REDUNDANCY OF GENOTYPES AS THE WAY FOR SOME ADVANCED OPERATORS IN EVOLUTIONARY ALGORITHMS – SIMULATION STUDY

The paper presents the model aimed to study effects of some advanced evolutionary operators. In the analogy to biological organisms, chromosomes contain two sets of genes. The first is a set of so-called *phenotype* genes, that is, the genes which are expressed in the organism's phenotype. The second set consists of the genes which are *redundant*, they are called also *latent* genes, because they do not influence on the organism's phenotype. This partition is not constant during organisms evolution, genes can change their role. The model uses a non-binary coding. Some advanced evolutionary operators, as neutral mutation and macromutation are proposed and studied. A possibility of achieving a new fitness niche by an evolving population – escaping from the local optimum – is the main interest of the paper. Preliminary results of the model's simulations are described and briefly discussed.

Introduction

The day when Darwin and Wallace presented their theories on evolution of biological organisms was the day of revolution in biology. Recently, Neo-Darwinism is a generally accepted evolutionary paradigm [5]. It considers, that the history of life can be explained by statistical processes operating within populations and species: reproduction, mutation, competition and selection.

Reproduction is the necessary process for species to survive, but potential abilities of species to reproduce are so large, that the size of population would exponentially increase if all individuals could reproduce with success. Mutation guarantees the positive entropy of a biological system. Because of limitation of the environment of an evolving population, there is a competition between individuals. The outcome of selection is elimination of some individuals due to competition: only a part of individuals can survive and have offsprings. Phenotype diversity is a consequence of recombinations and errors in genome transcription. The selection operates on diversified phenotypes. The Neo-Darwinian paradigm asserts that, the only force of evolution is natural selection [8].

The living organism can be perceived dually: as its *genotype* – encoded genes, and its *phenotype* – the way of response and morphology of an organism (the expressible feature of the organism). Each expressed feature of an individual which influence its fitness is called a *phene* (in analogy to a *gene*), and the set of such features of an individual is called its *phenotype*. We can specify two spaces: the n-dimensional genotype space $\boldsymbol{G}$ and the m-dimensional phenotype space $\boldsymbol{P}$, which represents the expressed features of organisms' genotypes (usually $n>>m$). We can imagine that over the phenotype space, the quality surface $\boldsymbol{Q}$ (i.e., fitness) is spread. Each point in the multidimensional space $\boldsymbol{G}$ (i.e., each genotype) corresponds exactly to one point in the multidimensional space $\boldsymbol{P}$ (its phenotype), but different genotypes can correspond to the same phenotype. Further, to each phenotype a *quality* value, also called a *fitness* value, is assigned. The organisms' ability to survive and to give offsprings depends on their fitness values.

The modeling and simulation of biological evolution are a good way for better understanding a tempo and nature of the evolution [5]. Another reason for analyzing evolution is its natural adaptation ability. This feature of evolution can be imitated by artificial systems. Many real problems can be presented as optimization problems if the goal of optimization is stated as the gradual improvement of solutions but not the convergence to the best. This is more 'human like' optimization.

The paper presents the model which is labeled *K-Model*. It does not use a binary alphabet, the genes are integer values. Additionally, the pleiotropy and polygene effects are considered, and the redundant genes are assumed and used for modeling neutral mutations and macromutations.

Preliminary results of simulated evolution using this model, and the effects of some advanced operators are discussed.

1. An outline of a classic genetic algorithm

Computer programs based on a genetic algorithm use the strategy of natural selection to achieve their aims [6]. The following steps are required for using a classic genetic algorithm (the two first steps have to be done by the user manually, the rest can be done automatically):

Step 1: Precisely setting the problem (what must be optimized, the scope of parameters, constraints, etc.).

Step 2: Defining the problem in terms of genetic algorithm: coding a set of potential solutions into bits strings (that is, defining genotypes of evolving individuals) and the fitness function, taking into account all constraints of the problem. This phase is often the most difficult.

Step 3: Creation of an initial population (a set of individuals). It can be done by random choice. Now, the evolution can proceed.

Step 4: Evaluation of each individual in the current population.

Step 5: Checking if the satisfactory solution exists in the population. If *yes*, then stop the run. If *no*, the next steps must be performed.

Step 6: Creation a new generation of the population:

a) Selection individuals for the reproduction (according to the assumed selection method).

b) Reproduction of individuals. Production offsprings with crossover and mutation operators.

Step 7: Go to the step 4.

2. Essential differences between the classic genetic algorithm and biological evolution

Genetic algorithms use a vocabulary borrowed from the genetics, their performance is similar to the biological populations' evolution [2]. New solutions (offsprings) are produced from the knowing ones (parents) using the genetic operators, such a crossover, mutation, inversion etc. Biological evolution is only an inspiration for GAs, in spite of their simplicity GAs are often sufficient to solve an intended task. Sometimes, for hard GAs cannot find satisfactory solution [3]. Therefore, it seems to be reasonable to search some other operators which can be useful in artificial genetic (evolutionary) algorithms. A nature can be a good place for such searching.

A tempo of biological evolution is still a severe problem for evolutionists, and it still stimulates hot discussions. In a long perspective, the evolution of species goes with a different tempo and in different directions [5]. The Neo-Darwinian paradigm assumes that both, a rate and a direction of populations' evolution are changed when a new fitness niche is gained. Macroevolution is described as the process with macromutations, which give the radical phenotype effect [9]. Macromutations are present in the nature. Dawkins says, that during the history of life, there could exist some sudden, step changes, singular macromutations, and in the effect, the child is essentially different from its parents [4].

Building the *K*-Model we have taken into account a mechanism which imitates such phenomena, as macromutations.

2.1. Neutral mutations and redundant genes

It is well known that only a small part of biological organisms' genotypes encodes the expressible features of the organisms. We call such genes *active* genes or *phenotype* genes. The rest can be seen as redundant genes, because they do not influence on the phenotype of the organism. Such genes are called *redundant* or *latent* genes. Evolution does not produce the perfect set of genes,

we can say that most part of a genome is a kind of "wast" [5]. These genes, as a part of chromosomes, are the subject of genetic operators, such mutation and crossover. Therefore, more changes in a genome, at least in the first period, are neutral, or potentially destructive. When the individual is preferred due to its "expressed" features, the destructive (but redundant) gene can "convey" itself on this individual for a long period. The partition of a genotype between the active genes and the redundant ones, is not fixed. As an effect of some processes (for example, a reorganization of chromosomes), a redundant gene can become the expressible one, and, from such a moment, this gene influences the organism's phenotype. So, it starts to be the active gene.

We believe that nature acts efficiency. Therefore, the following question can arise: Why redundant genes are not considered in an artificial genetic (or widely, evolutionary) algorithms? What benefit can we have from redundant genes?

Presented model includes redundant genes as well as some operators which make these genes useful (see section 3).

2.2. Pleiotropy and polygene effect

Usually GAs use a simple relation: *one gene – one phene*, what means that each gene is directly a coded parameter of an optimized function. In biology, a single gene of individual can have an impact, at the same time, on several phenotype features. Such an effect is called a *pleiotropy*. On the other hand, each single phenotype's feature of an individual (its phene) can be determined by simultaneous influence of a number of genes – this effect is called a *polygene*.

A model without above effects, as it is in a classic genetic algorithm, is a great simplification. The paper [10] presents a comparison of three models: a classic GA; a GA with small pleiotropy and polygene effects, and real coded genes; and the third – GA acting directly on the parameters of optimized function (without a genotype level). Simulations show that the tempo and mode of evolution (number of generations needed to achieve a higher fitness peak, and the diversity of populations) depend on the used model, and evolution goes better with pleiotropy and polygene effects. Presented in this paper the *K-Model* includes these effects. Fig.1 illustrates a possible representation of an individual (a chromosome) with pleiotropy and polygene effects and with redundant genes. This exemplary individual consists of eight phenotype genes ($g_1 \div g_8$), four redundant genes ($g_9 \div g_{12}$), and five phenes ($p_1 \div p_5$) which are the parameters of an optimized function **Q**. It is seen that gene g_3 has an impact on the second and fourth phenes, gene g_6 has an impact on the second and third phenes. Also, phenes p_1, p_2 , p_4 and p_5 depend of some numbers of genes.

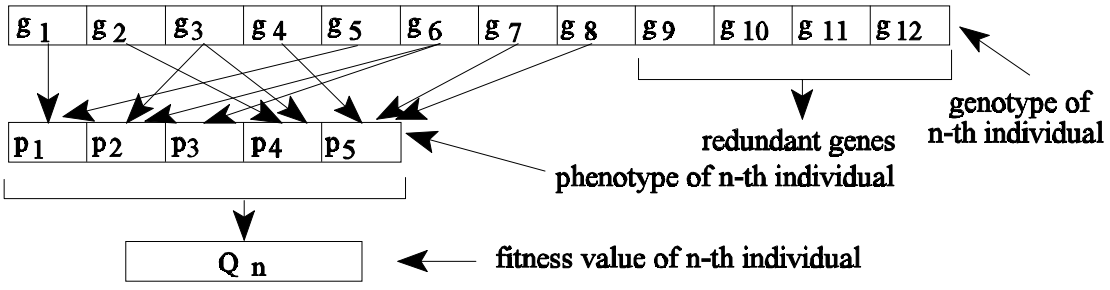

Figure 1. The representation of individuals with pleiotropy and polygene effects and with redundant genes

Such representation allows to model the process called *epistatic*, that is one gene, called *epistatic* one, can suppress the phenotype expression of another gene, called *a hypostatic* gene [1]. The dependency between phenotype genes and the phenes for each individual in the population can be expressed in the linear form:

$$|\boldsymbol{P}|_{1\times m} = |\boldsymbol{G}|_{1\times n} \times |\boldsymbol{A}|_{n\times m}, \quad (1)$$

where $|\boldsymbol{P}|$ – a vector of m phenes,
$|\boldsymbol{G}|$ – a vector of n phenotype genes,
$|\boldsymbol{A}|$ – a matrix characterizing the influence of genes on the phenes. The elements a_{ij} are real values.

Fig. 2 allows to compare the representation of individuals in classic genetic algorithms, where each gene is a bit (*zero* or *one*), some number of the first bits code the first parameter of optimized function, the next bits of chromosome code the second parameter (and so on, for more parameters of the function).

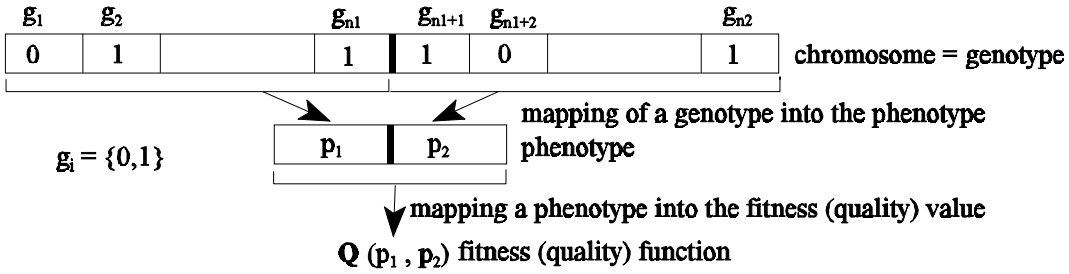

Figure 2. The representation of an exemplary individual in the classic GA

A representation used in the *K*-Model is similar to that in Fig. 1, but each individual consists of four chromosomes, and two phenes are considered.

2.3. The alphabet of genes coding

Watson and Crick have discovered that genes are digital strings of information, but not binary as it is assumed in the classic genetic algorithm. In biology, genes are strings of four symbols: A,G,C,T (nucleotides), and have different lengths. It is difficult to justify the binary coding on the base of biological analogy [1]. The *Schema Theorem* states that the number of trials of short, low-order, above-average schemata (*building blocks*) increases as the evolution goes on. For the simple genetic operators (mutation and crossover), and for the large population, it can be shown that the lower order alphabet, that is binary, is optimal [6]. But lately, this thesis has been discussed, and many applications use a different alphabet suitable for the current task [7, 3].

In some real problems, for example designing of artificial neural networks, it is difficult to use the binary alphabet for coding a whole structure of a network. On the other side, using larger alphabet for genotype defining with simple genetic operators, as in the classic genetic algorithm, causes loss of efficiency. In the *K-Model* the gene can be an integer value, genes are located at a number of chromosomes (four chromosomes in our simulation). We use the multiletter alphabet and we can apply some advanced operators. The model allows us to investigate the advanced genetic operators' influence on the nature of evolution. As a measure of the rate of evolution we assume the number of generations needed by evolving population to achieve a new fitness niche, that is, a global maximum in adaptive topography [13]. Such a study can give an intuition to search for the optimal parameters and operators of genetic algorithms to specific implementations.

3. *K*-Model model used in presented study

The model differs from the classic genetic algorithm in the representation of individuals and

applied genetic operators [11]. The genotype of an individual consists of four chromosomes: ch_1, ch_2, ch_3 and ch_4, and each of them can contain up to 21 genes. Genes are divided into two groups: the first constitutes *phenotype genes* (*active ones*) and the second – *redundant genes* (*latent*). There are ten (gp_1, ..., gp_{10}) phenotype genes, and they influence the phenotype of individuals. The number of redundant genes can change itself during the evolution. Fig. 3 illustrates a location of genes in the chromosomes. Each gen is an integer value (not a bit).

For each individual e_k, the number of its offsprings is calculated according to the Poisson distribution, with expected value λ_k equal to the ratio of its quality $\boldsymbol{Q}(e_k)$ to the average quality of the population

$$\lambda_k = \frac{Q(e_k)}{\frac{1}{N}\cdot\sum_{i=1}^{N} Q(e_i)}, \tag{2}$$

where N is a number of individuals in the evolving population.

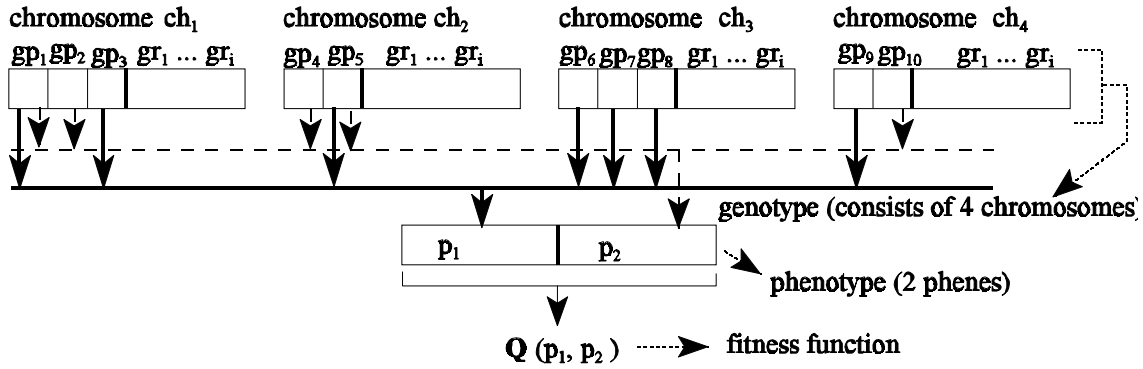

Figure 3. The representation of an individual in the *K*-Model

Values of the phenes p_1 and p_2 are calculated according to the equation 1, and only the phenotype genes are considered in this equation. Phenotypes genes are placed as follow: three first genes in the first chromosome, two first in the second, three first in the third, and the two first genes in the fourth chromosome. We see that the place of genes is important in the model.

Genetic operators

Recombination: reproduced individual can exchange its chromosome with randomly selected another individual. The probability of recombination p_r is assumed, and it is the same for each chromosome of an individual.

Mutation: it is implemented as a random change of one gene of an individual. Each gene (either phenotype or redundant) can mutate with a given probability p_m. Because genes are integer values, the mutation is implemented as a random change of a gene's value (similar to the *Creep* mutation operator in [3]). The size of this change is randomly selected within the assumed maximal range. Mutation of redundant genes is seen as the *neutral mutation*, because this change has no effect in the phenotype (and in its fitness value). However, this change can be expressed in the phenotype if that gene takes place one of the phenotype genes, during the *transposition* process.

Transposition: from time to time, one redundant (latent) gene exchanges its place with a randomly selected phenotype gene. It occurs with an assumed probability p_{trs}. In this situation, the redundant gene becomes the phenotype one (the active one), the active becomes a redundant one. This process of exchanging the place between one phenotype gene and one redundant gene is

called *transposition*.
Assuming a high probability of transposition, we can obtain big changes of values of phenotype genes. It models the macromutations.
Transition: a single gene (in fact, its copy) can be moved from one individual to another one. It is done with an assumed probability p_{trn}. Each gene from one individual can be moved and added to a randomly selected chromosome of a randomly selected individual, but always as a redundant gene.
The number of redundant genes changes during evolution. We can start with individuals without redundant genes, but due to *transition* process, redundant genes appear in the chromosomes. Assuming the probability of transition equal to zero, we can emulate the population of individuals without redundant genes, and therefore, without neutral mutations.
Recrudescence: the process called the *recrudescence* goes as follows. In each generation of an evolving population, a number of individuals (with an assumed probability p_{rcd}) have enlarged probabilities of mutation, recombination and transposition processes. It gives a radical reorganization of genotypes [3,10]. Such reorganization is named by Mayr "*loosing the cohesion of genotype*" [12]. One can expect that most of them are eliminated, but randomly, there arise *hopeful monsters* – offsprings that survive, and, they can enable the population to achieve a new fitness niche. In biology, internal stabilizing factors are responsible for that process. The recrudescence models the macromutations.
Crisis: it is also a radical reorganization of genotypes, but it concerns all individuals in an evolving population, and it cannot happen frequently. In biology, external factors are responsible for such processes. Crisis also models macromutations, but they occur rarely and in all individuals in the population. In the model a mean number of generations between crisis is assumed.

4. Experiments

The EVOLUTION program has been developed and used for simulation of the *K-Model*. The following ***fitness function Q*** with two peaks, and two parameters is used in all experiments:

$$Q = h_1 \cdot e^{(-((x-x_1)^2+(y-y_1)^2)\cdot n_1)} + h_2 \cdot e^{(-((x-x_2)^2+(y-y_2)^2)\cdot n_2)}, \qquad (3)$$

where:
x_1, y_1; x_2, y_2 – coordinates of the first and the second peak of the function, equal to (5,5) and (20,20), respectively,
h_1, h_2 – altitudes of the first and the second peak respectively, equal to 1 and 1.5,
n_1, n_2 – slope of the first and the second peak respectively, both equal to 0.02.

Izolines and the profile of this function are illustrated on Fig. 4. and Fig. 5. This form of function can be easy modified by adding new peaks (new component of the sum with h_i) or more parameters (new component in the exponents for x_i).

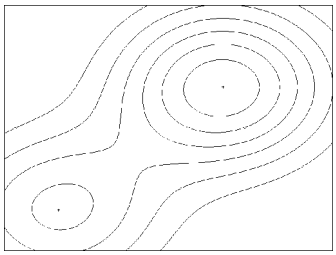

Figure 4. Izolines of fitness function, respectively: 0.35; 0.55; 075; 0.95; 1.15; 1.35

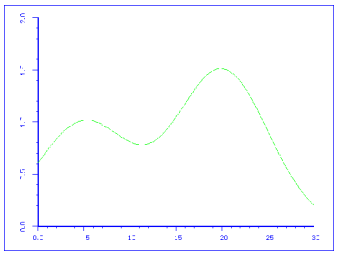

Figure 5. The profile of fitness function

The size of the evolving population has been set as 100 individuals. The evolution has been initiated from the local optimum – all individuals of the initial population were placed at the point (5,5) in the phenotype space. All chromosomes of initial individuals consist of only phenotype phenes, the redundancy is included by the transition process.

4.1. The results of simulation

A number of simulations show that the population reaches the global optimum during 300 - 500 generations, when the parameters are following: p_m = 0.002 (mutation), p_r = 0.02 (recombination), p_{trs} = 0.005 (transposition), without crisis. These values are considered as the base values for presented experiments. All simulation runs were repeated at least ten times.

The exemplary run with above parameters is shown in Fig. 6. There are four windows: the first presents the changes of the average quality of the population, the second – the coordinates, in the phenotype space, of the average values of the first and the second phenes, and the 3rd window – the trace of the moving population. A little window at the right-hand lower corner shows the actual number of generations.

Only mutation and recombination: the base parameters are taken, but the probability of transposition is set to zero (p_{trs} = 0). In this case, the probability of the transition is also set to zero (p_{trn} = 0), because the transition requires one phenotype gene and the redundant as the second one. The exemplary run is shown in Fig. 7.

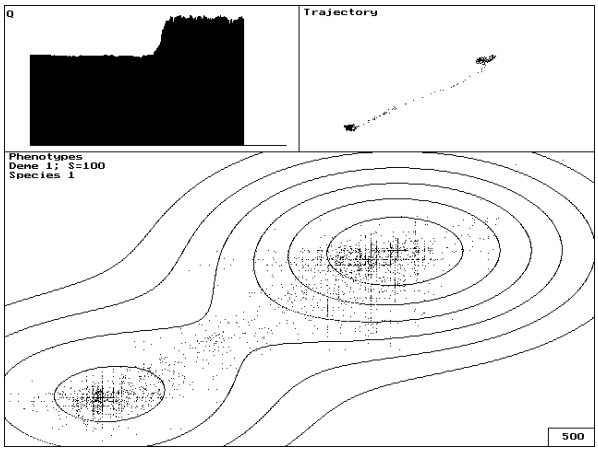

Figure 6. A typical run of evolution with the redundancy and without crisis

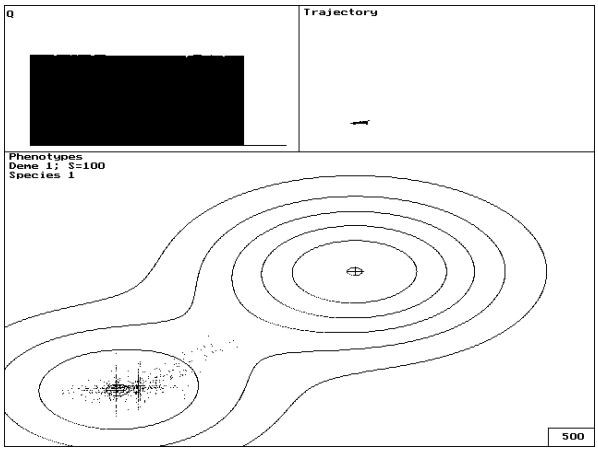

Figure 7. A typical run without redundancy and crisis

The population cannot reach the higher peak. During the assumed 1000 generations, there was no simulation run in which the best solution was found. Even no one individual originates near

the highest peak. So, we can say, that the simple genetic operators (mutation and recombination) do not work well in the model. The reason can be searched in the assumed maximal distance of mutation of a single gene; the proper mutation can change a single phene no more than 6 units, but the distance between the peaks is equal to 15 units.

Adding the crisis (without redundancy): we include the crisis, on average, after every 20 generations. It means that during each twentieth generation the genotypes of all individuals are reorganized. A similar situation as in the previous experiment arises. Without redundant genes, even frequently performed crisis has no effects. The population as a whole, or any individual, do not achieve the global optimum. Frequent crisis was also tested, but it does not give better results.

Including redundancy (crisis is present as in above experiment): when we impose positive values of probabilities of transition and transposition, the redundant genes are present and they play their role. We assume the following values of probabilities: $p_{trs} = 0.01$ and $p_{trn} = 0.005$. A typical run is shown in Fig. 8. The higher peak is reached by the whole population during 180 - 300 generations.

Comparing the results obtained in the above experiments, we can say that the role of redundant (latent) genes is essential. Always the single individuals are present near the highest optimum around 300 generations. We can say that redundant genes cumulate the changes and when they became the phenotype genes, they enable achieving distant area in phenotype space.

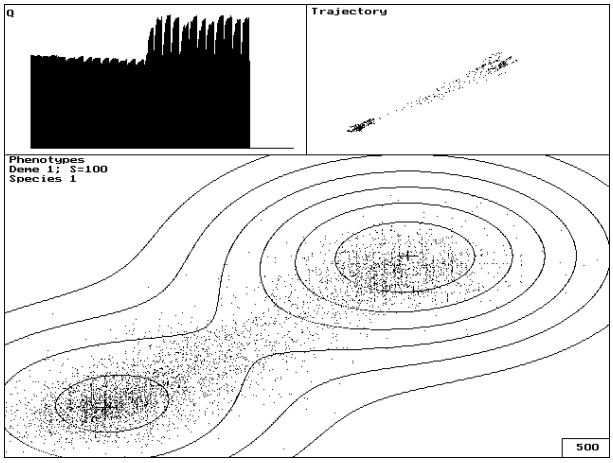

Figure 8. A typical run of evolution with the crisis

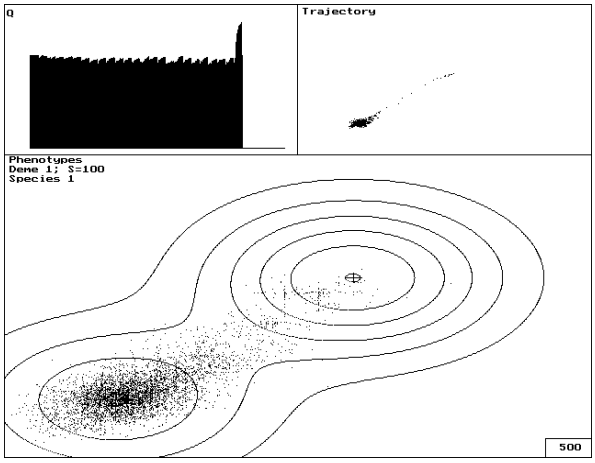

Figure 9. A typical run with the recrudescence

Redundancy and recrudescence (without crisis): we include the recrudescence with p_{rcd}=0.2. The evolution is a bit slower than in experiments with the crisis (see Fig. 9). Enlarging the probability of a recrudescence to 0.05 what causes that – on an average – after 20 generations all individuals should have reorganized its genotype, does not give a positive effect. The results seem to be a little worse than with crisis every 20 generations: the population usually reaches the global optimum but it fails from time to time.

These experiments illustrate the role of redundant genes, and crisis which is understood as the reorganization of genotypes of all individuals in the population. We had also run experiments without crisis and redundant genes to check the influence of recombination and mutation in the model. The first experiment is made without recombination and with enlarged mutation (changed from 0.002 to 1). The average quality of the population is significantly reduced. The evolution for $p_m = 0.3$ is presented in Fig.10. The population occupies almost whole phenotype space and therefore its average quality is not high. Fig. 11 shows the result of evolution with high frequency of crisis. From the biological population's point of view such a large variety of individuals is not optimal. There must be a balance between a variety and an achieved average quality. Next

experiment is with high recombination (changed from 0.02 to 1). The result is quite bad (Fig. 12 shows the run for $p_r = 0.8$), the global optimum is not reached.

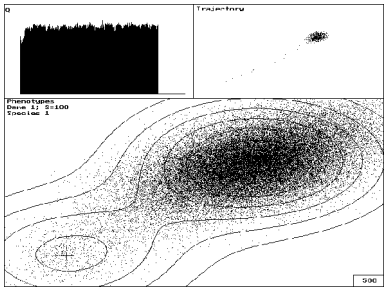

Figure 10. High mutation

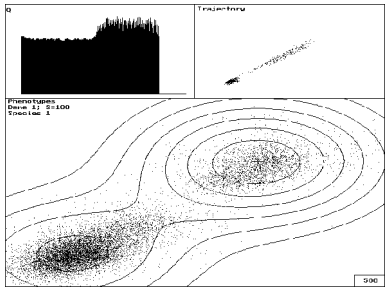

Figure 11. Frequent crisis

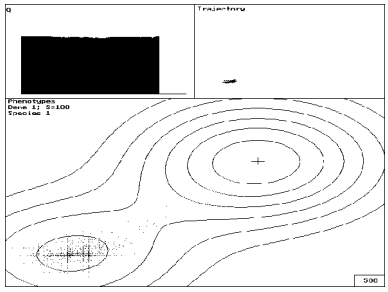

Figure 12. High recombination

5. Summary

Many real problems solved by using *Genetic Algorithms* are the optimization problems with many local optimums. The possibility of escaping from the local optimum is a very significant feature of genetic algorithms. But still applications of genetic algorithms to specific problems (e.g., a neural network design problem) do not give satisfactory results. Therefore, the investigation of dependence of evolutionary algorithms on applied genetic operators and parameters seems to be significant. It can suggest an efficient approach to build good optimizing tools, applicable to a wide spectrum of problems.

The present study concerns the evolution of the population which started from the local optimum. The main aim of the study is to investigate the possibility of escaping from the lower peak and reaching the global optimum. In the presented model the population almost always reaches the higher niche if the initial population is generated randomly. It needs no more then 30 generations.

Preliminary results of the simulations of *K*-Model do not allow us to univocally formulate a conclusion concerning the redundancy. It seems that the redundant genes play a significant role in the evolution of the population. The problem to investigate is, if a number of redundant genes influence the tempo and mode of evolution.

Similarly, the role of crisis seems to be not appreciated in related publications. Crisis seems to be an important operator. It cannot occur very frequently because the balance between searching for new solutions and climbing a hill can be violated.

References

[1] Back T. (1996), *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York, Oxford, 1996.

[2] Beasley D., Bull D.R., Martin R.R., *An Introduction to Genetic Algorithms*, Vivek, A Quarterly in Artificial Intelligence, January 1994, National Centre for Software Technology, Bombay, India.

[3] Beasley D., Bull D.R., Martin R.R., *Research Topics in Genetic Algorithms*, Vivek, A Quarterly in Artificial Intelligence, April 1994, National Centre for Software Technology, Bombay, India.

[4] Dawkins R. (1995), *Rzeka genów* (*River Out of Eden. A Darwinian View of Life*), Wydawnictwo CIS, Oficyna Wydawnicza MOST, Warszawa.

[5] Fogel D.B. (1992), *Evolving Artificial Intelligence*, University of California, SanDiego, A dissertation for the degree Doctor of Philosophy.

[6] Goldberg D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine*

Learning, Addison-Wesley Publishing Company, Inc.
[7] Goldberg D.E. *Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking*, University of Illinois at Urbana-Champaign, Urbana, (not published).
[8] Hoffman A. (1983) *Wokół ewolucji*, (*About evolution*), Biblioteka Myœli Współczesnej, PIW, Warszawa.
[9] Holland J.H. (1975), *Adaptation in Natural and Artificial Systems*, The University of Michigan, U.S.A.
[10] Kwaœnicka H., *Plejotropowoœæ i poligenicznoœæ w algorytmach ewolucyjnych* (*Pleiotropy and polygene effects in evolutionary algorithms*), V International Conference on Intelligent Systems, June 9-13, 1997, Zakopane, Poland (Proceedings in preparation)
[11] Kwaśnicka H., Markowska-Kaczmar U. *Do Semi-Isolated Subpopulations Evolve Quicker? Genetic and Evolutionary Algorithms*, MENDEL'96 2nd International Mendel Conference on Genetic Algorithms, June 26-28, 1996, Brno, Czech Republic.
[12] Mayr E. (1982), *Speciation and Macroevolution*, Evolution 36 (6), 1982.
[13] Waddington C.H. (1977), *Stabilization in Systems. Chreods and Epigenetic Landscapes*, Futures, April 1977.