COMPUTATIONAL PROBLEMS OF ELECTRICAL ENGINEERING

Vol. 6, No. 1, 2016

USING GENETIC ALGORITHMS FOR MODELLING INFORMATIONAL PROCESSES

Nataliya Boyko
Lviv Polytechnic National University, Lviv, Ukraine
nataliya.i.boyko@lpnu.ua

© Boyko N., 2016

Abstract. In this article genetic algorithms are considered including their types and practical applications. The scientific works of domestic and foreign researchers have been studied. This article presents methods and examples of solving tasks of data mining for genetic algorithms. The description of main components of models of genetic algorithms is presented. A parallel between biological systems and systems aimed at solving technical problems is drawn. The review and analysis of approaches to modeling of information processes with the use of genetic algorithms is carried out. The basic principle of modeling information processes on the basis of the evolutionary approach is analyzed. The models of the evolutionary process of information system are selected. The article highlights the practical use of the principles of genetic algorithms as tools for solving classical optimization tasks. Problems that have arisen with popularizing the tools of genetic algorithms are described. Several tasks of functional optimization described in mathematical language are analyzed.

Key words: information system, modeling, genetic algorithms, population, crossover.

1. Introduction

Genetic Algorithms are a specific type of analytical techniques which have been taken from nature having been used and verified by evolution. This type of algorithms is used for solving typical problems of function optimization or requests in databases. It is also possible to use them for data mining in complex distributed information systems. In the tasks of data mining these algorithms are used for combining the patterns of the rules of induction, of neural network exploration, of finding and comparing samples of data, of finding regularities in texts, etc. The scientists define the genetic algorithm (GA) as the main tool in data mining [2].

Genetic algorithms operate with a set of individuals (population) who constitute strings encoding certain solutions. The property of singling out a decision for its improvement while solving the optimization problem distinguishes the GA from most existing types of optimization algorithms.

For example, considering the function of adaptability, it is possible to sort a whole population into such groups:

– most adapted individuals (more appropriate solutions) which are able to interbreed and produce offspring;

– not adapted individuals (ineffective solutions) which are removed from the population and do not produce offspring.

If the solutions are used properly, the adaptability of new generation in the GA on average is higher than in the initial stage of the task. So, the possibility of the correct and proper use of adapted individuals during the study (effective decision) affects the quality of the solution of the optimization task as a whole. It happens, because in the classic GA [2]:

– initial population is formed randomly;

– population size (a number of individuals N) is fixed and does not change during the work of all algorithm;

– each individual generates a random L-bit string, where L is a length of the individual coding;

– length of coding is the same for all individuals.

Natural selection plays a key role in evolution. It consists in the fact that most adapted objects have better chances for surviving and have more descendants than less adapted ones. Due to the transfer of genetic information (genetic inheritance), descendants receive their main properties from parents. However, the natural selection alone can not ensure the development of species. The problem is that if differences between all descendants were very little, generations would differ only in number, but they will not differ in their adaptability. Therefore, it is important to investigate how the inheritance occurs, that is, how parents transfer their properties to descendants.

2. Main stages of the genetic algorithm

The basic law of inheritance states that descendants and their parents are to be alike. In particular, descendants whose parents are better adapted to existence are likely to be most adapted individuals in their generation. To understand the basics of this similarity, the author examined the structure of a cell of living organisms.

Almost every cell of any living being has a number of chromosomes which carry information about it. The main part of the chromosome is a molecule of deoxyribonucleic acid (so-called DNA strand) which consists of four types of connections called nucleotides, which are embedded in a certain sequence. They are denoted by letters A, T, C and G. Their order defines all genetic properties of a certain organism. DNA determines what reactions will occur in a given cell, the path of its development and a function this cell will perform. So, the genetic code of every living organism can be presented as a long strand of the combinations of four letters (A, T, C and G), and the gene is a segment of DNA chain responsible for some property, such as eye color, hair type, skin color etc. Different alternative forms of the same gene are called alleles. Features of the human beings are encoded by approximately 60 thousand genes which together include more than 90 million nucleotides [8].

There are two types of cells: somatic and gametic. Each human somatic cell consists of 46 chromosomes. These chromosomes form 23 pairs, where one chromosome in the pair is taken from a father and the other one from a mother. Chromosomes in each pair are responsible for the same features, for example, the father's chromosome may include a gene of green eyes and the mother's one a gray eyes gene. There are certain laws that govern the participation of genes in the development of their owner. In particular, there are dominant genes that suppress the recessive genes.

In gamete cells there are 23 chromosomes. During the process of fertilization male and female gametes merge and form an embryo cell which includes 46 chromosomes and is defined as a somatic cell. Getting certain features to a descendant depends on what gametes are involved in fertilization. The process of creating gametes (meiosis) is random, that is why descendants are very different from their parents [4].

In meiosis, paired chromosomes of somatic cells come close to each other, then their DNA breaks into pieces in several random places and chromosomes exchange identical areas. This process is responsible for the appearance of new variants of the chromosome and is called chromosomal crossover. Every newly formed chromosome then appears inside one of the gametes and its genetic information can be realized in the descendants of the individual.

Another important factor affecting the process of heredity is a mutation (a change in the genetic structure of an animal or plant that makes it different from others of the same kind) which is the permanent alteration of the nucleotide sequence of the genom of an organism. Mutations are random and can be caused by external factors, such as radiation. If the mutation takes place in gametes, the altered gene can be passed to the offspring and show up as a hereditary disease or some new feature of a child. The mutations cause the emergence of new types of individuals, and the crossover determines differences within species [2].

Population is a set of representatives of a certain type of organisms, which may mate, have their own territory and to some extent are separated from neighboring populations. Each population carries out the process of reproduction which is a combination of sequences and is called the creation of a new descendant.

From the point of genetic algorithms, the author gives a simple scheme for selecting effective solutions for optimal problem solving. First, the initial population of objects is formed being a certain number of solutions to the problem. This process is usually random. Further, reproduction of the representatives of this population is modeled. To do this randomly, several pairs of the population representatives are selected, crossing chromosomes between them is conducted, but the result is placed into the population of a new generation. In the genetic algorithm the principle of natural selection is stored: the more adapted the individual is (to carry out his tasks), the more likely the effective solutions in the crossing are.

The next stage of algorithm is the simulation of mutations in several random objects of new generation, certain genes being replaced. After this the previous population is partially or totally deleted and, according to the algorithm, the previous steps are repeated for the next generation. The population of the next generation often consists of the same amount of individuals as the initial one, but adaptability in it is on average higher due to conducted selection. Reduction is an operation that is used in the construction of GA and brings the current number of objects to a particular population size [1].

The use of genetic algorithms allows future generations to create unique solutions of problems. Solutions can be worse and better, but due to the selection the percent of better solutions will increase. In the following sections, the author will present basic stages of the genetic algorithm.

2.1. Function of selection in genetic algorithms

Intermediate population is a set of individuals who are able to reproduce. Individuals with the highest adaptability can enter the intermediate population several times and those with low adaptability are unlikely to enter there at all.

The author considers the probability of proportional selection of intermediate population in the classic GA. This selection can be implemented in several ways [2]:

- Stochastic sampling: in this realization individuals are placed onto roulette wheel so that the

area they occupy is proportional to their adaptability. After N spinnings of roulette, an intermediate population with a necessary number of individuals is chosen.

- Remainder stochastic sampling: for each individual the relationship between the adaptability of that person to the intermediate population is formed. As a result, received number indicates the required information, namely, its integer part corresponds to the number of times needed to record individuals in the population, and its fractional part indicates the probability of individuals' getting into the intermediate population. This method can be implemented quite easily, namely, by placing the applicants onto a roulette wheel just like it has been described above. Now, if the roulette is supplied not with one arrow, but with N arrows, one start of this roulette will select all N individuals expected in the intermediate population.

2.2. Mechanism of mating in the genetic algorithm

The mechanism of mating consists in the fact that individuals from intermediate population are randomly coupled and then reproduce with a certain probability. After that, a couple of descendants are obtained, which will be recorded in the next generation. But if individuals do not mate and reproduce, they themselves are recorded in a new generation [5, 10].

For this, in the classic GA, a one-point crossover operator can be used. It randomly determines the dividing point in the parental lines in which, after the exchange, parts of lines are cut off and descendants are formed using a set of these parts. An example of this mechanism is provided by the author below:

011010.01010001101 -> 111100.01010001101

111100.10011101001 011010.10011101001

2.3. Principle of mutations in genetic algorithms

The author considers mutation to be quite useful in the process of selection and reproduction in a new generation, as well as to protect the population against premature convergence and moving away from a local extremum.

Bits of every person are inverted with certain probability. But it should be noted that this probability is very low, often less than 1 %.

1011001100101101 -> 1011001101101101

The author proposed the selection of a certain number of points in the chromosome. For inversion it is enough to choose the required number of points and enter them randomly. There are cases, when a large sequence of points in a row is inversed. Choosing the number which is responsible for the probability of mutations, it is recommended to use the value of 1 / L or 1 / N.

2.4. Stopping criteria of genetic algorithm

According to the above algorithm, the author proposes conducting evolution an infinite number of times. So, it is important to choose the correct stopping criteria in the algorithm. They can be a number of iterations and to what extent the population converges.

The convergence is a state of population, when its lines converge within any extremum or can be almost identical. In such cases the crossover has little effect on the population and after mutation the individuals are less adapted. So, the population convergence means practical reaching the ideal solution of the optimization problems.

It can be achieved because the genetic algorithm searches for solutions by:

- hyperplane sampling which is performed by crossover, because the last one combines and replaces parents' samples in their children;
- method of hill-climbing which provides mutation: individual randomly changes, then bad variants are eliminated from population, useful changes are saved [9].

If the task is simple because of small population, the GA with mutation and without crossover will very quickly find the solutions. But if the function is complicated, the GA with crossover should be used. This mechanism is more reliable, although it increases requirements to the size of population.

If we consider the GA from the point of view of The Schema Theorem, we can see that the mutation, actually, inhibits the growth rate of the number of individuals which are the representatives of good schemata and destroys these individuals once again. As the mutation is a mandatory procedure for the GA with a small population, they have a problem of premature convergence, when in certain places all individuals in population have the same bit which does not match the global extremum.

For the GA analysis, selection pressure is used. It is the measure of the difference in the probabilities of better and worse individuals of population which have got to the intermediate population. In the case of proportional selection among the individuals of population, this value decreases together with the increase in average adaptiveness and thus approaches 1, so all individuals of population have the same chances to create a new generation.

When the probabilities of mating or mutation increase and selection pressure decreases, the rate of reproduction among those representatives who are already adapted decreases, but the speed of searching for

other schemata greatly increases. During the reverse GA, when the probability of mating or mutation decreases and selection pressure increases, the intensity of using found good schemata increases, but the search of new ones slows down.

The author has concluded that for finding the most optimal solution and saving the efficiency of working GA, it is necessary to find the right balance between investigating and using some effective solutions. For achieving this balance, the convergence of the algorithm should be balanced too. The reasons why it should be done are:

– if fitness is fast, the algorithm may converge on nonoptimal solution;

– if fitness is slow, we can lose the best individual, which we have found.

It is known, that the methodology of fitness management has not been developed yet. Therefore, the author suggested some variants of GA modification for making management solutions.

3. Modifications of genetic algorithms: encoding

The advantages of encoding with a binary alphabet:

– using hyperplanes provides a better search, thus it gives the maximum number of them. For example, encoding values with a binary alphabet, the number of hyperplanes will be greater than with the use of a four-digit alphabet;

– to get every symbol in each position you should work with a smaller population.

Even in the case of only two lines, there is the probability that in each point of the population there is either 0 or 1. In the case of the powerful alphabet a large part of the search space will not be available for the crossover before the implementation of mutation and after using the mutation the rest of the space will not be available too.

However, the non-binary alphabets are often more visual, so it is easier for a user to imagine the solution of the problem.

All genetic algorithms will work better using the Gray code rather than the simple direct binary code. This is because the Hamming distance between bit representations of the data may not display closeness, for example, the difference between numbers 7 and 8 is 4 bits. Also the search is complicated because the binary encoding adds more breaks [8].

The author gives an example of function minimization: if negative decisions dominated in the initial population, the solution would usually become -1 = 11 ... 1. But in this case the global minimum 00 ... 0 will be practically unattainable, because any change of a bit will only worsen the solution. If we use the Gray code, this problem will not occur.

The scientists [8–9] describe examples of encoding with floating point, which is better than direct binary encoding. This is because in some cases this encoding displays the concept of similar options of individuals better.

3.1. Selection of strategies as a method of modifying genetic algorithms

The author highlightes some types of strategy selection in the genetic algorithms [9, 10]:

1. Rank selection: for every individual the probability of getting into the intermediate population is his/her serial number in population, which is chosen by the increase in adaptivity. This type of selection does not depend on the average adaptivity of the population.

2. Roulette-wheel selection: it selects individuals with n “spins” of the roulette. The roulette wheel contains one sector for each member of the population.

3. Tournament selection: n individuals in the population are selected randomly and then the best one is selected among them and is put into the intermediate population. This process must be repeated N times until the intermediate population will be completely filled. Usually, this method is used when $n = 2$.

4. Truncation selection: from the population, sorted by fitness, the given part of the best individuals is taken, from which, randomly, one individual is chosen which will develop in the future.

3.2. Crossover in genetic algorithm

A crossover is called two-point crossover for two points to be selected on the parent organism strings. Everything between the two points is swapped between the parent organisms, rendering two child organisms:

010.1001.1011 -> 010.1011.1011

110.1011.0100 110.1001.0100

In this case, the length of two-point crossover is measured in circle, for example, for 1 ***** 1 it will be 1, and using one-point crossover for this sample it will be equal to 6.

Uniform crossover uses a fixed mixing ratio between two parents, where one of the descendants with probability p_0 inherits every bit from the first parent and with probability $(1–p_0)$ – from the second, then another descendant receives what is left of the first descendant. Usually, in this crossover $p_0 = 0.5$.

Uniform crossover, mostly, destroys the sample, so it is not used for the hyperplanes selection. However, working with a small population, it prevents premature transition from parents to descendants.

3.3. The strategies of forming a new generation

The author examines the process taking place after performing crossover and mutation. There are two basic types of formation a new generation [2, 11]:

– parents are replaced by descendants;

– new generation includes descendants and their parents.

In this case the principle of elitism is used: new generation includes only the specified number of best individuals from the previous generation. Basically it is one and the best individual.

Using elitism and the second basic strategy next time, it is impossible to lose the best solutions. For example, if the mutation led one of the lines of individuals to the global maximum, and the population is converged to a local maximum, then when parents are eliminated, most likely, this individual will be lost after crossing and the problem will not be solved. Using elitism, the best solution will remain in the population until solution better than previous will be found.

4. Some models of genetic algorithm

The author highlights specific models of genetic algorithms which contribute to providing the best solutions to problems in the process of modeling the information processes [11].

4.1. Genitor (Whitley) model

This model differs from the others in using a quite specific selection strategy. In this model, at each step, only one pair of randomly chosen parents creates one descendant, which replaces the worst individual in the population and does not replace the parents. So, in the population, at each step, only one individual is updated. The research conducted by the authors [3, 1, 8] proved that searching the hyperplanes, in this case, is better and convergence is achieved faster than in the classical genetic algorithm.

4.2. CHC (Eshelman) model

CHC means Cross-population selection, Heterogeneous recombination and Cataclysmic mutation. In this model, for new generation, N best representatives of the population are selected among parents and children. In this model, lines duplicating is not allowed. For crossing, all individuals are arranged in pairs, but crossing is allowed only for those pairs, where the Hamming distance is greater than the threshold. If it comes to crossing we will use so-called HUX-operator (Half Uniform Crossover scheme) which is a variant of uniform crossover, in which each of the descendants gets exactly half of the bits of each parent. This time, the population size is small and it is normal to use the uniform crossover. But, this algorithm, due to the luck of mutation, converges very fast. In this case, CHC uses cataclysmic mutation: all rows, except of the most adapted, are subjected to the powerful mutation, which changes about a third of the bits. After this, the algorithm reboots and continues working using only crossover.

4.3. Hybrid algorithm (Davis) model

That model allows uniting advantages of genetic and classic algorithms. Genetic algorithms allow finding right decisions. Usually, finding an optimal decision is a very difficult task because of stochastic behavior of algorithm. So authors [5] suggest using genetic algorithms at the start for effective restriction of searching space, and later apply one of "classic" methods of optimisation to the best individuals of the population. The author offers to use classic methods inside the genetic algorithms. In every generetion, after the creation of descendants, each of them is optimized by one of classic methods and, after that, each individual of the population reaches the local maximum. The next steps are selection, crossing and mutation. This method worsens the ability of the algorithm to find a solution by using hyperplanes, but it increases the chance of one of individuals to become a solution to the task.

4.4. Island model

The island model is a model of a parallel genetic algorithm. To use this model, the population should be divided into subpopulations. Each of them will evolve individually with the help of the chosen genetic algorithm. Author proposes "settling" individuals on a few isolated islands. Periodically, but rarely (for example, once per five generations) the migration starts, during which populations of the islands swap some good individuals [10] (Fig. 1).

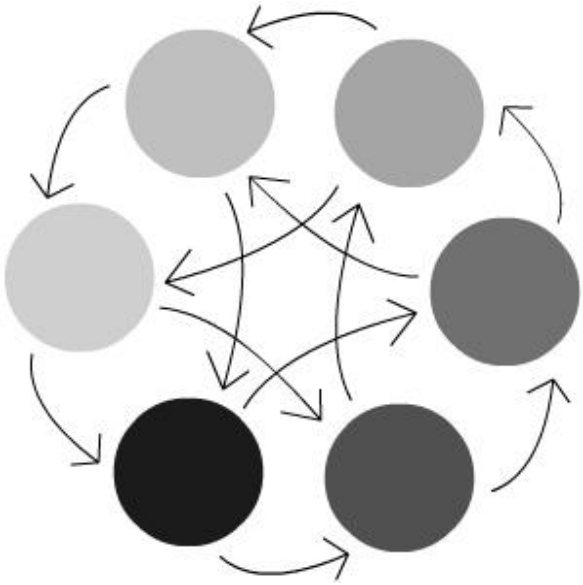

Fig. 1. Using island model.

Due to small quantity of individuals on the islands, subpopulations prone to early convergence, so the author introduces some requirements for the frequency of migration:

– in case of too frequent migration all subpopulations will interfuse and island model will lose its advantages turning into the typical genetic algorithm;

– in case of too rare migration early convergence of subpopulations will occur.

So, genetic algorithms are stochastic and different applications of these algorithms can result differently, because one population can lead to different solutions. The island model allows launching the algorithm for different solutions, which where found previously, what allows choosing really the best solution to a problem, set before this genetic algorithm.

5. Example of task solution using genetic algorithm

In this example author describes a task about N chess queens. On a chessboard of (N x N) size N chess queens should be placed in such a way, so they could not beat each other. For small N backtracking should be used. Below the author offers an example of genetic algorithm, which allows finding solution for N different values where N equals several thousands.

The difficulty of the task directly depends on the initial location of the queens. If only one figure is located in each row, it can be presented as O(N^N), and if only one queen stays in each row and column, it can be determined as O(N!). A number of variants can be decreased even more by analizing diagonal positions of figures.

5.1. Objective function and rejection of uneffective decisions

Solution to a task is a tuple of a size N, where index is a row, in which the queen is located, and value is a number of columns. For example, for N = 8 the solution is (4, 0, 3, 5, 7, 1, 6, 2), as shown in Fig. 2.

Fig. 2. Task solution for N=8

Individuals of initial population generate only those decisions, which do not make any threats horizontally and vertically. So, those decisions in which the value of objective is higher, than average number of all of generation individuals should be changed. At every iteration about a half of the population will be renewed.

5.2. Crossing functions

The author shows two functions, which can be used for creating descendants:

- *crossover* (for example: parent I (**4**, **1**, 6, 1, 2, **3**, **1**, 8);

  parent II: (**4**, **1**, 4, 6, 5, **3**, **1**, 9);

  descendant: (**4**, **1**, 7, 2, 4, **3**, **1**, 2).

The main advantage of this function is exchange of genetic materials. But because of permanent generation of new values in descendant it will work slowly.

- *gemmation*: descendant is generated, based on its parent, and two elements randomly replace each other in it. For example:

  parent: (**2**, 0, 3, 1, 6, 8, **7**, 5);

  descendant: (**7**, 0, 3, 1, 6, 8, **2**, 5).

This function will work faster, than crossover, but practically it represents cloning (genetic material does not change often).

5.3. Testing

For testing the author has chosen a population consisting of 75 individuals with the probability of mutation 3 %. The results of testing the genetic algorithm are shown in Table 1.

Table 1

Results of function testing

N	Crossover function		Gemmation function	
	Work time, sec	Generations, thou	Work time, sec	Generations, thou
500	663	43.6	69	10.3
1000	4269	94.7	233	24.8
2500	67180	297.3	1780	53.5
5000	Tests were not made		87569	227.6

5.4. Conclusions about function testing

1. The algorithm strongly depends on crossing function speed, so it should be chosen very carefully.

2. The mutation is less important, than crossing, because in real tasks it is almost impossible to get into local minimum.

3. Using genetic algorithm does not guarantee that a solution will be found, even if it is supposed to be known. If we need not only to find any solution, but every solution to a problem, then the genetic algorithm is not an optimal variant.

6. Conclusion

This theme is quite topical, since not only scientists but also managers of different companies are interested in automatic processing of a big amount of data. Nowadays optimization tasks are promising both for science and everyday life, because making a right decision means generating all possible alternative solutions, analyzing and, finally, choosing optimal solution. For practical and theoretical tasks the objective of optimization task is choosing a possible configuration of solutions from a set of alternatives. It means that GA technique can provide an optimum with objective function given in a process of treating some limitations.

Genetic algorithms are the most effective methods of wide variety of optimization methods. In most cases, by using the GA in any of variants finding the best solution or the closest to it becomes possible. For searching local optima the GA checks all solution space, even there, where they are the best. Simultaneously, some special mechanism works which does not allow algorithm to stop after choosing one solution and makes it look for new optimums.

In this article the author reviewed the main methods and approaches of data mining and possibilities of applying them using genetic algorithms. The basic concepts of genetic algorithms, their types, main models and modifications, which are used to find solutions to tasks, were reviewed in the article as well. They are so important, because the right choice of effective solutions is the main condition of modeling informational processes in the field of scientific and technical development.

References

[1] D. Whitley, An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls, *Journal of Information and Software Technology*, vol. 43, no. 14, pp. 817–831, 2001.

[2] D. Whitley, Genetic Algorithm Tutorial, *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[3] D. E. Goldberg and K. Sastry, A Practical Schema Theorem for Genetic Algorithm Design and Tuning, in *Proc. 2001 Genetic and Evolutionary Computation Conference*, pp. 328–335, 2001.

[4] J. Holland, *Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence*. London, UK: Bradford book edition, 1994.

[5] K. Deb and S. Agrawal, *Understanding Interactions Among Genetic Algorithm Parameters*, 1998.

[6] K. A. De Jong and W.M. Spears, A formal analysis of the role of multi-point crossover in genetic algorithms, *Annals of Mathematics and Artificial Intelligence*, no. 5(1), pp. 135–142, 1992.

[7] K. A. De Jong and W. M. Spears, An Analysis of the Interacting Roles of Population Size and Crossover, in *Proc. International Workshop "Parallel Problems Solving from Nature"* (PPSN'90), pp. 458–470, 1990.

[8] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: The MIT Press, 1996.

[9] R. Biesbroek, *Genetic Algorithm Tutorial. 4.1 Mathematical foundations*. 1999.

[10] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*, London, UK: A Bradford book, The MIT Press, 1992.

[11] S. Rana, *Examining the Role of Local Optima and Schema Processing in Genetic Search*. PhD thesis, Fort Collons, CO, USA: Colorado State University, 1998.

ВИКОРИСТАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ ДЛЯ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ

Наталія Бойко

Розглянуто поняття генетичних алгоритмів. Висвітлено практичні підходи та різновиди генетичних алгоритмів. Досліджено наукові роботи вітчизняних та зарубіжних дослідників. Наведено методи та приклади розв'язку задач дейтамайнінгу (data mining) для генетичних алгоритмів. Зазначено тезаурус основних складових моделей генетичних алгоритмів. Подано паралель між біологічними системами та системами, що спрямовані на вирішення технічних завдань. Проведено огляд та аналіз підходів до моделювання інформаційних процесів із використання генетичних алгоритмів. Проаналізовано базовий

принцип моделювання інформаційних процесів на підставі еволюційного підходу. Виокремлено моделі еволюційного процесу інформаційної системи. Звернуто увагу на питання практичного застосування інструментарію генетичних алгоритмів для класичних задач оптимізації. Окреслено проблеми, що виникли з популяризацією інструментарію генетичних алгоритмів. Проаналізовано декілька завдань виду функціональної оптимізації, описаних математичними засобами.

Nataliya Boyko – graduated from the Lviv Academy of Commerce, Ukraine (major "Economic Cybernetics"). She defended Ph. D. dissertation thesis in the sphere of mathematical modeling of economic processes. She works at Lviv Polytechnic National University, Ukraine. Research interests: methods of analysis and control of logistics activities, the problem of artificial intelligence and intelligent systems.