Lecture 8

Evolutionary optimization methods (EM)

Heuristic optimization methods

Heuristic methods are based on subconscious thinking and are characterized by an unconscious (intuitive) way of acting to achieve conscious goals.

The Heuristic Method only tries to find a good, but not necessarily optimal, solution.

- it does not guarantee that the best solution will be found
- it does not guarantee that a solution will be found, even if it certainly exists
- it may give the wrong solution in some cases

Heuristic optimization methods

These algorithms are widely used in solving problems with high computational complexity.

Heuristic algorithms can significantly reduce the time spent on a solution, but the resulting solution will not be the best compared to other solution methods. But still, in most cases, these algorithms provide an acceptable solution both in terms of quality and time.

Formulation of the problem

The objective function $f(x) = f(x_1, x_2, \dots, x_n)$ is given defined on the set of feasible solutions $D \in R^n$.

We need to find a global conditional minimum of the function $f(x)$ on a set $D \in R^n$:

$$f(x^*) = \min_{x \in D} f(x)$$

where $D = \{x | x_i \in [a_i, b_i], i = 1, \dots, n\}$

Classification of heuristic methods for searching for extrema

Evolutionary methods (direction in artificial intelligence (section of evolutionary modeling), which uses and models the processes of natural selection):

- genetic algorithms
- artificial immune system methods
- scattering method
- variable mesh optimization
- differential evolution method

Classification of heuristic methods for searching for extrema

Artificial Swarm Intelligence (method of amplifying the collective intelligence of networked human groups using control algorithms modeled after natural swarms):

- particle swarm optimization strategy
- ant colony method
- bacterial foraging optimization
- artificial bee colony
- fish school search
- glowworms swarm optimization

Classification of heuristic methods for searching for extrema

Methods that simulate physical processes:

- central force optimization
- simulated annealing method
- adaptive simulated annealing method
- harmony search method

Genetic algorithms

The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution.

At each step, the genetic algorithm selects individuals from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

Genetic algorithms

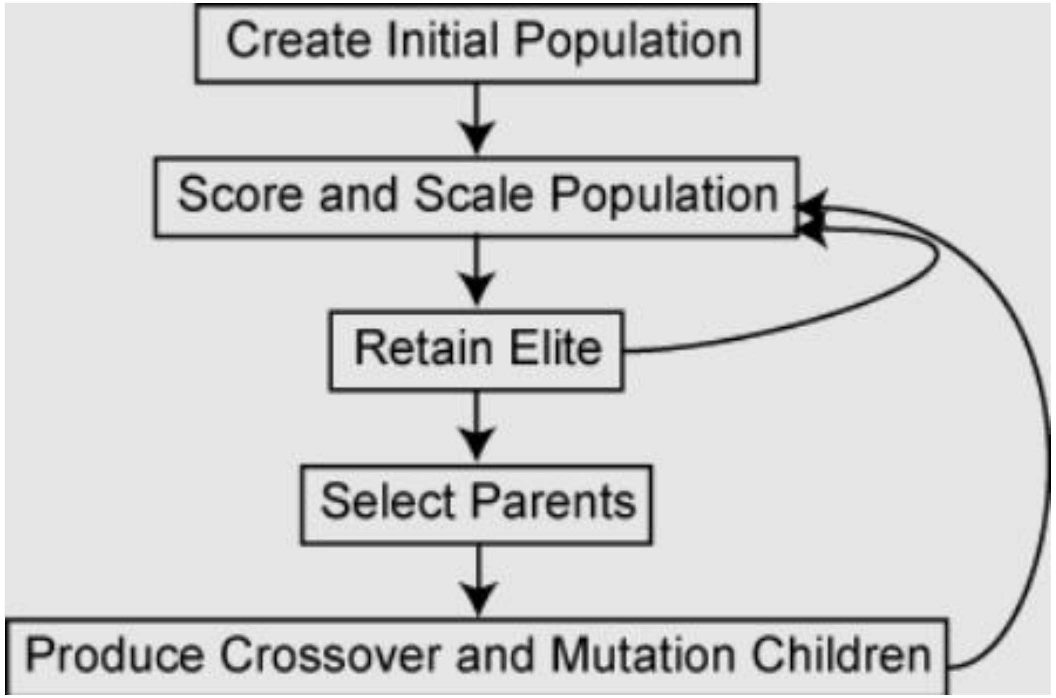
The genetic algorithm uses three main types of rules at each step to create the next generation from the current population:

•Selection rules select the individuals, called parents, that contribute to the population at the next generation. The selection is generally stochastic and can depend on the individuals' scores.

•Crossover rules combine two parents to form children for the next generation.

•Mutation rules apply random changes to individual parents to form children.

Genetic algorithms



Genetic algorithms

GAs imitate natural optimization methods in their work:

-genetic inheritance

-natural selection

The objective function $f(x)$ is equivalent to the natural concept of fitness of a living organism

Parameter vector $x = (x_1, x_2, \ldots, x_n)$ of objective function is called a **phenotype.**

Genetic algorithms

A **genotype** is a set of hereditary characteristics, information about which is contained in a chromosome set.

Phenotype is the totality of all the characteristics and properties of an organism that are formed in the process of interaction between its genotype and the external environment.

GAs search for solutions only at the genotype level.

Genetic algorithms

Each vector coordinate $x_i \; of \; x = (x_1, x_2, \dots, x_n) \in D$ is represented in some form s_i is called **gene**.

Let's transform $x = (x_1, x_2, \dots, x_n) \in D$ into some structure

$s = (s_1, s_2, \dots, s_n) \in S$ is called **chromosome** (genotype, individual):

$$D \to S \; (e)$$
$$S \to D \; (e^{-1})$$

Genetic algorithms

The general scheme of the algorithm looks like:

1. Creating an Initial Population
2. Selection (parent selection)
3. Crossbreeding (producing descendants)
4. Mutation (random changes in the genes of descendants)
5. Formation of a new population
6. Checking the termination condition

Artificial immune system method

The Artificial Immune Systems (AIS) method uses ideas borrowed from immunology, imitating the functioning of the immune system of a living organism.

The immune system of a living organism is the subsystem uniting organs and tissues that protect the body from diseases. The purpose of the immune system of a living organism is that it identifies and destroys foreign bodies that have entered the body, and improves, accumulating experience in dealing with them.

Artificial immune system method

An **antigen** is a substance that is perceived by a living organism as foreign and from which the body tries to protect itself. In order for the body to protect itself from antigens, it produces antibodies with the help of special immune cells.

An **antibody** is a substance that recognizes an antigen and contributes to its destruction.

A **memory cell** is an immune cell that stores information about new antibodies that can recognize an antigen, so that the next time the same or similar antigen enters the body, the immune system can work more efficiently.

Artificial immune system method

The objective function $f(x)$ is equivalent to the natural concept of the adaptability of an immune cell to fight antigens, i.e. the cell's ability to produce antibodies.

$f(x) -$ fitness function

Parameter vector $x = (x_1, x_2, \ldots, x_n)$ of objective function is called a **immune cell**.

Artificial immune system method

When solving the global minimization problem, finite sets $I = \{x^k = (x_1^k, x_2^k, \dots, x_n^k), k = 1,2, \dots, K\} \subset D$ of possible solutions which called populations are used, where x^k immune cell number k, K – population size.

Artificial immune system method

General scheme of the method of artificial immune systems:

1. Creating an Initial Population
2. Cloning
3. Mutation
4. Selection, formation of a new population
5. Population update
6. Checking the search termination condition

Advanced Artificial Immune Systems Method

A feature of the extended method is the combination of local and global search for a solution.

Local search is implemented as a cyclic iterative process during which biological operators are applied to the population: cloning, mutation and selection. Thus, the population changes to a new one.

Advanced Artificial Immune Systems Method

Global search is implemented as a cyclic iterative process. Each iteration of the global search is a local search. At the end of each iteration, population reduction occurs using clustering ideas. The remaining immune cells are marked as memory cells, after which the termination conditions of the method are checked. If the termination conditions are not met, then new individuals are added to the population, the number of which is proportional to the size of the population.

Advanced Artificial Immune Systems Method

General algorithm of the extended method:

1. Creating an Initial Population
2. Cloning
3. Mutation
4. Selection, formation of a new population
5. Checking the end condition of a local search
6. Population decline
7. Checking the global search end condition

Scattering method

In this method, the objective function $f(x)$ under consideration is called the fitness function, and the vector of parameters $x = (x_1, x_2, \dots, x_n)$ is called an **individual**.

The smaller the value of the objective function $f(x)$, the more adapted individual x is, i.e. suitable as a solution.

Scattering method

The search procedure begins with the generation of a basic set of individuals A.

To do this, first, on the segment $[a_i, b_i]$ changes in each coordinate, s subintervals of the same length are identified.

One individual is sequentially added to the base set:

1. For each segment $[a_i, b_i]$ number j_i is generated with a probability inversely proportional to the number of times this subinterval has already been selected
2. Coordinate values x_i are generated randomly

Scattering method

3. If the minimum distance between the formed individual and the individuals of the base set is greater than a given value σ, then the

continues.

When solving the problem, finite sets $I = \{x^k = (x_1^k, x_2^k, \ldots, x_n^k), k = 1,2, \ldots, Np\} \subset D$ are used, $Np -$ population size.

Scattering method

Scattering ideas also apply when forming an initial population: $Np = b_1 + b_2$ individuals are selected from the base set of individuals A.

Here b_1 is the number of individuals selected by quality, b_2 is the number of individuals selected by distance, i.e. the total distance from them to the individuals already present in the initial population should be minimal.

Scattering method

General algorithm of the scattering method:

1. Creating a Base Set
2. Creating an Initial Population
3. Generating pairs from population elements (selection)
4. Generating many descendants of a population (crossbreeding)
5. Formation of a new population from many descendants and the current population
6. Checking the end condition of a local search. If yes, Population decline. If not, step 3.
7. Checking the global search end condition

Evolutionary strategy for covariance matrix transformation

The generation of new individuals, the so-called mutation, occurs randomly according to the distribution formed by the evolutionary strategy. The best solutions found participate in the further evolutionary process.

Normal distribution is used to generate new individuals.

The objective function $f(x)$ under consideration is called the fitness function, and the vector of parameters $x = (x_1, x_2, \dots, x_n)$ is called an **individual**.

The smaller the value of the objective function $f(x)$, the more adapted individual x is, i.e. suitable as a solution.

Evolutionary strategy for covariance matrix transformation

When solving the problem, finite sets $I = \{x^k = (x_1^k, x_2^k, \dots, x_n^k), k = 1,2, \dots, Np\} \subset D$ are used, $Np -$ population size.

At each iteration, a new population is generated according to the distribution generated by the strategy. After this, the new population is analyzed and the distribution parameters are adjusted: step $\mu$and covariance matrix C, after which the transition to a new iteration is carried out.

The search procedure ends after the specified number of populations has been formed.

Evolutionary strategy for covariance matrix transformation

1. Determining the initial parameters of the strategy
2. Generation of a new population
3. Calculating a new reference point
4. Step calculation
5. Covariance Matrix Transformation
6. Checking the search end condition

Dynamic grid method

The evolution of the initial population occurs - the replacement of one generation by another through expansion and subsequent reduction of the population.

In the dynamic grid method, a population is represented as a grid consisting of a set of solutions called nodes.

The objective function $f(x)$ under consideration is called the fitness function, and the vector of parameters $x = (x_1, x_2, \dots, x_n)$ is assigned to each grid node.

Dynamic grid method

When solving the problem, finite sets $I = \left\{x^j = \left(x_1^j, x_2^j, \dots, x_n^j\right), j = 1,2, \dots, Np\right\} \subset D$ are used, where x^j is a node with number $j, Np -$ number of nodes in the grid.

The smaller the value of the objective function, the more suitable the node as a solution

Dynamic grid method

During the expansion process, new nodes are added to the grid:

1. Local extension
2. Global expansion
3. Additional expansion

Genetic algorithms

Genetic algorithms

A genetic algorithm is a heuristic search algorithm used to solve optimization and modeling problems by sequentially selecting, combining, and varying desired parameters using mechanisms reminiscent of biological evolution.

This is a method of enumerating solutions for those problems in which it is impossible to find a solution using mathematical formulas. The algorithm takes a group of solutions and searches for the most suitable ones among them. Then he changes them a little - he gets new solutions, among which he again selects the best and discards the worst.

Genetic algorithms

GAs consist of four main operators: selection, crossover, mutation, and creation of a new generation.

The cycle of crossing and mutation followed by fitness assessment is called **generation**.

Genetic algorithms

The algorithm for the process of forming a new generation can be represented as follows:

1. Create an initial population of K chromosomes (individuals). To do this, a finite set $X^k = \left(x_1^k, x_2^k, \dots, x_n^k\right), k = 1, \dots, K$ of trial solutions is randomly generated.

2. Assess the degree of fitness of each individual (calculate the values of the objective function $f_k(x) = f_k\left(x_1^k, x_2^k, \dots, x_n^k\right)$.

3. Exit if the stopping criterion is met, otherwise go to step 4.

Genetic algorithms

4. Select K parents from the population using the selection method (the probability of choosing a parent should depend on the degree of its fitness).

5. Select a pair of parents from the parent pool for reproduction. Using the crossover operator with probability p_c to obtain a child.

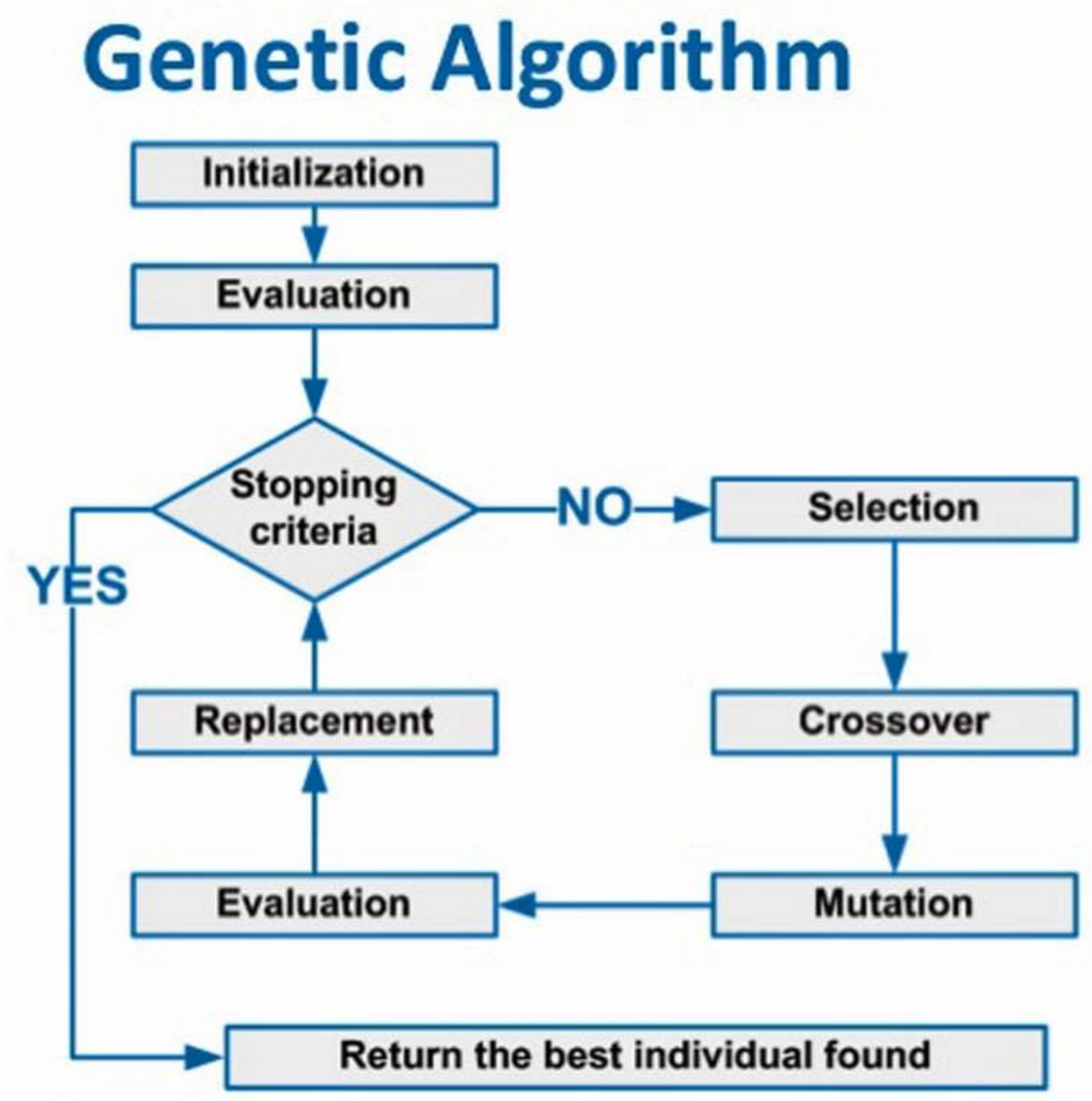
6. Subject descendants to mutation with probability p_m.

7. Repeat steps 5–6 until a new generation of population containing K chromosomes is generated.

Genetic algorithms

8. Assess the degree of fitness of each individual in a new population.

9. Go to step 4 if the number of generations does not exceed the allowed number.



Genetic Algorithm
Initialization
Evaluation
Stopping criteria
NO
Selection
YES
Replacement
Crossover
Evaluation
Mutation
Return the best individual found

Genetic algorithms

Population is a set of individuals (a set of solutions to a problem). At the beginning of the algorithm, an initial population (set of solutions) is randomly generated. These solutions will become better (evolve) as the algorithm operates until they satisfy the conditions of the problem.

Crossover is an operation in which two chromosomes exchange their parts.

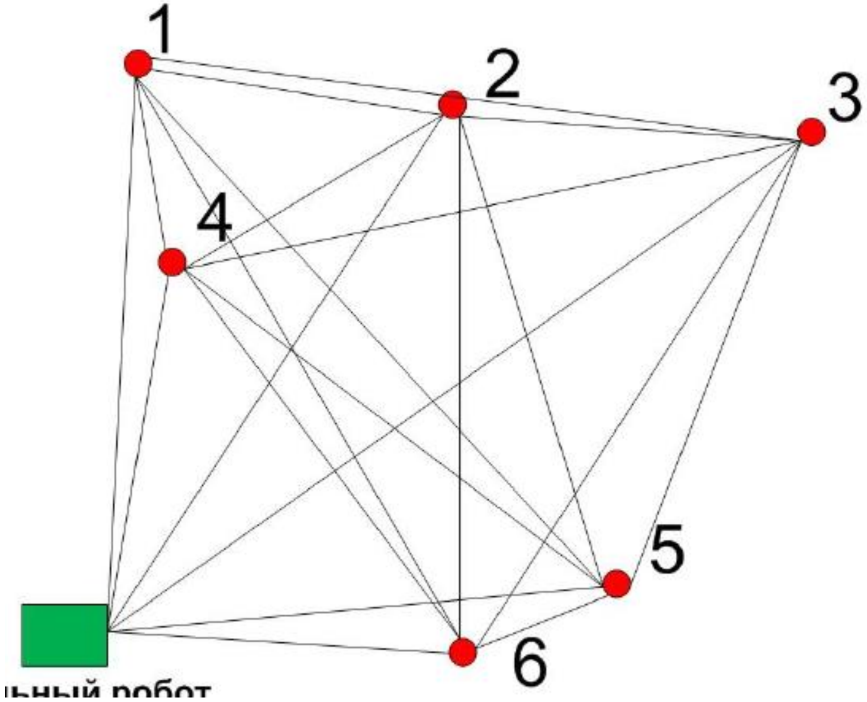
Mutation is a random change in one or more positions in a chromosome.

Inversion is a change in the order of bits in a chromosome or in its fragment.

Recombination is an operation in which two chromosomes exchange their parts.

Genetic algorithms

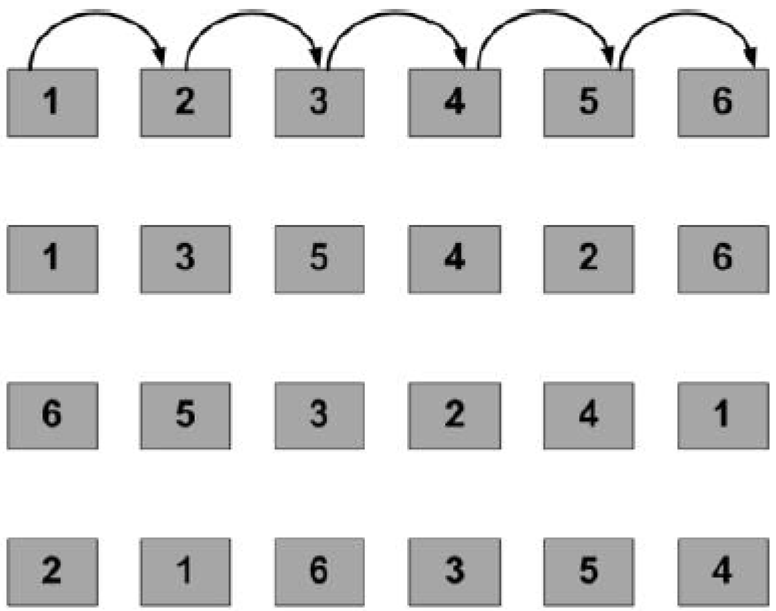
Example. Let's say the robot needs to go around six checkpoints in the least amount of time. The distance from each point to each is specified as a distance matrix.



	1	2	3	4	5	6
1		6	24	4	40	38
2	6		7	5	30	32
3	24	7		34	27	31
4	4	5	34		23	25
5	40	30	27	23		4
6	38	32	31	25	4	

Genetic algorithms

Let's take several possible solutions (individuals) - this is the initial population.



Genetic algorithms

Fitness function. In our example, this will be the sum of the distances from point to point in the selected route.

Let's calculate the fitness functions. For the first individual: $f_1(x) = 6 + 7 + 34 + 23 + 4 = 74$. For the remaining individuals we obtain in the same way: $f_2(x) = 111, f_3(x) = 47, f_4(x) = 125$

Thus, individual No. 3 is the best, and No. 4 is the worst.

Genetic algorithms

Coding determines what the chromosomes will look like. In the classical GA presented by Holland, binary coding of parameters is performed. In this case, chromosomes take on the form of binary sequences.

Mutation occurs by inverting a bit, and crossover occurs by exchanging parts of bit strings.

x_1				x_2				x_3				x_4				x_5				x_6			
1	1	0	0	0	1	0	1	1	0	1	0	1	0	0	1	0	0	0	0	1	1	1	1

Genetic algorithms

The selection operator is designed to improve the average fitness of a population in a new generation. The roulette-wheel selection method is often used as a selection operator.

This method received its name due to the fact that its principle of choosing parental individuals is reminiscent of playing roulette. The roulette wheel is divided into sectors; after starting and stopping the wheel, the arrow is more likely to point to the sector whose area is larger than the others.

Genetic algorithms

We can find the probability of choosing each sector. In a GA, the roulette wheel is divided into a number of sectors equal to the size of the population, and the areas of the sectors are compared with the fitness values of each individual.

Thanks to this, parent individuals are selected with a probability proportional to their fitness function: the “better” an individual is, the more likely it is to reproduce. The probability of choosing each individual is calculated by the formula:

$$p_s(x) = \frac{f_s(x)}{\sum_{i=1}^{K} f_i(x)}$$

Tournament method

It is a selection operator that selects individuals depending on the value of the fitness function. The main idea is to select the individual with the highest fitness from a certain number of individuals and allow it to reproduce (enter the parent pool). In this method, t individuals are randomly selected from a population containing K individuals, and the best individual is recorded in an intermediate array. This operation is repeated K times. The individuals in the resulting intermediate array are then used for crossing (also randomly). The size of the group of lines selected for the tournament is usually 2 ($t = 2$) or 3 ($t = 3$).

Crossover

Crossover is the exchange of genetics within a population. Moreover, it is understood that their genetic material will mainly be passed on by individuals with good fitness, because the purpose of crossing over is to improve the average quality of the population.

Crossover

In order to resist the algorithm to converge to a local extremum, the mutation operator is used.

Let i –th and j –th individuals of k –th population are presented as $p_i^k = (a_1, a_2, \dots, a_n) \in P^k, p_j^k = (b_1, b_2, \dots, b_n) \in P^k$.

In a single-point crossing, two elements of the population $k+1$ will arise: $p_i^{k+1} = (a_1, a_2, \dots, a_c, b_{c+1} \dots, b_n) \in P^{k+1}, p_j^{k+1} = (b_1, b_2, \dots, b_c, a_{c+1} \dots, a_n) \in P^{k+1}$ where point c is chosen randomly.

Mutation

Mutations are necessary to maintain the diversity of individuals in the population and prevent the solution from converging to a local optimum. Mutations play a role both in restoring lost genetic material and in the appearance of new characteristics in individuals.

Thanks to the mutation operator, parts of chromosomes are randomly changed.

The probability of a mutation occurring is usually taken to be $\frac{1}{L}$, where $L -$ chromosome length.

Operators for selecting individuals into a new population

The classic method of selection into a new population is to replace the old population with its descendants. There is also an elite selection strategy. There are two types of elitism: competitive and non-competitive. The first is the selection of the best individuals from the parent pool and the offspring pool. The second method involves the transfer of the best individuals from the old population to the new one, even if their fitness is worse than that of their descendants.

The main differences between genetic algorithms and traditional methods for finding solutions

1. Genetic algorithms work with code strings on which the values of the arguments of the target function and, accordingly, the value of the target function itself depend.

2. To find the best solution, the genetic algorithm uses several search points simultaneously in a separate step. This allows one to overcome one of their disadvantages – the danger of falling into a local extremum of the objective function.

3. The genetic algorithm uses both probabilistic rules for generating new points for analysis, and deterministic rules for moving from one point to another.