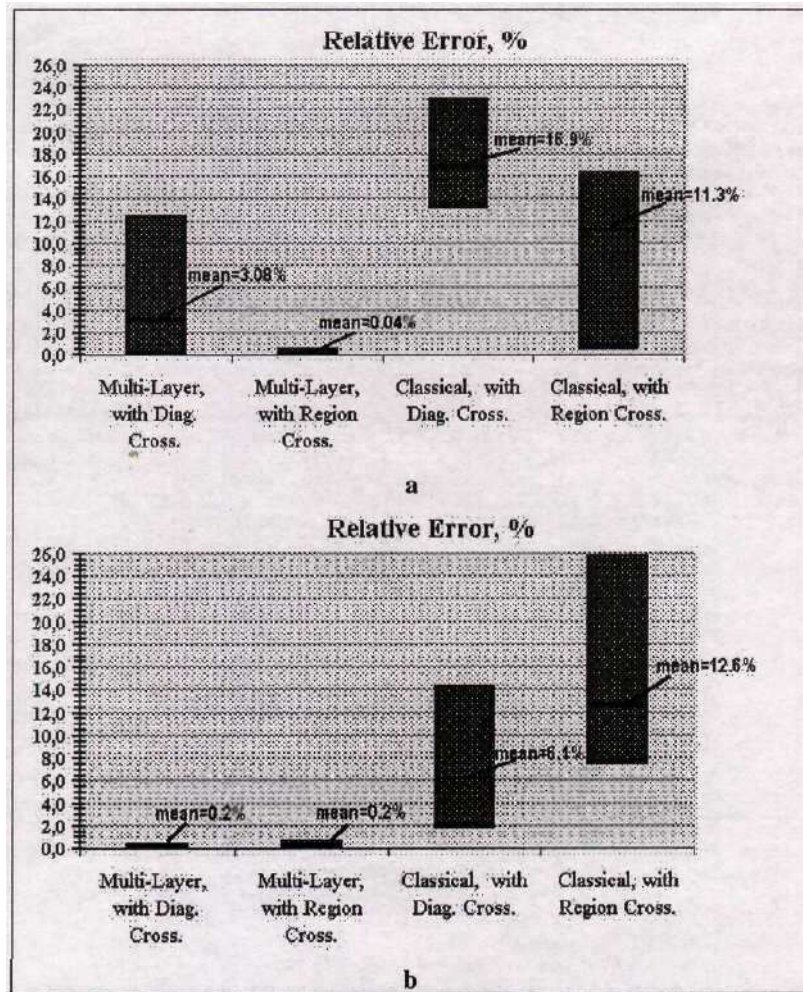


Fig. 5 The mean value of relative error (in 18 experiments) /or Multilayer and Classical algorithms while searching for the maximum of "waved bowl" function (a) and "broken slope" function (b)

At the end of Chapter 3 the effect of diagonal crossover on premature convergence is discussed.

*In Chapter 4* the restricted genetic algorithm is proposed to search for global maximums of functions, containing regions of global optimums. The problem of search in such tasks is explained.



In Chapter 4 the function depicted in Fig. 6 is proposed as a test function for genetic algorithms. In the research it is called "volcano-shape" function, because it has a shape of a volcano with a crater in its centre. It can be obtained using these expressions:

$$f(x_1, x_2) = -10 + F_B + (-10 + F_B) \cdot (-1 + 0.9 \cdot \Phi(-10 + F_B)) + 11 \quad , \text{ where} \quad (9)$$

$$F_B = 7.9 \cdot \sqrt{x_1^2 + x_2^2} \quad (10)$$

$$\Phi(x) = \begin{cases} -1 & \forall x < 0 \\ 0 & x = 0 \\ 1 & \forall x > 0 \end{cases} \quad (11)$$

The parameter space of the function is:

$$\begin{aligned} -2 &\leq x_1 \leq 2 \\ -2 &\leq x_2 \leq 2 \end{aligned}$$

In Chapter 4 the possible behaviour of the genetic algorithm, searching for the maximums of the volcano-shaped function, is analysed (see Fig 7.).

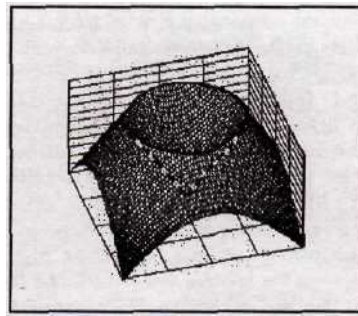


Fig. 6 Volcano-shape function

In a few single tests with classical genetic algorithm we can see the movement of population until at some moment (during certain generation) it is located in the inner part of the volcano, i.e. inside the crater (see Fig 7, case c). The movement away from the centre of the crater (see Fig 7, case c) in all four directions (which we expect), never happens. Classical genetic algorithm can move its population only in one direction (see Fig 7, case b).

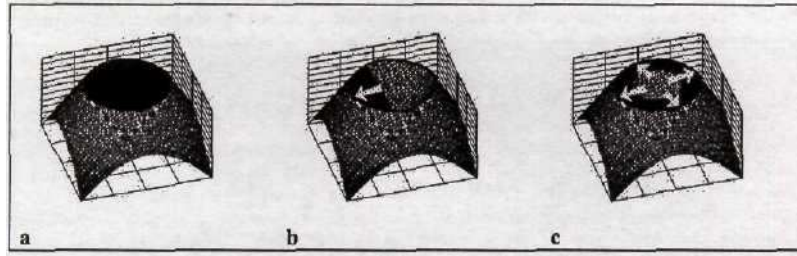


Fig. 7 The behaviour of classical genetic algorithm in the task of maximisation of volcano-shape function: location of its population during the first generations (a), shift to one side caused by asymmetry in distribution and premature convergence (b) and ideal behaviour (c)

In *Chapter 4* the author offers restricted genetic algorithm which is a model of a purposeful selection. Its main idea is borrowed from the social life models of humans - an individual creates children with other individual which is not casually chosen, but which is chosen according to certain qualities. The marriage within the certain nation is an illustrative example for such coupling.

The idea of the restricted genetic algorithm is the following: at first, the population is sorted in the ascending sequence of parameter  $x_1$ . Then it is divided into subpopulations with approximately same number of individuals. For example, if the population contains 200 solutions and we intend to divide the population into 5 subpopulations, then in the first subpopulation 40 solutions with the smallest  $x_1$  value would be included. The second subpopulation would contain the next 40 solutions with larger  $x_1$  values and so on. The fifth subpopulation would contain 40 solutions with the largest  $x_1$  values. Let us call the subpopulations "Races" and look at the illustration depicted in Fig. 11 (a). If the optimisation space is a two-parameter space, then we can divide every "Race" into smaller populations depending on  $x_2$  values. Suppose, we want to divide every "Race" into 3 smaller subpopulations. As  $40=13+13+14$ , we would get two subpopulations with 13 solutions and one with 14 solutions. Let us call the small subpopulations "Nations". At first we sort the solutions of the first "Race" in the ascending sequence of parameter  $x_2$ . Then we include the first 13 solutions (with the lowest  $x_2$  values) within the first "Race" into the first nation, the next solutions - to the second and so on. The last 13 plus remaining solutions belong to the last "Nation" of the first "Race" (see Fig. 11 a). Such dividing (or re-dividing) happens at the beginning of every interaction (it is part of forming a new generation). The crossover operator is only permitted within the "Nation" while the selection operator is performed on the whole population as if it was not divided.

Further in *Chapter 4* a description of the restricted genetic algorithm is presented. The restricted genetic algorithm contains the following steps:

1. generate initial population as set of random numbers (say, population of 200 solutions);
2. calculate the fitness value  $Y_i$  of each solution and its relative fitness (from the "floor"  $Y_0$ )  $Y_{Relative}=Y_i - Y_0$ , where  $Y_0 = \min (Y_j)$ ;

3. save the results;
4. divide the population into 5 Races by assigning the number of Race to every individual:
  - 4.1. sort the population in the ascending sequence of parameter  $X_1$  and to the first  $m$  solutions (say, to the first 40 solutions) assign  $RaceNumber=1$ ;
  - 4.2. to the next 40 solutions having higher values of  $X_1$  assign  $RaceNumber=2$ ;
  - 4.3. to the next 40 solutions assign  $RaceNumber=3$ ;
  - 4.4. to the next 40 solutions assign  $RaceNumber=4$ ;
  - 4.5. to the last 40 solutions assign  $RaceNumber=5$ ;
5. divide every Race into 3 Nations by assigning number of Nation to every individual:
  - 5.1. sort the population in the ascending sequence of parameter  $X_2$  and to the first  $m$  solutions (say, to the first 13 solutions) assign  $NationNumber=1$ ;
  - 5.2. to the next 13 solutions having higher values of  $X_2$  assign  $NationNumber=2$ ;
  - 5.3. to the last 14 solutions assign  $NationNumber=3$ ;
6. in cycle over all Races and all Nations: sort the solutions of the certain nation in the descending sequence of  $Y$ . Assign the couple identification numbers:  $CoupleNumber=1$  for the first two solutions,  $CoupleNumber=2$  for the next two and so on (coupling operator, used in Multilayer genetic algorithm);
7. cross-over (the region-based crossover operator depicted in Fig. 1) parents (solutions with the same couple number, belonging to the same Race and the same Nation) to obtain 2 children. In case of both parents being copies of the same solution, mutate the one of the parents and both children. The mutation is performed by re-generating one of the parameter values (chosen randomly) accordingly to uniform distribution in the whole range of the solution space;
8. calculate the fitness value  $Y_j$  of each solution and its relative fitness (from the "floor"  $Y_{Relative\ i} = Y_i / Y_0$ , where  $Y_0 = \min(Y_j)$ );
9. calculate survival probability  $SELPROB_i$  (with which it will be selected for the next generation) for every solution as
 
$$\frac{Y_{Relative\ i}}{\sum_j Y_{Relative\ j}};$$
10. calculate cumulative probability  $CUMPROB_i$  for  $i$ -th solution as  $CUMPROB_{i-1} + SELPROB_i$ ;
11. perform roulette wheel selection (Population\_Size times);
12. check if the final condition is met (for example, if 100 iterations has been performed). If it is not, then go back to step 3, otherwise end the procedure.

In Chapter 4 an example of the restricted genetic algorithm working with population of 24 solutions in the task of maximization of "volcanic" shape function is described. It is a detailed analysis of a real experiment having the following parameters:

The size of population  $Population\_Size=24$ , the terminating condition is  $Loop\_Count=12$ , the probability of crossover  $CrossoverRate=1$  and number of children in one family is 2 (every couple has two children), the type of crossover is

"Region- based", the type of the function is "Volcanic shape", the type of algorithm is "Restricted", number of Races  $Race\_Count=2$  and number of Nations  $Nation\_Count=2$ . The search space is delimited by inequalities:

$$-2.0 \leq x_1 \leq 2.0 \quad -2.0 \leq x_2 \leq 2.0$$

At first the initial population is generated as a set of solutions. Each parameter of every initial solution is generated as a random number in interval  $]-2; 2]$ . After that the function value Y is calculated. The initial population of the experiment is shown in Table 7.

The initial population

Table 7

NR	X1	X2	Y	NR	X1	X2	Y
1	-0,47	1,22	10,74	13	-1,56	0,41	8,55
2	-0,87	-1,15	9,72	14	0,42	1,30	10,30
3	0,00	0,23	3,64	15	-1,20	-1,30	7,40
4	1,42	0,64	8,93	16	1,66	1,90	2,08
5	0,25	-1,92	6,21	17	1,39	0,74	8,79
6	-0,19	1,13	10,18	18	1,09	-0,79	10,41
7	-1,38	0,25	10,05	19	-1,27	-1,17	7,74
8	-0,40	-0,91	9,05	20	1,16	-0,54	10,91
9	0,82	-0,14	7,90	21	-1,12	0,28	10,21
10	0,39	0,19	5,12	22	0,54	1,47	8,87
11	-1,51	-0,73	8,05	23	0,47	-1,27	10,36
12	-1,09	1,31	7,90	24	-0,92	1,52	7,38

The initial population has to be divided into subpopulations called "Races" (step 4 of the restricted algorithm). At first, the solutions are sorted in the ascending sequence of the parameter  $x_1$ . To the first 12 solutions the Race=1 is assigned, but to the next 12 solutions Race=2 is assigned. The population, divided into two races is shown in Table 8.

The initial population divided into Races

Table 8

NR	X1	X2	Y	RACE	NR	X1	X2	Y	RACE
13	-1,56	0,41	8,55	1	3	0,00	0,23	3,64	2
11	-1,51	-0,73	8,05	1	5	0,25	-1,92	6,21	2
7	-1,38	0,25	10,05	1	10	0,39	0,19	5,12	2
19	-1,27	-1,17	7,74	1	14	0,42	1,30	10,30	2
15	-1,20	-1,30	7,40	1	23	0,47	-1,27	10,36	2
21	-1,12	0,28	10,21	1	22	0,54	1,47	8,87	2
12	-1,09	1,31	7,90	1	9	0,82	-0,14	7,90	2
24	-0,92	1,52	7,38	1	18	1,09	-0,79	10,41	2
2	-0,87	-1,15	9,72	1	20	1,16	-0,54	10,91	2
1	-0,47	1,22	10,74	1	17	1,39	0,74	8,79	2
8	-0,40	-0,91	9,05	1	4	1,42	0,64	8,93	2
6	-0,19	1,13	10,18	1	16	1,66	1,90	2,08	2

Next the races are divided into nations (step 5 of the restricted algorithm). At first, the first race is sorted in the ascending sequence of the parameter  $X_i$  -The first 6 solutions obtain Nation=1, but the next six get Nation=2. After that the second race is sorted in the same way and again, the first six solutions are included in its first Nation,

but the -next six - in the second. The population, completely divided into subpopulations is depicted in Table 9.

The initial population divided into Races and Nations

Table 9

NR	X1	X2	Y	RACE	NATION	NR	X1	X2	Y	RACE	NATION
15	-1,20	-1,30	7,40	1	1	5	0,25	-1,92	6,21	2	1
19	-1,27	-1,17	7,74	1	1	23	0,47	-1,27	10,36	2	1
2	-0,87	-1,15	9,72	1	1	18	1,09	-0,79	10,41	2	1
8	-0,40	-0,91	9,05	1	1	20	1,16	-0,54	10,91	2	1
11	-1,51	-0,73	8,05	1	1	9	0,82	-0,14	7,90	2	1
7	-1,38	0,25	10,05	1	1	10	0,39	0,19	5,12	2	1
21	-1,12	0,28	10,21	1	2	3	0,00	0,23	3,64	2	2
13	-1,56	0,41	8,55	1	2	4	1,42	0,64	8,93	2	2
6	-0,19	1,13	10,18	1	2	17	1,39	0,74	8,79	2	2
1	-0,47	1,22	10,74	1	2	14	0,42	1,30	10,30	2	2
12	-1,09	1,31	7,90	1	2	22	0,54	1,47	8,87	2	2
24	-0,92	1,52	7,38	1	2	16	1,66	1,90	2,08	2	2

Then the coupling operator (step 6 of the restricted algorithm) follows. The solutions belonging to the same Race and same Nation are sorted in the descending sequence of the function value Y. The first two solutions are assigned Couple = 1, the next two are assigned Couple = 2 and so on. The initial population with assigned couple numbers is shown in Table 10.

Initial population, divided into couples

Table 10

NR	X1	X2	Y	RACE	NATION	COUPLE	NR	X1	X2	Y	RACE	NATION	COUPLE
7	-1,38	0,25	10,05	1	1	1	20	1,16	-0,54	10,91	2	1	1
2	-0,87	-1,15	9,72	1	1	1	18	1,09	-0,79	10,41	2	1	1
8	-0,40	-0,91	9,05	1	1	2	23	0,47	-1,27	10,36	2	1	2
11	-1,51	-0,73	8,05	1	1	2	9	0,82	-0,14	7,90	2	1	2
19	-1,27	-1,17	7,74	1	1	3	5	0,25	-1,92	6,21	2	1	3
15	-1,20	-1,30	7,40	1	1	3	10	0,39	0,19	5,12	2	1	3
1	-0,47	1,22	10,74	1	2	1	14	0,42	1,30	10,30	2	2	1
21	-1,12	0,28	10,21	1	2	1	4	1,42	0,64	8,93	2	2	1
6	-0,19	1,13	10,18	1	2	2	22	0,54	1,47	8,87	2	2	2
13	-1,56	0,41	8,55	1	2	2	17	1,39	0,74	8,79	2	2	2
12	-1,09	1,31	7,90	1	2	3	3	0,00	0,23	3,64	2	2	3
24	-0,92	1,52	7,38	1	2	3	16	1,66	1,90	2,08	2	2	3

The solution space and the initial population divided into Races, Nations and couples are depicted in Fig.8. The solutions, belonging to the second Race are black, other solutions belong to the first Race. The solutions belonging to the first nation are denoted with rings, while those belonging to the second one are denoted with triangles. The members of every couple are connected, therefore we can evaluate the possible location of their children. Larger rings (or triangles) have a higher fitness value Y.



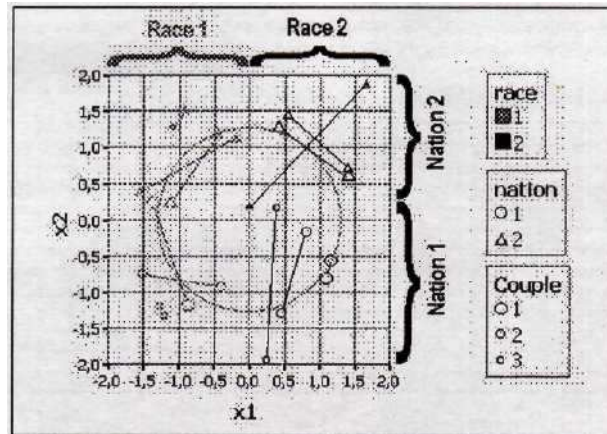


Fig. 8. The initial population - divided into Races, nations and couples.  
The circle marks the target line (of global maximums)

Next the cross-over operator is performed (step 7 of the restricted genetic algorithm). The region crossover is exactly the same which was used in Multilayer genetic algorithm. The parameters of the children  $x_1$  and  $x_2$  are generated as random numbers in the area delimited by parameter values of their parents. The only difference is that both parents in every couple have not only similar Y values, but also similar  $x_1$  and  $x_2$  values. The children with their parents are shown in Table 11.

Children generation from parents

Table 11

NR	Rn	Nn	COUPLE	MEMBER	X1	X2	Y	NR	Rn	Nn	COUPLE	MEMBER	X1	X2	Y
1	1	1	1	child_1	-0,99	-0,17	9,16	25	2	1	1	child_1	1,14	-0,61	10,82
2	1	1	1	child_2	-1,20	-0,06	10,52	26	2	1	1	child_2	1,14	-0,67	10,57
3	1	1	1	mother	-0,87	-1,15	9,72	27	2	1	1	mother	1,09	-0,79	10,41
4	1	1	1	father	-1,38	0,25	10,05	28	2	1	1	father	1,16	-0,54	10,91
5	1	1	2	child_1	-1,34	-0,87	8,68	29	2	1	2	child_1	0,52	-0,35	6,43
6	1	1	2	child_2	-0,56	-0,75	8,64	30	2	1	2	child_2	0,49	-1,00	9,93
7	1	1	2	mother	-0,40	-0,91	9,05	31	2	1	2	mother	0,82	-0,14	7,90
8	1	1	2	father	-1,51	-0,73	8,05	32	2	1	2	father	0,47	-1,27	10,36
9	1	1	3	child_1	-1,23	-1,17	7,92	33	2	1	3	child_1	0,35	-0,93	9,08
10	1	1	3	child_2	-1,26	-1,23	7,50	34	2	1	3	child_2	0,34	-1,77	7,19
11	1	1	3	mother	-1,20	-1,30	7,40	35	2	1	3	mother	0,25	-1,92	6,21
12	1	1	3	father	-1,27	-1,17	7,74	36	2	1	3	father	0,39	0,19	5,12
13	1	2	1	child_1	-0,55	1,05	10,41	37	2	2	1	child_1	0,47	0,69	7,95
14	1	2	1	child_2	-1,10	0,83	10,22	38	2	2	1	child_2	0,97	1,23	8,90
15	1	2	1	mother	-0,47	1,22	10,74	39	2	2	1	mother	1,42	0,64	8,93
16	1	2	1	father	-1,12	0,28	10,21	40	2	2	1	father	0,42	1,30	10,30
17	1	2	2	child_1	-1,12	0,90	9,77	41	2	2	2	child_1	0,66	1,33	9,45
18	1	2	2	child_2	-0,30	0,53	6,34	42	2	2	2	child_2	0,82	1,19	9,72
19	1	2	2	mother	-0,19	1,13	10,18	43	2	2	2	mother	1,39	0,74	8,79
20	1	2	2	father	-1,56	0,41	8,55	44	2	2	2	father	0,54	1,47	8,87
21	1	2	3	child_1	-1,03	1,50	7,03	45	2	2	3	child_1	0,89	1,02	10,38
22	1	2	3	child_2	-0,99	1,44	7,55	46	2	2	3	child_2	1,14	1,59	6,06
23	1	2	3	mother	-1,09	1,31	7,90	47	2	2	3	mother	0,00	0,23	3,64
24	1	2	3	father	-0,92	1,52	7,38	48	2	2	3	father	1,66	1,90	2,08

In the column titles of Table 11 the following text abbreviations are used: "Race" is replaced with Rn (Race number), but "Nation" is replaced with Nn (Nation number).

Next the fitness value  $Y$  and its relative value  $Y\_Relative$  for every solution is calculated (step 8 of the Restricted algorithm). The function value of volcano-shaped function can be calculated using expressions (9), (10) and (11). Then the minimal value of  $Y$  is found:  $Y\_min=2.08$  (it is 48<sup>01</sup> solution in table 11). From step 8 we remember that  $Y_0=Y\_min$ , therefore the "floor" of this population is 2.08. Now the relative fitness value of every solution can be obtained:

$$Y_{Relative\ i} = Y_i - Y_0$$

For example, for the third solution:

$$Y_{Relative\ 3} = Y_3 - Y_{min} = 9.72 - 2.08 = 7.64$$

The obtained  $Y\_Relative$ , values for all 48 solutions are shown in Table 12. The column title "Nr" denotes the identification number of the solution.

Values of  $Y\_relative$

Table 12

NR	Y RELATIVE	NR	Y RELATIVE	NR	Y RELATIVE	NR	Y RELATIVE
1	7.08	13	8.33	25	8.73	37	5.86
2	8.43	14	8.13	26	8.49	38	6.81
3	7.64	15	8.66	27	8.33	39	6.85
4	7.97	16	8.12	28	8.83	40	8.21
5	6.60	17	7.68	29	4.35	41	7.37
6	6.55	18	4.26	30	7.85	42	7.64
7	6.97	19	8.10	31	5.81	43	6.71
8	5.97	20	6.46	32	8.27	44	6.78
9	5.83	21	4.95	33	7.00	45	8.29
10	5.42	22	5.47	34	5.11	46	3.97
11	5.32	23	5.82	35	4.12	47	1.55
12	5.65	24	5.29	36	3.04	48	0.00

After that the sum of all  $Y$  relative is calculated, and  $\sum Y_{Relative\ i} = 310.67$ . The survival probability (probability to be selected for the next generation)  $SELPROB_i$  is calculated (see step 9 of the restricted genetic algorithm):

$$SELPROB_i = \frac{Y_{Relative\ i}}{\sum_{j=1}^{48} Y_{Relative\ j}}$$

For example, for the third solution the calculation (of  $SELPROB_3$ ) can be done using the equation in such a way:

$$SELPROB_3 = \frac{Y_{Relative\ 3}}{\sum_{j=1}^{48} Y_{Relative\ j}} = \frac{7.64}{310.67} = 0.0246$$

Then we can obtain the cumulative probability (see step 10 of the restricted genetic algorithm)  $CUMPROB_i$  for  $i$ -th solution as  $CUMPROB_{i-1} + SELPROB_i$ . Therefore for the third solution we have:

$$CUMPROB_3 = CUMPROB_2 + SELPROB_3 = 0.0499 + 0.0246 = 0.0745$$

The calculated values of survival probabilities and cumulative probabilities of all 48 solutions are given in Table 13.

Survival probabilities and cumulative probabilities Table 13

NR	SEL PROB	KUM PROB	NR	SEL PROB	KUM PROB
1	0,0228	0,0228	25	0,0281	0,5453
2	0,0271	0,0499	26	0,0273	0,5727
3	0,0246	0,0745	27	0,0268	0,5995
4	0,0256	0,1002	28	0,0284	0,6279
5	0,0212	0,1214	29	0,0140	0,6419
6	0,0211	0,1425	30	0,0253	0,6671
7	0,0224	0,1649	31	0,0187	0,6858
8	0,0192	0,1841	32	0,0266	0,7125
9	0,0188	0,2029	33	0,0225	0,7350
10	0,0174	0,2203	34	0,0164	0,7514
11	0,0171	0,2374	35	0,0133	0,7647
12	0,0182	0,2556	36	0,0098	0,7745
13	0,0268	0,2824	37	0,0189	0,7934
14	0,0262	0,3086	38	0,0219	0,8153
15	0,0279	0,3365	39	0,0220	0,8373
16	0,0262	0,3626	40	0,0264	0,8638
17	0,0247	0,3874	41	0,0237	0,8875
18	0,0137	0,4011	42	0,0246	0,9121
19	0,0261	0,4271	43	0,0216	0,9337
20	0,0208	0,4479	44	0,0218	0,9555
21	0,0159	0,4639	45	0,0267	0,9822
22	0,0176	0,4815	46	0,0128	0,9950
23	0,0187	0,5002	47	0,0050	1,0000
24	0,0170	0,5172	48	0,0000	1,0000

Next the roulette wheel selection is performed 24 times (see step 11 of the restricted genetic algorithm). The roulette-wheel selection is applied to both parent and children solution intermediate population. It means that 24 random numbers belonging to interval ]0; 1] are generated. Once  $RAND=0.1789$ , we can figure out that it points to the roulette wheel segment denoting the eight solution (see Table 14). When at some moment  $RAND=0.1836$ , pointing to the same solution is generated, it is saved in the field  $RAND1$ . As every solution in this task can be chosen 3,4, ...24 times, other fields  $RAND2$ ,  $RAND3$ ,  $RAND4$  etc. were used, but in this experiment only twice-selected cases appeared. The data obtained during the execution of the selection operator is collected in Table 14.



NR	X1	X2	Y	SEL PROB	KUMPROB	RAND	RAND1	SELECTED
1	-0,99	-0,17	9,16	0,0228	0,0228	0,0009	0,0007	2
2	-1,20	-0,06	10,52	0,0271	0,0499			0
3	-0,87	-1,15	9,72	0,0246	0,0745	0,0704		1
4	-1,38	0,25	10,05	0,0256	0,1002	0,0921		1
5	-1,34	-0,87	8,68	0,0212	0,1214			0
6	-0,56	-0,75	8,64	0,0211	0,1425			0
7	-0,40	-0,91	9,05	0,0224	0,1649			0
8	-1,51	-0,73	8,05	0,0192	0,1841	0,1789	0,1836	2
9	-1,23	-1,17	7,92	0,0188	0,2029			0
10	-1,26	-1,23	7,50	0,0174	0,2203	0,2048		1
11	-1,20	-1,30	7,40	0,0171	0,2374	0,2306		1
12	-1,27	-1,17	7,74	0,0182	0,2556			0
13	-0,55	1,05	10,41	0,0268	0,2824			0
14	-1,10	0,83	10,22	0,0262	0,3086	0,2862		1
15	-0,47	1,22	10,74	0,0279	0,3365			0
16	-1,12	0,28	10,21	0,0262	0,3626	0,3564		1
17	-1,12	0,90	9,77	0,0247	0,3874	0,3869		1
18	-0,30	0,53	6,34	0,0137	0,4011			0
19	-0,19	1,13	10,18	0,0261	0,4271			0
20	-1,56	0,41	8,55	0,0208	0,4479	0,4380		1
21	-1,03	1,50	7,03	0,0159	0,4639	0,4536		1
22	-0,99	1,44	7,55	0,0176	0,4815	0,4738		1
23	-1,09	1,31	7,90	0,0187	0,5002			0
24	-0,92	1,52	7,38	0,0170	0,5172			0
25	1,14	-0,61	10,82	0,0281	0,5453			0
26	1,14	-0,67	10,57	0,0273	0,5727			0
27	1,09	-0,79	10,41	0,0268	0,5995			0
28	1,16	-0,54	10,91	0,0284	0,6279	0,6275		1
29	0,52	-0,35	6,43	0,0140	0,6419	0,6361	0,6289	2
30	0,49	-1,00	9,93	0,0253	0,6671			0
31	0,82	-0,14	7,90	0,0187	0,6858			0
32	0,47	-1,27	10,36	0,0266	0,7125			0
33	0,35	-0,93	9,08	0,0225	0,7350	0,7320		1
34	0,34	-1,77	7,19	0,0164	0,7514			0
35	0,25	-1,92	6,21	0,0133	0,7647	0,7559		1
36	0,39	0,19	5,12	0,0098	0,7745	0,7687		1
37	0,47	0,69	7,95	0,0189	0,7934	0,7822		1
38	0,97	1,23	8,90	0,0219	0,8153			0
39	1,42	0,64	8,93	0,0220	0,8373			0
40	0,42	1,30	10,30	0,0264	0,8638	0,8410		1
41	0,66	1,33	9,45	0,0237	0,8875	0,8662		1
42	0,82	1,19	9,72	0,0246	0,9121			0
43	1,39	0,74	8,79	0,0216	0,9337			0
44	0,54	1,47	8,87	0,0218	0,9555			0
45	0,89	1,02	10,38	0,0267	0,9822	0,9816		1
46	1,14	1,59	6,06	0,0128	0,9950			0
47	0,00	0,23	3,64	0,0050	1,0000			0
48	1,66	1,90	2,08	0,0000	1,0000			0

The second population was obtained as a set of selected solutions and it is shown in Table 15.

The second population

Table 15

NR	X1	X2	Y	NR	X1	X2	Y
1	0,47	0,69	7,95	13	0,52	-0,35	6,43
2	0,42	1,30	10,30	14	-0,99	-0,17	9,16
3	-0,99	1,44	7,55	15	-1,10	0,83	10,22
4	-1,12	0,90	9,77	16	-1,26	-1,23	7,50
5	-1,38	0,25	10,05	17	0,66	1,33	9,45
6	0,39	0,19	5,12	18	0,89	1,02	10,38
7	-1,20	-1,30	7,40	19	-1,51	-0,73	8,05
8	-1,56	0,41	8,55	20	-1,03	1,50	7,03
9	-0,87	-1,15	9,72	21	0,35	-0,93	9,08
10	-1,12	0,28	10,21	22	0,25	-1,92	6,21
11	-1,51	-0,73	8,05	23	1,16	-0,54	10,91
12	0,52	-0,35	6,43	24	-0,99	-0,17	9,16

Next the new population is divided into Races, Nations and couples, and the obtained subpopulations are slightly moved (see Fig. 9) to one side.

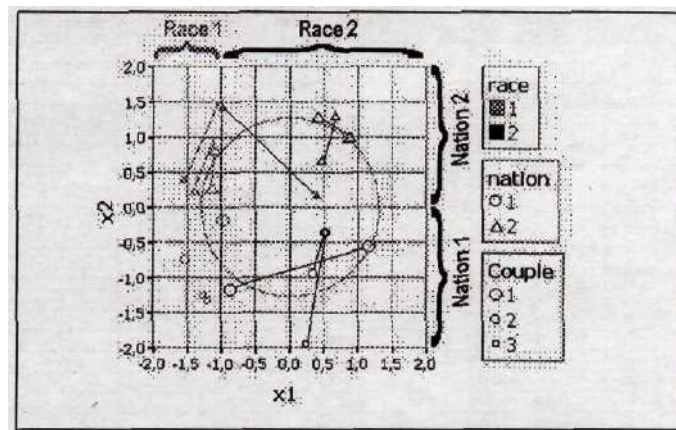


Fig. 9 The second population - divided into Races, Nations and couples

In Fig. 9 one can observe that in the second couple of the 1<sup>st</sup> Race and 1<sup>st</sup> Nation both parents coincide, i.e. they are copies of the same solution. That leads to the mutation of a randomly chosen "Mother's" parameter. In the same way the both children are mutated (the similar situation was described in the multilayer algorithm).

Next in *Chapter 4* the author describes a series of experiments. The maximization of "stool" function, "volcano-shape" function and "waved bowl" function are performed by means of the restricted genetic algorithm. The effectiveness

of the restricted genetic algorithm is compared with that of multilayer genetic algorithm and classical genetic algorithm.

The "stool" function has four global maximums (its solution space is enlarged now if we compare it with the first series of experiments). The "stool" function is depicted in Fig. 10 (see case a). Its cross-section with  $x_2 = 0$  is shown in Fig. 10 (b).

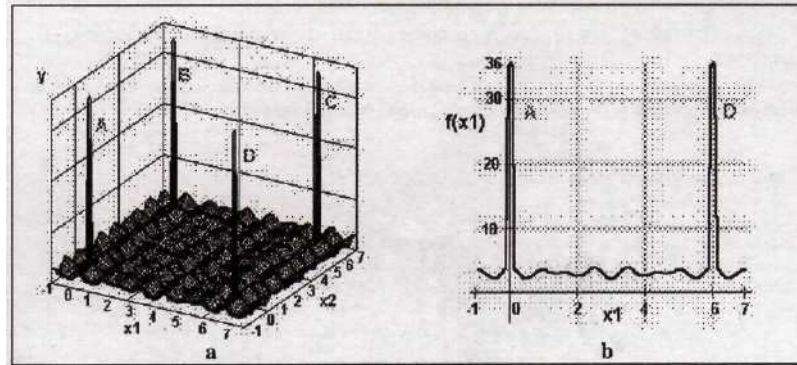


Fig. 10. The "stool" function which was used to test the uniformity of population clouds in classical genetic algorithm, multilayer genetic algorithm and restricted genetic algorithm: three-dimensional picture (a) and cross-section with  $x_2=0$  (b)

The "stool" function can be described with expressions (1), (2) and (3). The solution space parameters' range is:

$$-1 \leq x_1 \leq 7 \quad -1 \leq x_2 \leq 7$$

"Volcano-shaped" function has already been described by equations (9), (10) and (11) and it is depicted in Fig.6.

The solution space of the function in the series of experiments was:

$$-2 \leq x_1 \leq 2$$

$$-2 \leq x_2 \leq 2$$

The "waved bowl" function is depicted in Fig. 2(a) and it has already been described by equation (4). The solution space during the trial of experiments was the following:

$$-0.75 \leq x_1 \leq 0.75 \quad -0.75 \leq x_2 \leq 0.75$$

The main aim of the series of experiments was:

- 1) To verify that the restricted genetic algorithm is able to search in opposite directions, while the classical is not.
- 2) To verify that the restricted genetic algorithm has stronger uniformity within its population cloud than the classical one.

In Chapter 4 the results of experiments are presented. Both individual experiments with printouts of their population clouds and the convergence curves showing means of 18 similar experiments (with different random numbers) are displayed.

In Chapter 4 explanation of the restricted genetic algorithm behaviour in maximization task (of volcano-shape function) is given (see also Fig. 11).

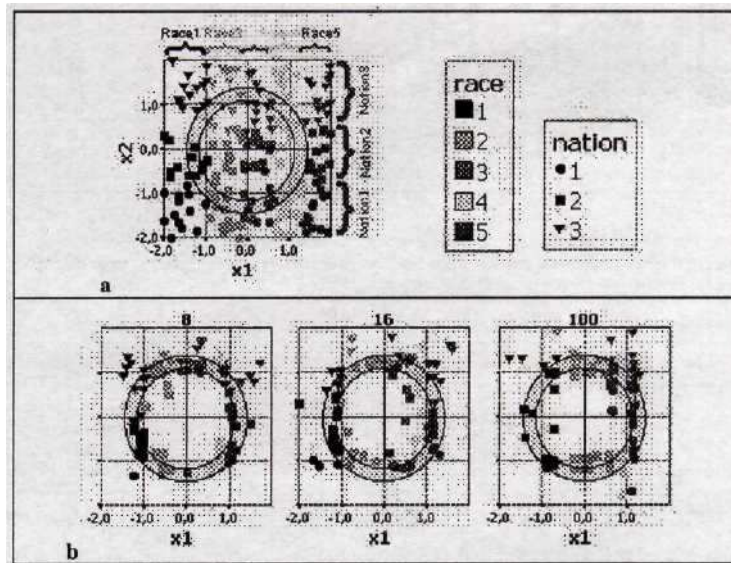


Fig 11. Fixed moments of the restricted genetic algorithm performance while maximizing "volcano-shaped" function: 1" (initial) population (a), 8th, 16th and 100th populations (b). The coloured ring is the target area.

In Chapter 4 the explanation of the behaviour of restricted genetic algorithm is given. The subpopulations of the initial population are approximately of the same size, therefore they cover approximately the same area. After the struggle for survival begins, subpopulations migrate to the areas of higher fitness, therefore the areas of lower fitness become deserted.

The series of experiments showed that restricted genetic algorithm can search in four different directions better than classical genetic algorithm or multilayer genetic algorithm.

*In Chapter 5* the author offers a new approach to multicriterial optimisation problems, and their solving by means of genetic algorithms. The aspects of search for Pareto set solutions are discussed. The maxi-min and maxi-max decision strategies are reviewed and their usage to optimisation with genetic algorithms is considered.

*In Chapter 5* the simultaneous maximization of two functions is discussed (see Fig 12 (a)). The functions can be calculated by the following equations:

$$Y = 18.5 + \frac{1}{2} \cdot F(x_1, x_2) + [-1 + 4 \cdot \text{sgn}(1 - (1 + F(x_1, x_2)))] \cdot F(x_1, x_2) \quad (12)$$

where

$$F(x_1, x_2) = 1 - (x_1 + x_2) \quad \text{or} \quad (13a)$$

$$F(x_1, x_2) = 1 - (x_1 + x_2 - 1) \quad (13b)$$

and

$$\text{sgn}(x) = \begin{cases} +1 & \forall x > 0 \\ 0 & x = 0 \\ -1 & \forall x < 0 \end{cases} \quad (14)$$

Here, equation (13a) must be used to calculate the first function (criterion) F1, but equation (13b) must be used to calculate the second function F2. The parameters' range is:

$$\begin{aligned} -1 &\leq x_1 \leq 3 \\ -1 &\leq x_2 \leq 3 \end{aligned}$$

As can be seen from the equations, the maximal function value is 18.5.

If we apply maxi-min decision strategy to the both functions, then the objective function at every point of the solution space can be obtained as minimal value of die both functions (see the cloud of initial population in Fig. 12 (b)). If maxi-max decision method is applied, then the value of the objective function can be obtained as the maximal value of the both functions (see the cloud of initial population in Fig. 12 (c)). If we use classical genetic algorithm to search for maximum of these new objective functions that it would find a single solution, but the solution found for objective function in Fig.12 (b) would never coincide with the one found for objective function in Fig. 12 (c). The question is - how to obtain several diverse solutions by means of genetic algorithm so that a decision maker who is not sure about his strategy (maxi-max or maxi-min) can obtain the best solutions for comparison and consideration?



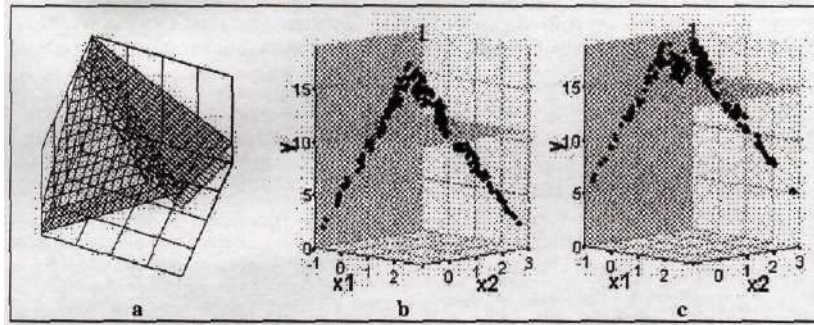


Fig 12. Two functions which were used for the task of simultaneous maximisation (a) and the initial populations showing objective functions if maxi-min decision method (b) or maxi-max method (c) is used. The level planes show the value of the mean fitness in each of the populations.

Further in Chapter 5 the multicriterial optimisation with genetic algorithm without usage of decision strategies is described. If one randomly chosen criterion is used as the objective function for some solutions while the other (randomly chosen) criterion is used as the objective function for the remaining solutions, then optimisation would proceed according to the values of both criteria (see Fig. 13).

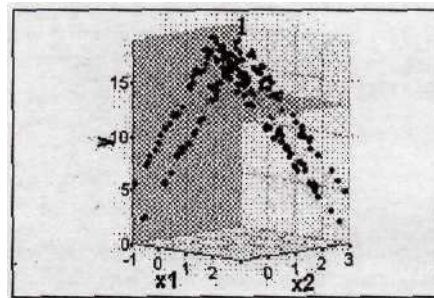


Fig. 13 The initial population for simultaneous optimisation of two criteria without usage of either maxi-min or maxi-max decision strategies. The level plane shows the value of the mean fitness of the population

The experiments in Chapter 5 prove that in such optimisation (without the decision strategies) the solutions belonging to the Pareto set are found. The Pareto set of the problem, depicted in the Fig. 13, is displayed in Fig. 14.

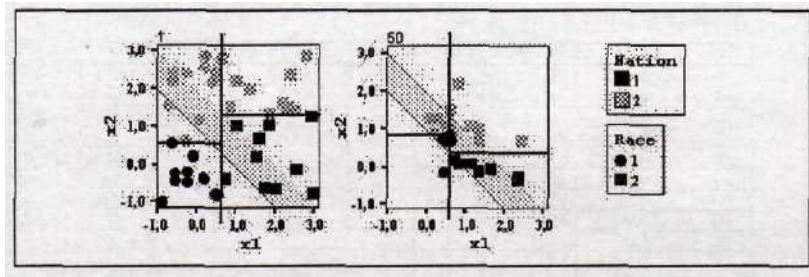


Fig. 14. Pareto set in (ask of two criteria maximization displayed in Fig 12 (a)). The 1<sup>st</sup> population and the 50th population of the restricted genetic algorithm when the subpopulation number is 4 (2 races and 2 nations in every race)

Next in *Chapter 5* an example with the restricted genetic algorithm maximising two criteria (see Fig. 2 (a)) is given. The parameters of the experiment were the following: the population size of the restricted genetic algorithm *Population\_Size*=24; the rate of crossover *Crossover\_Rate*=1; the terminating condition *Loop\_Count*=6; the number of races *Race\_Count*=2; the number of nations *Nation\_Count*=2; crossover type - *region - based crossover*, objective functions - two "broken slope" functions (see Fig.12 (a))

At first the initial population is generated as random numbers in interval [-1; 3]. After that both criteria values are calculated. Then, if a random number  $RAND \in [0; 1]$  is less or equal to 0.5, the fitness value is assigned equal to the value of the first criterion  $y=FUNC_1$ , otherwise it is assigned equal to the second one  $y=FUNC_2$ .

The initial population of the experiment is shown in Table 16.

						Table 16					
NR	X1	X2	FUNC 1	FUNC 2	Y	NR	X1	X2	FUNC 1	FUNC 2	Y
1	2.77	0.55	13.88	10.38	10.38	13	-0.10	1.29	14.87	17.83	17.83
2	-0.98	1.70	12.76	17.26	12.76	14	0.09	1.85	18.20	15.23	18.20
3	2.50	0.50	15.02	11.52	15.02	15	0.37	2.48	15.52	12.02	12.02
4	1.65	0.05	17.14	16.05	17.14	16	-0.99	1.92	13.66	18.16	13.66
5	-0.36	0.51	10.16	14.66	14.66	17	-0.33	-0.88	4.08	8.58	8.58
6	0.92	2.83	12.40	8.90	8.90	18	0.49	2.04	16.64	13.14	16.64
7	2.53	-0.79	17.32	15.92	15.92	19	1.68	2.30	11.58	8.08	8.08
8	2.79	0.45	14.17	10.67	10.67	20	0.20	1.80	18.46	15.03	18.46
9	1.56	1.11	16.16	12.66	16.16	21	-0.01	2.21	17.80	14.30	17.80
10	1.07	1.45	16.68	13.18	16.68	22	0.09	2.01	18.15	14.65	14.65
11	1.62	1.35	15.11	11.61	15.11	23	2.98	-0.88	18.14	14.64	18.14
12	2.44	1.74	10.87	7.37	10.87	24	-0.80	1.86	14.26	18.30	14.26

The process of dividing the initial population into races is shown in Table 17. The solutions are sorted in the ascending sequence of the parameter  $x_1$ . To the first 12 solutions the *Race*=1 is assigned, but to the next 12 solutions *Race*=2 is assigned (see Table 17).

The initial population divided into races					Table 17				
NR	X1	X2	Y	RACE	NR	X1	X2	Y	RACE
16	-0,99	1,92	13,66	1	6	0,92	2,83	8,90	2
2	-0,98	1,70	12,76	1	10	1,07	1,45	16,68	2
24	-0,80	1,86	14,26	1	9	1,56	1,11	16,16	2
5	-0,36	0,51	14,66	1	11	1,62	1,35	15,11	2
17	-0,33	-0,88	8,58	1	4	1,65	0,05	17,14	2
13	-0,10	1,29	17,83	1	19	1,68	2,30	8,08	2
21	-0,01	2,21	17,80	1	12	2,44	1,74	10,87	2
14	0,09	1,85	18,20	1	3	2,50	0,50	15,02	2
22	0,09	2,01	14,65	1	7	2,53	-0,79	15,92	2
20	0,20	1,80	18,46	1	1	2,77	0,55	10,38	2
15	0,37	2,48	12,02	1	8	2,79	0,45	10,67	2
18	0,49	2,04	16,64	1	23	2,98	-0,88	18,14	2

Next the initial population is divided into nations (according to the values of parameter  $x_2$ ). The initial population divided into races and nations is shown in Table 18.

The initial population divided into nations						Table 18					
NR	X1	X2	Y	RACE	NATION	NR	X1	X2	Y	RACE	NATION
17	-0,33	-0,88	8,58	1	1	23	2,98	-0,88	18,14	2	1
5	-0,36	0,51	14,66	1	1	7	2,53	-0,79	15,92	2	1
13	-0,10	1,29	17,83	1	1	4	1,65	0,05	17,14	2	1
2	-0,98	1,70	12,76	1	1	8	2,79	0,45	10,67	2	1
20	0,20	1,80	18,46	1	1	3	2,50	0,50	15,02	2	1
14	0,09	1,85	18,20	1	1	1	2,77	0,55	10,38	2	1
24	-0,80	1,86	14,26	1	2	9	1,56	1,11	16,16	2	2
16	-0,99	1,92	13,66	1	2	11	1,62	1,35	15,11	2	2
22	0,09	2,01	14,65	1	2	10	1,07	1,45	16,68	2	2
18	0,49	2,04	16,64	1	2	12	2,44	1,74	10,87	2	2
21	-0,01	2,21	17,80	1	2	19	1,68	2,30	8,08	2	2
15	0,37	2,48	12,02	1	2	6	0,92	2,83	8,90	2	2

After that the initial population is divided into couples (see Fig. 15) depending on Y value of every solution. The shaded diagonal region is the target area (of Pareto solutions).

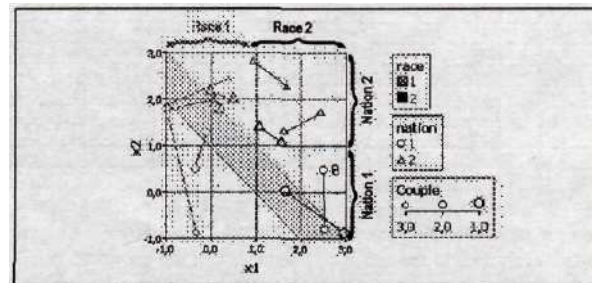


Fig. 15. The Initial population, divided into races, nations and couples



As we can see, the restricted algorithm works in the same way as it did in the case of the optimisation of only one criterion. The next four populations of the experiment, divided into races, nations and couples are depicted in Fig. 16.

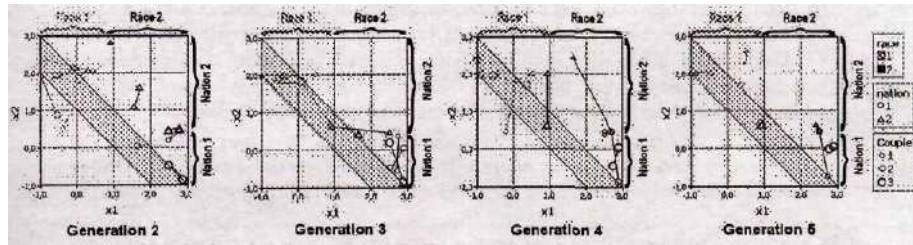


Fig. 16. The location of Populations, divided into races, nations and couples in the 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup> and 5<sup>th</sup> generations in simultaneous maximisation of two broken-slope functions with the restricted genetic algorithm

Next in Chapter 5 the usage of the restricted genetic algorithm in the optimisation of  $n$  criteria is described. The steps of this case of algorithm are similar to those when the optimisation of one criterion is performed. The differences in the description are marked by the shifted cursive:

- 1.generate the initial population as a set of random numbers (for example, 200 solutions);
- 2.generate a random number  $RAND \in [0; 1]$  for every solution. If  $(1/n) \cdot (m-1) \leq RAND \leq 1/n \cdot m$  where  $n$  is the number of criterion to be optimised and  $m$  is a natural number, then the value of  $m$  criterion is (randomly chosen) to be the fitness value for this solution;
- 3.calculate the fitness value  $Y_i$  of each solution (as value of the criterion chosen in step 2) and its relative fitness (from the "floor"  $Y_0$ )  $Y_{Relative} = Y_i - Y_0$  .where  $Y_0 = \min(Y_i)$ ;
4. save the results;
- 5.divide the population into 5 Races by assigning the number of Race to every individual:
  - 5.1.sort the population in the ascending sequence of parameter  $x_1$ .and to the first  $m$  solutions (say, to the first 40 solutions) assign RaceNumber=1;
  - 5.2.to the next 40 solutions having higher values of  $x_1$  assign RaceNumber=2;
  - 5.3.to the next 40 solutions assign RaceNumber=3;
  - 5.4.to the next 40 solutions assign RaceNumber=4;
  - 5.5.to the last 40 solutions assign RaceNumber=5;
- 6.divide every Race into 3 Nations by assigning number of Nation to every individual:
  - 6.1.sort the population in the ascending sequence of parameter  $x_2$ .and to the first  $m$  solutions (say, to the first 13 solutions) assign NationNumber=1;
  - 6.2.to the next 13 solutions having higher values of  $x_2$  assign NationNumber=2;

- 6.3. to the last 14 solutions assign NationNumber=5;
- 7.in cycle over all Races and all Nations: sort the solutions of the certain nation in the descending sequence of Y. Assign the couple identification numbers: CoupleNumber=1 for the first two solutions, CoupleNumber=2 for the next two and so on (Coupling operator);
- 8.cross-over (the region-based crossover operator depicted in Fig. 1) parents (solutions with the same couple number, belonging to the same Race and the same Nation) to obtain 2 children. In case of both parents being copies of the same solution, mutate one of the parents and both children. The mutation is performed by re-generating one of the parameter values (chosen randomly) accordingly to uniform distribution in the whole range of the solution space;
9. generate a random number  $RAND \in ]0; 1)$  for every solution. If  $(1/n) \cdot (m-1) \leq RAND \leq (1/n) \cdot m$  where  $n$  is the number of criterion to be optimised and  $m$  is a natural number, then the value of  $m$  criterion is (randomly chosen) to be the fitness value for this solution;
10. calculate the fitness value  $Y_i$  of each solution (as value of the criterion chosen in step 9) and its relative fitness (from the "floor"  $Y_0$ )  $Y_{Relative} = Y_i - Y_0$  .where  $Y_0 = \min(Y_j)$ ;
11. calculate survival probability SELPROB; (with which it will be selected for the next generation) for every solution as
- $$\frac{Y_{Relative\ i}}{\sum_j Y_{Relative\ j}} ;$$
12. to calculate cumulative probability  $CUMPROB_i$  for i-th solution as  $CUMPROB_{i-1} + SELPROB_i$ ;
13. to perform roulette wheel selection (Population\_Size times );
14. to check if the final condition is met (for example, if 100 iterations has been performed). If it is not, then go back to the step 4, otherwise end the procedure.

At the end of *Chapter 5* the serial experiments of multicriterial optimisation are considered. The task of simultaneous maximisation of two "broken slope" functions (see Fig. 12 (a)) in parameter region  $x_1 \in ] -1; 3]$  and  $x_2 \in ]-1; 3]$  with the classical genetic algorithm, multilayer genetic algorithm and the restricted genetic algorithm was performed. After that the task of simultaneous maximisation of two "volcano-shaped" functions (see Fig. 17 (a)) in parameter region  $x_1 \in ] -2.0; 2.0]$  and  $x_2 \in ]-2.0; 2.0]$  with the means of the same genetic algorithms.

To solve both tasks, Maxi-Min, Maxi-Max and no-strategy approaches were used.

The equations of the both "broken slope" functions were given above (see equations (12), (13a), (13b) and (14)).

The "volcano-shaped" functions are described by the following expressions:

$$Y = (-10 + F_B) \cdot ((-1) + 0.9 \cdot \text{sgn}(10 - F_B)) - 10 + F_B + 13 \quad (15)$$

$$\text{where} \quad F_B = 7.9 \cdot \sqrt{x_1^2 + x_2^2} \quad \text{or} \quad (16a)$$

$$F_B = 7.9 \cdot \sqrt{(x_1 + 0.25)^2 + x_2^2} \quad (16b)$$

$$\text{and} \quad (17)$$

$$\text{sgn}(x) = \begin{cases} +1 & \forall x > 0 \\ 0 & x = 0 \\ -1 & \forall x < 0 \end{cases}$$

The parameters' range is:

$$-2 \leq x_1 \leq 2 \quad -2 \leq x_2 \leq 2$$

Equations (16a) and (16b) both describe Fb , and expression (16a) appertains to the criterion F1, but expression (16b) appertains to the criterion F2 (see Fig. 17).

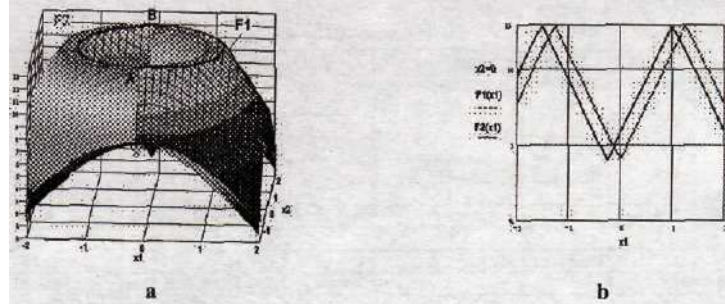


Fig.17. Two "volcano-shaped" criteria in the parameter space: three dimensional representation (a); cross-section with  $x_2=0$  (b)

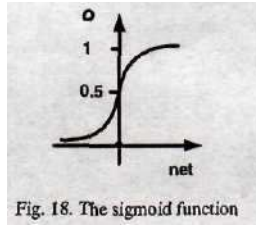
As can be seen from Fig. 17, the solution of the simultaneous maximisation of two "volcano-shaped" criteria problem are points A and B of the solution space, because there both criteria have the highest value. The experiments showed that Maxi-Min strategy is the most suitable to find the Pareto set solutions.

Chapter 6 describes the application of the genetic algorithms to the solution of a real optimisation problem.

The task is to find optimal weights for a neural network. Let's assume that a neural network having weights  $w_i$  for input signals  $x_i$  and output signal  $o$  and using sigmoid function as the activation function, has the following analytical expression to describe its output:

$$o = \frac{1}{1 + e^{-c \cdot net}} \quad net = w_0 + \sum_{i=1}^n w_i x_i \quad c = 12.5$$

The graph of the sigmoid function is shown in Fig. 18.



The neural network is supposed to solve the continuous XOR problem (on real numbers), and it is trained by using the solution set that is given in Table 19.

Input signal $x_1$	Input signal $x_2$	Supervisor's reply $d$	Output signal $o_5$	Error $E= d-o_5 $	Fitness $L=1-E$
0.2	0.5	0	1,000	1,000	0,000
0.5	0.2	0	0,453	0,453	0,547
-0.2	0.5	1	0,054	0,946	0,054
-0.5	0.2	1	0,926	0,074	0,926
-0.2	-0.5	0	1,000	1,000	0,000
-0.5	-0.2	0	1,000	1,000	0,000
0.2	-0.5	1	0,420	0,580	0,420
0.5	-0.2	1	0,420	0,580	0,420

min=0,000

The task is to maximize the minimal of all fitness values for each solution (see Table 19). The structure of the neural network is shown in Fig. 19.

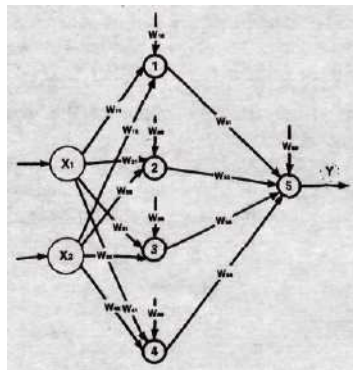


Fig. 19. The structure of the neural network solving continuous XOR problem

The structure of the neural network was transformed into a genotype of genetic algorithm, represented by a string of real parameters. The structure of the string is represented in Fig. 20.

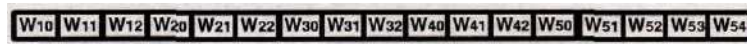


Fig. 20. The structure of the string

To accomplish the task, the Wright genetic algorithm, classical genetic algorithm, multilayer genetic algorithm and restricted genetic algorithm were used. From the experiments it can be seen that the multilayer genetic algorithm solves the problem in fewer iterations than the classical genetic algorithm, because in 18 experiments the mean fitness value of the population reached the value of global maximum in 50<sup>th</sup> generation for classical genetic algorithm, but in 30\* for the multilayer one (see Fig.21 case b and c). In the experiments with the Wright algorithm some increase in the value of the mean fitness value was observed even hi the 100<sup>th</sup> generation (see Fig. 21, a).

In experiments with the Wright algorithm 4 trials out of 18 didn't reach the global maxima (with its fitness value 1), but reached some solutions with fitness value 0.4, which, probably, is the local maxima of the fitness function. In all the 18 trials using this algorithm the mean value of the fitness was less than its maximal value, which can be explained by high rate of mutations (rate of mutations was 0.2). It also might show that the solutions found by this algorithm do not have high precision.

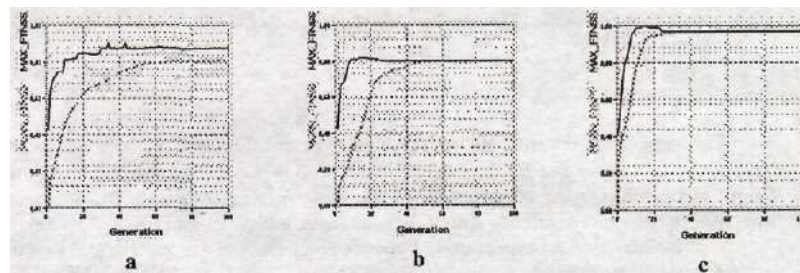


Fig. 21. The mean and maximal fitness values in 18 trials while optimising the weight vectors of the neural network shown in Fig. 1 9, using: a - the Wright algorithm, b - the classical genetic algorithm, c - the multilayer genetic algorithm

Experiments with the classical genetic algorithm showed that 6 trials out of 18 don't find the global maxima. In the successful trials the mean value of the fitness coincides with that of the global maxima and it proves that the best solution is found.

Experiments with the multilayer genetic algorithm were the most successful: only one trial out of 18 could not find the global maxima. Most of the experiments showed a rapid growth of the mean fitness value in the first 15 or 20 generations. Extremely rapid growth was observed in 5 trials when the global maxima was reached in the first 10 generations.

Similar experiments were performed with the restricted genetic algorithm (see Fig.22 a, b), but they were very time-consuming. Every trial lasted approximately 2.5 hours (for comparison - it took the multilayer algorithm 30 minutes, but the classical and Wright algorithm only 20 minutes to perform every single run). The terminating condition was the number of generations = 100. The population size was 800 in all the experiments, except for the restricted algorithm with population size 1600.

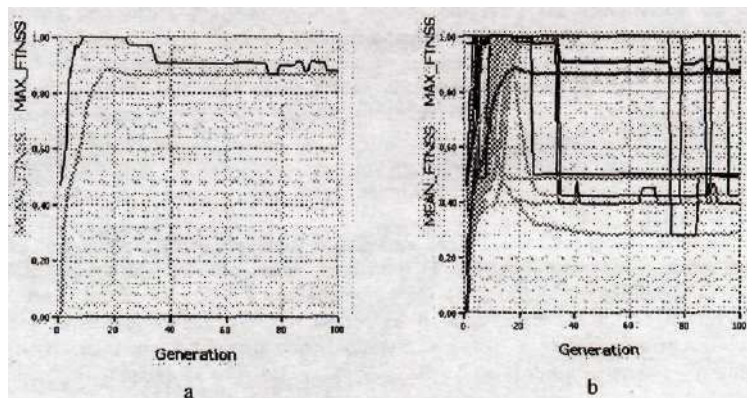


Fig. 22 The mean and maximal illness values in 18 trials while optimising the weight vectors of the neural network shown in the Fig.19 , using the restricted genetic algorithm: a - mean values of 18 trials, b - in every trial separately

The restricted genetic algorithm didn't reach the global maxima in 4 experiments of 18 (see Fig. 22, b), which is similar to the result obtained by using the Wright algorithm. On the other hand, we can observe a rapid growth of the mean fitness value, which is similar to that of the multilayer genetic algorithm.

Next in *Chapter 6* an application of genetic algorithms to a more complex neural network weight optimisation problem is demonstrated. In this application, the neural network is supposed to recognise hand-written figures "1", "2" and "3".

For the experiments, a set of 15 hand-written figures was used (see Fig. 23). Every figure is encoded as a set of 9 binary parameters. In the beginning, the recognition for all 10 figures was developed. A neural network with 9 inputs, 6 neurons in the first layer and 10 neurons in the second layer was created, but after the first runs the memory of the computer turned out to be too small and the processor speed too low for such a huge task. Therefore a new neural network was developed, consisting of 9 inputs, 5 neurons in the first layer and 3 neurons in the second layer (see Fig. 23).



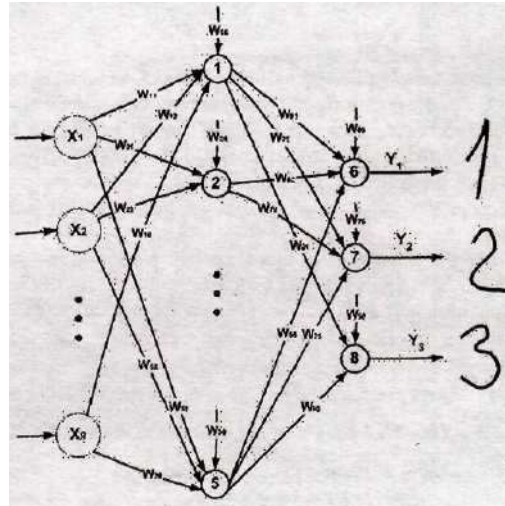


Fig. 23. The structure of the neural network solving recognition of hand-written figures "1", "2" and "3"

Then the data used for the neural network fitness calculations are shown (see Table 20).

Patterns for neural network fitness calculation Table 20

NR\X1	X2	X3	X4	X5	X6	X7	X8	X9	D1	D2	D3	Figure
1	1	0	0	1	0	0	0	0	1	0	0	1
2	0	1	1	1	0	1	1	1	0	1	0	2
3	0	1	1	0	1	1	0	1	1	0	0	3
4	0	0	1	1	0	0	0	0	1	0	0	1
5	0	1	1	0	0	1	0	1	1	0	1	2
6	0	1	1	0	1	1	1	1	1	0	0	3
7	0	0	0	1	0	0	0	0	0	1	0	1
8	0	1	1	0	0	1	1	1	1	0	1	2
9	0	1	1	0	1	1	1	1	1	0	0	3
10	1	0	1	1	0	0	0	0	0	1	0	1
11	0	1	1	1	0	1	0	1	1	0	1	2
12	0	1	1	0	1	1	0	1	1	0	0	3
13	0	0	1	1	0	0	0	0	0	1	0	1
14	0	1	1	0	0	1	1	1	1	0	1	2
15	0	1	1	0	1	1	0	1	1	0	0	3

As in the previous application, the activation function of this neural network is a sigmoid function (see Fig. 18). The structure of the real parameters string was similar to the one shown in the Fig. 20 with the exception that in these experiments the string consisted of 68 parameters (not only of 17). As the optimisation has to be carried out in the space of 68 parameters, various combinations of the parameters give us various values of the output signal. Suppose we have a training set consisting of patterns

shown in Table 20. Then the weighted input signal of every single neuron is calculated according to this equation

$$net = w_0 + \sum_{i=1}^n w_i x_i$$

As this neural network has 68 various values of weights, even the obtaining of the value 0.00001 in the output of this complicated neural network will be a very rare phenomenon. For example, there were 3820447 random solutions to be generated to obtain just 50 solutions with fitness larger than 0.00001. Therefore, the initial population was not generated as randomly generated solution space, but as randomly generated and after that selected as solutions which had their fitness values larger than 0.0001.

In *Chapter 6* the calculation of fitness for every solution is shown. The fitness of every combination of weights was calculated from the values of the errors. The error for the net output number  $I$  and training pattern number  $k$  can be calculated as

$$E_i^k = |d_i^k - o_i^k|, \text{ where } i = 1, 2, 3, \text{ but } k = 1, 2 \dots 15.$$

The fitness of every output of every neuron can be calculated as

$$L_i^k = 1 - E_i^k, \text{ where } i = 1, 2, 3, \text{ but } k = 1, 2 \dots 15$$

The fitness of the solution is calculated as follows:

$$L = \min(L_i^k), \text{ where } i = 1, 2, 3, \text{ but } k = 1, 2 \dots 15$$

From this equation we see that if at least one of the error values  $E$  was larger than 0.99999, then the fitness of the solution was less than 0.00001 and the solution was not included in the initial population. If any of the offspring had such a low fitness value, then it had some chances to survive.

In *Chapter 6* the results of the experiments are shown (see Fig. 24).

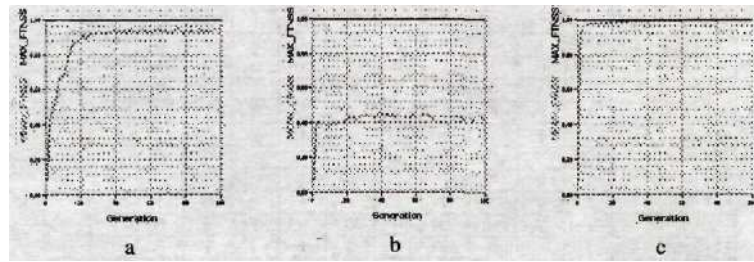


Fig 24. Recognition of hand-written figures "1", "2" and "3" - the mean and maximal fitness values in 10 trials: a - Wright genetic algorithm, b - classical genetic algorithm and c - multilayer genetic algorithm.

From the fitness values depicted in Fig. 24 it can be seen that multilayer genetic algorithm has found a solution in all 10 experiments in much fewer generations than Wright genetic algorithm (see Fig 24 cases a and c). The classical genetic algorithm showed premature convergence in 9 trials from 10 by concentrating the population around solutions which didn't have maximal fitness (see Fig 24, case b). We can also



see that in this 68 dimensions solution space Wright genetic algorithm is more effective than the classical genetic algorithm, which is an opposite feature to that we observed in 2-parameter solution space, because we can obtain a larger variety of solutions via recombination of 68 parameters than with recombination of just 2 parameters.

The fitness values of the first 5 trials for each algorithm are depicted in Fig.25.

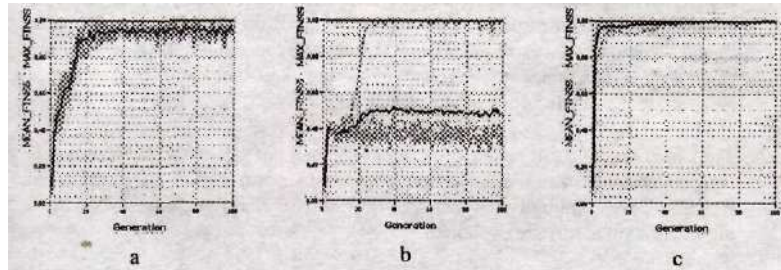


Fig 25. Recognition of hand-written figures "1", "2" and "3" - the mean and maximal fitness values in (he first 5 trials: a - Wright genetic algorithm, b - classical genetic algorithm and c - multilayer genetic algorithm.

The restricted genetic algorithm was not used in this application of figure recognition, because it is too time-consuming. On the other hand, the performance of multilayer algorithm was so good, that it didn't require searching in separate subpopulations. We know that search in separate subpopulations will certainly increase time, but we also must be aware that it may worsen the result (see Fig 22 of the previous application).

Chapter 7 contains the conclusions and the main advantages and disadvantages of the offered genetic algorithms. Also the perspective of the further research is given.

The appendix provides software printout and technical documentation to repeat experiments with the described algorithms. The software was developed in the FoxPro 2.6 for Dos.

## THE MAIN RESULTS OF THE WORK

The results of the doctoral research are the following:

1. The usage of genetic algorithms in shaping new "creative" solutions for a multicriterial problem is tested experimentally. The inability of the genetic algorithm to generate "bad" solutions (if required) on the basic training with "good" solutions is formulated and theoretically considered. The software "Genalg" is developed to generate new "creative" solutions by means of genetic algorithms for the specific object field.
2. The phenomenon of premature convergence of genetic algorithms is formulated and examined. Particular cases of optimisation of multimodal functions are explored and reasons of premature convergence of genetic algorithms (in them) are explained.
3. A new genetic algorithm is proposed (restricted genetic algorithm) which is very suitable for multimodal function optimisation. The behaviour of the restricted genetic algorithm is explored and compared to that of the classical genetic algorithm experimentally. The difference in the behaviour of the both algorithms and its reasons are considered theoretically.
4. The drawbacks of optimisation of functions which contain regions of global optimums are studied. Experiments to achieve uniform distribution of diverse solutions within the population were performed. An approach to the usage of the restricted genetic algorithm as more effective for functions which contain regions of global optimums is offered. A new test function to verify the ability to search for diverse solutions is suggested (volcano-shaped function).
5. The ability of genetic algorithms to search for the Pareto optimal solutions is examined. The phenomenon of uneven density of population of the classical genetic algorithm while searching for Pareto optimal solutions is observed. An approach to the usage of the restricted genetic algorithm as more effective for multicriterial optimisation is offered. New test functions are proposed.
6. The software "Rgenalg" is developed to work with real genetic algorithms (classical, multilayer, restricted genetic algorithms). It is usable for optimisation of one criterion or for simultaneous optimisation of two criteria.
7. An application of the restricted genetic algorithm to optimise the weights of a neural network (17 parameters), which is supposed to solve the XOR problem on real numbers is proposed.

The following papers related to the subject of this research were published in the referred editions:

### **Publications**

1. I. Lovtsova and A Takahashi (2003). Investigation of the crossover operators in genetic algorithm. *Scientific Proceedings of Riga Technical University (Information Technology and Management Science), Series 5, Volume 14*, Riga, P.104-112.
2. A. Takahashi, A. Borisov (2001). Decision Strategies in Evolutionary Optimization. *Computational Intelligence. Theory and Applications. Int.Conference, Proc. 17th Fuzzy Days*, Dortmund, Germany, October 1-3, 2001. Bernd Reusch (ed.): Springer, (Lecture Notes in Computer Science: Vol.2206), P.345-356.
3. A. Takahashi, A. Borisov (2001). Investigation of Decision strategies in Evolutionary Optimization. *Proceedings of SCAI'01 (Seventh Scandinavian Conference on Artificial Intelligence)*, University of Aarhus, Denmark, P. 123-132.
4. A.Takahashi and A.Borisov. (2001). Genetic Algorithm Behaviour in the Task of Finding the Multi-Maximum Domain. *Research and Development in Intelligent Systems XVII. Proc. of ES2000, the Twentieth SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, Cambridge, December 11th-13th, 2000. M. Bramer, A.Preece and RCoehenen (eds.) Springer-Verlag London, P. 185-198.
5. A. Takahashi (2000). Using genetic algorithms to search for maximum of the multimodal function. *Proceedings of The Second International conference "Simulation, Gaming, Training and Business Process Reengineering in Operations"* RTU September 8-9, P. 60-65
6. A-Takahashi (2000). Genetic Algorithm Behaviour in the Multiobjective Optimisation Task. *Proceedings of "KES'2000", Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies*, University of Brighton, UK, 30, 31 August, 1 September, P. 514-519
7. A. Takahashi (2000). Using Genetic Algorithms to Search for Maximum of the Multimodal Function. *Proceedings of MENDEL2000. (6<sup>th</sup> International Conference on Soft Computing, Brno, Czech Republic, June 7-9, 2000)*. Brno University of Technology, Faculty of Mechanical Engineering, P. 159-164.
8. A.Takahashi, A.Borisov (2000). Genetic Algorithm Behavior in the Optimization Task. *Lecture Notes of Nordic-Baltic-NW Russian Summer School NBR'2000, Applied Computation Intelligence to Engineering and Business*, St Petersburg, Russia June 4-11, 2000. K. Wang, J. Grundspenkis, and A. Yerofeyev (eds.) RTU, Riga 2001, P. 116-139
9. A. Takahši (2000). Ģenētiskie algoritmi reālu parametru optimizēšanai. *"Baltijas Reģiona Valstu Integrācijas Problēmas Ceļā uz Eiropas Savienību", starptautiskās konferences materiāli*. Rēzekne, Latvija. 2000.gada 2.-3.marts, 197-203 lpp
10. A. Takahashi, A. Borisov (1999). Generation of Objects and Decision Rules on the Basis of Genetic Algorithms. *Proceedings of EUROFUSE - SIC '99. (The Fourth Meeting of the EURO Working Group on Fuzzy Sets and The Second International Conference on Soft and Intelligent Computing. Budapest, Hungary May 25-28, 1999)*. Edited by B.De Baets, J.Fodor, L.T. Kozcy, P. 491-495.

11. A- Takahashi, A. Borisov (1999). Determination of Decision Rules on the Basis of Genetic Algorithms. // *Computational Intelligence : Theory and Applications; International Conference; Proceedings/ 6th Fuzzy Days*, Dortmund, Germany, May 25-28, 1999. Bernd Reusch (ed.): Springer, (Lecture notes in computer science: VoL 1625), P. 40-46.

#### **Conference theses**

1. A. Takahasi (1999). Determination of Decision Rules on the Basis of Genetic Algorithms. *The 2nd International Conference Environment. Technology. Resources, June 25-27, 1999.*, Rezeknes augstskola 1999-gada 25. - 27. jūnijs. Lpp. 205-206.
2. A. Takahaši (1999). Lemšanas likumu noteikšana, izmantojot ģenētiskos algoritmus. 40. RTU studentu zinātniskās un tehniskās konferences materiāli 1999. gada 26. - 30. aprīlī, Lpp.71.