Computing and Informatics, Vol. 38, 2019, 1418–1443, doi: 10.31577/cai_2019_5_1418

GENERALIZED SELECTION METHOD

Jana Lovíšková

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: jana.loviskova@savba.sk

Daniel Pernecký

Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Ilkovičova 3
812 19 Bratislava, Slovakia
e-mail: xperneckyd@stuba.sk

Abstract. In this paper we introduce new selection method, 3-selection method. This method tries to generalize the most used selection methods in Genetic Algorithms (GA). Our new method involves both proportional and rank-based methods (order-based) and, moreover, it allows scaling of selection pressure with higher precision. This method is based on defining the shape of probability density distribution which is adjustable by parameters of our method. In addition, our method has one more attribute which adds randomness of selection.

Keywords: Selection method, evolutionary algorithm (EA), genetic algorithm (GA), continuous scalability of selection pressure, continuous scalability of randomness

Mathematics Subject Classification 2010: 97R40

1 INTRODUCTION

Evolutionary algorithms as metaheuristic optimization methods stand on few basic ideas. These ideas came from evolutionary principles found in living nature. By the influence of evolution, nature was able to make incredibly specialized and adapted species, by its own. Adaptation process to surrounding conditions could be seen as some kind of optimization. Adaptation process is conditioned by surroundings pressure, which determines the quality-fitness of individuals by life or death. Therefore surrounding is a quality criterion. Individual's quality is taken into account in two aspects. Fitter individual has a higher probability to transmit its good genetic information to offsprings and, moreover, has a higher probability to outlive than less fit individuals. This mechanism is called Darwin's natural selection.

Next key inspiration was already mentioned in information transmission from parents to offsprings. This mechanism consists of few joined principles. One of them is coding individual's properties into form of genetic sequence/string – chromosome, where each part represents one property/variable of problem being solved (optimization task). This string can be then modified by variation operators. Variation operators could be in form of recombination – crossover or mutation like in living nature. Chromosome modification allows to create individuals with new properties. Another main aspect is the existence of more solutions/individuals (population) at the same time. Unlike in classic optimization methods which work with only one solution. The existence of population is necessary for application of evolutionary principles, but, moreover, it allows to search the space of feasible solutions in parallel.

In evolutionary algorithms two main principles are used. Variation (recombination/crossover and mutation) which creates potential solutions and selection which applies the quality criterion. Selection mechanism is one of the key processes in evolutionary algorithms. Selection setup directly influences the speed of optimization process and quality of the solution found. If selection policy is setup to prefer only the best solutions, then it usually leads to the searching process stuck in local optimum also called as premature convergence. On the other hand, if selection strategy which has very low selection intensity is chosen then the time needed to find feasible solution could be too long, and the solution could not be found at all.

In basic EA the selection is applied in two phases. Choosing individuals – parents for reproduction process or other variation operator such as mutation, and choosing which individuals will survive to a next generation – called survivor selection.

1.1 Generational/Steady State EA

The application time and place/order of selection can significantly change the whole algorithm behavior. In the genetic algorithms there exist two modifications of such different usage. The first type and mostly used is a generational model of EA. Generational EA selects a big portion of individuals from the population. Selected individuals make pairs of parents and undergo variation operations. In the next step the survivor selection is used. New population is a joined group of modified individuals and only a few unchanged individuals from the old population.

The second type is a steady-state EA. In each generation only one pair of parents is selected. Parents undergo the variation operations and these modified individuals undergo the survivor selection. The survivor selection in this case decides which individuals in the population will be replaced by new children individuals.

These two types of EA differ only in the usage of the selection, therefore they have different selection schemes. The different selection scheme could be seen as unimportant diference, but it significantly changes algorithm behavior [1].

However, selection is not only used in the survivor and the parents selection. As the field of EA increased new types of algorithms were developed.

As mentioned previously, the selection operator in evolutionary algorithms is very important and has a huge influence on the convergence speed and on the quality of the final solution. Selection by itself is independent on the EA dialect. Whether it is a genetic algorithm, an evolutionary strategy, a genetic programming or other EA dialect, selection method is generally applicable for all types because of its only dependence on the individual's fitness or individual's genotype. The axiom of preferring better individuals on the expense of the worse ones is a base for most of the selection methods and fitness dependence is widely used.

The aim of our work was to develop a new fitness-based selection method, which is an extension and improvement of the already existing fitness-based selection methods. Our selection method involves both proportional and order-based methods and, moreover, it allows scaling of selection pressure with higher precision over the range of its possible values $\langle 0, 1\rangle$. Scaling of selection pressure with so high precision over the range of its possible values is not allowed by any of the fitness-based selection methods up to now. Our method has one more attribute which adds randomness of selection from range $\langle 0, 1\rangle$.

All of these three attributes (generalization) allow our proposed selection method, within a particular tackled task (solution landscape) and within the specific settings of algorithm's decision parameters, to achieve a better solution of the task than other well-known fitness-based selection methods.

We named this new general selection method based on the prescribed form of a probability density distribution as 3-selection method. We discuss it in detail in Section 3.

In addition, we have proposed a new classification of selection methods because we consider the schemes found in the current existing resources as improper and uncomplete. Our proposed classification of selection methods is stated in Section 2.

2 RELATED WORKS

First selection method was proposed by Holland in [2]. This method tried to take fitness of all individuals into account as objectively as possible. So, the logical step was to take individuals' fitnesses to a proportion, with individuals with better fitness having a higher possibility to be chosen than the worse ones. This method imitates a roulette wheel. Roulette wheel is divided into pieces with different sizes. Each size is proportionate to the fitness of each individual where better individuals have bigger portion of the wheel and vice versa. The wheel is then spin and the marked individual is chosen. The roulette as a hazard simulates aspect of surviving hazard-randomness in the living nature.

This method was later modified and extended. The modifications tried to solve the problem of premature convergence. The most used one is a stochastic universal selection (SUS) method, which has, in general, the same outcome as the roulette wheel selection method, but it solves a worse statistical properties of the basic roulette wheel method [3]. Despite the inherent simplicity of the roulette wheel method, it has been recognized that the roulette wheel algorithm does not, in fact, give particularly a good sample of the required distribution. Whenever more than one sample is to be drawn from the distribution, the use of the stochastic universal selection (SUS) algorithm is preferred [4].

Another classic method which works on completely different principles and one of the most used selection method is a tournament selection. It is a very powerful and simple selection algorithm.

Simply the method compares randomly chosen individuals and the better one is chosen. Here the rate of fitness difference or fitness proportion is not considered, but the main role is taken on individuals fitness order.

Comparison on tournament selection and roulette wheel selection and its modifications was made by more authors, for example in [3].

Natural development in EA research area has produced many other selection methods, which tried to solve premature convergence problem or tried to increase the quality of the final solution process by different approaches. Due to the existence of various selection methods based on different principles it is necessary to categorize the selection methods in order to achieve a clear and comprehensive overview.

At present, the categorization of the selection methods has not been clearly defined and there are not many resources where selection methods are strictly classified (to our knowledge there are only two resources). The authors in the first scheme [6] classified selection methods from the historical point of view. Authors in the second scheme [7] categorized selection methods in the base of their operation to "proportional", "ordinal" (or "order-based") and "steady state" categories.

But there exist many other selection methods which work on a different principle. More methods which select individuals based on their differences were developed. This type of selection tries to solve the premature convergence problem, by not using fitness as the selection criterion but as some kind of genotype metrics such as diversity.

Selection methods can be divided into fitness-dependent (classic selection methods), genotype-dependent (selection methods reflecting genotype) and special methods. Fitness-dependent methods can be further divided into proportional and order-based. Category of a special methods contains for example random selection method, which does not belong to any of the other categories, but also methods such as correlative tournament selection and correlative family-based selection.

Two main categories belonging to group of the fitness-based selection are the proportional selection methods and the order-based methods. Proportional selection selects individuals based on their fitness values relative to the fitness of the other individuals in the set population. Developed scaling policies are able to manipulate the fitness proportions distribution in population [5]. Order-based selection strategy also called ranking is based on order/rank of fitness values in population [14, 6, 7]. These methods were developed to overcome problems in proportional selection methods. For proportional methods there is a general disadvantage in cases where a very fit individual could come across among the population and that will significantly increase the selection intensity. In the order-based methods it is not important how big is the difference between individuals, whether it is few times more, or it is just a small difference, the ordering is the same.

With respect to the abovementioned facts we consider the first scheme [6] and the second scheme [7] as improper and uncomplete.

Therefore we proposed already mentioned own classification. In the proposed classification, the selection methods are divided into three main groups.

1. Fitness-based selection methods

   Fitness-dependent methods are divided into proportional and order-based. Proportional selection methods include: Roulette wheel selection (SSR, SSPR) [2, 9], Stochastic universal selection (SUS) [10], Stochastic remainder selection (with replacement, without replacement), and Deterministic sampling. Order-based selection methods include: Elitism, Tournament selection (Binary tournament selection, Larger tournament selection, Boltzmann tournament selection [11, 12]), Linear ranking selection [5, 8, 14, 15], Exponential ranking selection [8], and Truncation selection [8].

2. Genotype-based selection methods

   This group includes the Diversified selection method and methods based on gender-specific selection: Genetic algorithm with chromosome differentiation (GACD) [16], Restricted mating [17, 18, 19, 20], Genetic relatedness-based selection, Fitness uniform selection scheme (FUSS) [21], and Reserve selection [22].

3. Special selection methods

   This group consists of the following methods: Random selection, Correlative tournament selection [13], and Correlative family-based selection [13].

The proposed classification is a summary of selection methods found in literature. Unfortunately, only few of these methods have been seriously analyzed and compared [8, 3]. Some researchers tried to analyze selection methods by some measurable characteristics [3, 23, 8]. The most frequently used measurable characteristics are takeover time and selection pressure [3, 23]. These characteristics can be used to compare selection methods in a certain way and can provide an information why is some setting of evolutionary algorithm better than the other for particular cases. But they can say nothing about which selection method is more suitable than the other in general, because a different setting of selection pressure and takeover time may be appropriate in particular cases. Convergence of the EA depends also on the settings of the other algorithm's decision parameters and the solution landscape is a priori unknown (no free lunch theorem).

3 SELECTION METHOD BASED ON PRESCRIBED PROBABILITY DENSITY FUNCTION (3-SELECTION METHOD)

Selection based on the prescribed form of a probability density distribution arised from the idea to construct a general selection method for genetic algorithms. It represents the original, our suggested selection method for the fitness-based selection methods. If this method is sufficient in general, it must provide a wide scalability of selection rate, from a purely random selection to the elitist behavior.

The main role of selection is to maintain perspective pieces of information contained in the genotype, therefore a selection method must naturally prefer better individuals over the worse ones. Hence each selection method is based on the fact that a fitter individual has a higher probability of selection than a less fit individual. We solve minimization tasks, which means that the most fit (fittest) individual represents the minimum value of the fitness function.

Our method is based on defining the shape of probability density distribution. This shape of selection method must satisfy the criteria (if we assume minimization problem):

$$p(f_1) \leq p(f_2) \leq \ldots \leq p(f_i) \leq \ldots \leq p(f_n)\,;\, f_1 \geq f_2 \geq f_3 \geq \ldots \geq f_i \geq \ldots \geq f_n \tag{1}$$

where

- $p(f_i)$ is probability of selection of i^{th} individual whose fitness is f_i,
- f_1 is fitness value of the least fit individual,
- f_n is fitness value of the fittest individual.

If we preserve property defined in (1) and we consider that the prescribed probability curve is defined with some function $p(f_i) = F(f_i)$, then we can convert this feature to a specific selection method. The algorithm of a "general" selection is as follows (minimization problems):

1. Normalization of fitness values in the interval $\langle 0, 1\rangle$ by Equation (2) for proportional selection and by Equation (4) in case of order-based selection.
2. Selection of random individual whose normalized fitness is f_{n_i}.
3. If $F(f_{n_i}) \geq r$ then this individual is copied to a group of selected individuals; r is random generated number from the range $(0, 1\rangle$.

4. If the number of selected individuals equals to the number of individuals to be selected then end, else go to step 2.

Standardization of fitness values in the interval $\langle 0,1\rangle$ is performed in our selection from two reasons. First, for the generality of different scales of fitness landscapes. Second, for our prescribed form of the probability density distribution function. Standardization of fitness values is done in such a way that the fittest individual (in our case, with the minimum fitness value) will have the value of a normalized fitness 1, and the least fit individual (in our case, with the maximum fitness value) will have the value of normalized fitness 0. Other individuals will have fitness values between the boundary points of the interval $\langle 0,1\rangle$.

Fitnesses and normalized fitnesses have not the same ordering due to the construction of our prescribed form of the probability density distribution function, which is in form of (5), (6).

$$f_{n_i} = 1 - \frac{(f_i - \min(f))}{(\max(f) - \min(f))} \to f_{n_i} \in \langle 0,1\rangle\,. \tag{2}$$

Taking into account the fact that the selection methods can be generally classified into two main groups, namely a proportional and order-based group, a logical consequence is that if we replace in step 1 of our algorithm the standardization of fitness values for the standardization of order, considering sorted individuals:

$$\mathrm{srt}(f) = f_1 \geq f_2 \geq f_3 \geq f_4 \geq f_i \geq \ldots \geq f_n, \tag{3}$$

$$f_{n_i} = \frac{i}{N} \to f_{n_i} \in (0,1\rangle\,, \tag{4}$$

then we get the order-based selection method.

Our goal was to make the method of selection as general as possible, so it needs to be adjustable from a random selection to an elitist selection. Function F was therefore chosen in the form of (5), (6) where $\varphi \in \langle 0,1\rangle$ is the input parameter which defines selection pressure of the method. In (6), if $\varphi = 1$ we assume that $1^{infinity} = 1$ and numbers less than 1 including zero raised to an infinity tend to 0.

- If $\varphi \leq 0.5$

$$F(f_{n_i}) = f_{n_i}^{2\varphi}, \tag{5}$$

- if $\varphi > 0.5$

$$F(f_{n_i}) = f_{n_i}^{\frac{0.5}{1-\varphi}}. \tag{6}$$

With regard to another important feature of our selection method (adding randomness of selection), at first we clarify how the selection, respectively the feature of selective pressure, behaves in the evolution of the genetic algorithm.

If the selective pressure is high, it is causing fast convergence just by new individuals emerging in the population being created from the best individuals, and therefore the algorithm is searching in the direction of the fastest descent criterion function. This is due to the fact that selection allows manifestation of only this information which manifests itself immediately or in a few generations and from this reason some random change caused by a mutation here has not a big chance to survive and consequently to manifest. When the selective pressure is high, the algorithm otherwise converges quickly, but in the case of hard multimodal functions it results as stuck on local suboptimal solution.

On the other side, when the selective pressure is too low the algorithm has good ability to avoid local extremes but the time of convergence to the solution may be too long. In this case, the information resulting from global mutation have a big chance to survive longer because the selection allows individuals to carry this information to get to the next generation. However, a disadvantage is that many times the unperspective areas of the task are unnecessarily scanned, and it increases the time complexity of the whole algorithm.

A well-functioning selection method should, on the one hand, provide sufficient ability to increase the selective pressure, but, on the other hand, it should provide the possibility of randomness, it means a certain chance for unsuccessful individuals to survive as well. Consequently, there comes the possibility to combine worse and better individuals, and thereby to increase the probability of finding a global solution. We tried to incorporate this idea into the proposed selection method. In case of the probability density distribution shapes (Figure 1) it means shifting the whole curve upwards, and that is caused by the additional randomness (Figure 3). In other words, to a given probability density distribution curve we add a uniform probability density distribution with a certain amplitude σ, where $\sigma \in \langle 0,1 \rangle$ is a random parameter. Parameter of selection pressure $\varphi \in \langle 0,1 \rangle$. The final formula for function F is:

- if $\varphi \leq 0.5$

$$F(f_{n_i}) = (f_{n_i}^{2\varphi} + \sigma)/(1+\sigma), \tag{7}$$

- if $\varphi > 0.5$

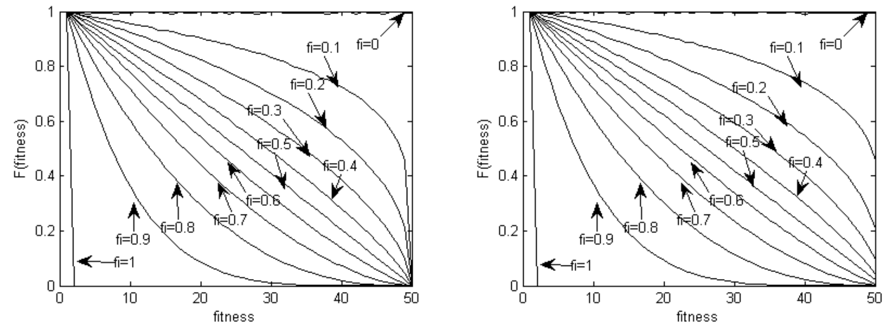
$$F\left(f_{n_i}\right) = \left(f_{n_i}^{\frac{0.5}{1-\varphi}} + \sigma\right) / \left(1+\sigma\right). \tag{8}$$

In (8), if $\varphi = 1$ we assume that $1^{infinity} = 1$ and numbers less than 1 including zero raised to an infinity tend to 0.

Probabilistic selection model of 3-selection method with adding randomness of selection:

- if $\varphi \leq 0.5$

$$p\left(X_i\right) = \frac{\left(f_{n_i}^{2\varphi} + \sigma\right) z\left(f_i\right)}{\sum\left(f_{n_i}^{2\varphi} + \sigma\right)}, \tag{9}$$



a) b)

Figure 1. Generated shape of selection probability according to Equations (5) and (6). On the left we can see the proportional selection for different parameters of φ, on the right there is the order-based selection. In this illustrative example, on the x-axis there are 50 fitness values where 1 is the best value and 50 is the worst value. On the y-axis we can see a corresponding selection probability of $F\left(f_{n_i}\right)$ for parameter φ.

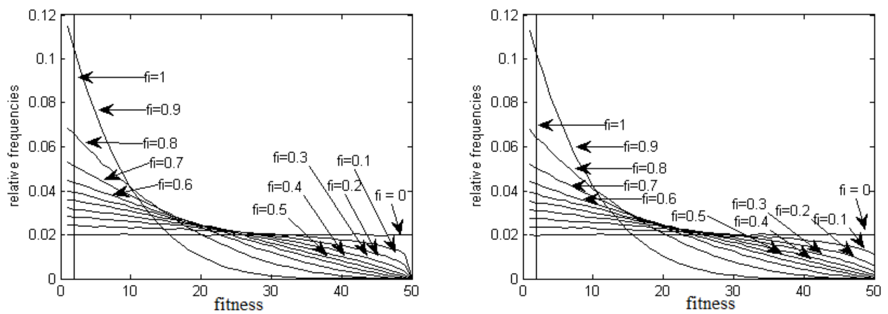


Figure 2. Corresponding relative frequencies for selection curves in Figure 1. On the left side there is the proportional selection method for different φ values, on the right side there is the order-based selection. On the x-axis there are 50 fitness values where 1 is the best value and 50 is the worst value. On the y-axis we can see a corresponding relative frequencies for parameter φ.

- if $\varphi > 0.5$

$$p\left(X_i\right) = \frac{\left(f_{n_i}^{\frac{0.5}{1-\varphi}} + \sigma\right) z\left(f_i\right)}{\sum \left(f_{n_i}^{\frac{0.5}{1-\varphi}} + \sigma\right)}, \tag{10}$$

$z\left(f_i\right)$ is the number of repeating of the i^{th} individual (fitness) in the population.

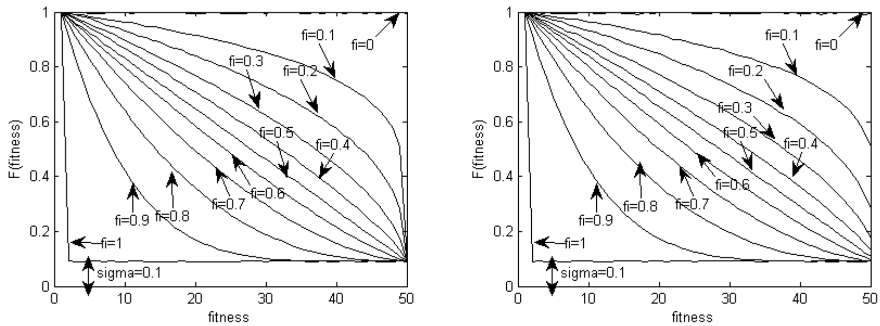


Figure 3. Generated shape of selection probability according to Equations (7) and (8). On the left there is the proportional selection for different parameters of φ, on the right there is the order-based selection. In this illustrative example, on the x-axis there are 50 fitness values where 1 is the best value and 50 is the worst value. On the y-axis we can see a corresponding selection probability of $F(f_{n_i})$ for parameter φ.

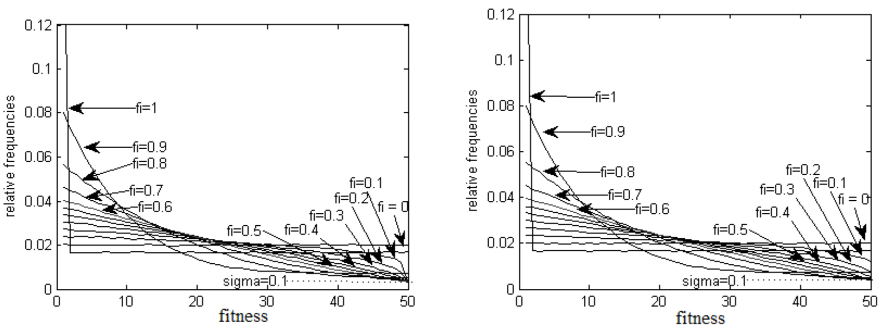


Figure 4. Corresponding relative frequencies for selection curves in Figure 3. On the left side there is the proportional selection method for different φ values, on the right side there is the order-based selection. On the x-axis there are 50 fitness values where 1 is the best value and 50 is the worst value. On the y-axis we can see a corresponding relative frequencies for parameter φ.

A significant advantage resulting from the stated properties of our selection is the ability to continuously change the value of selective pressure (φ) as well as the degree of randomness in the selection (σ).

To clarify the way 3-selection method works we present the algorithm of our proposed 3-selection method (Algorithm 1).

Algorithm 1: 3-selection method

Data: selection rate φ, rand rate σ, pp – proportional ($pp = 1$) or order-based selection ($pp = 2$), pop – input population, fit – fitness vector of individuals in input population, n – required number of selected individuals, norm_fit – normalized fitness

Result: popout – output population, fitout – fitness of individuals in output population

```
initialization popout, fitout;
if φ > 0.9999 then
    φ = 0.9999;
end if
φ = φ * 2;
if pp = 1 then
    if max(fit) − min(fit) ≠ 0 then
        norm_fit = (fit − min(fit))./(max(fit) − min(fit));
        norm_fit = 1 − norm_fit;
    else
        norm_fit = (fit)./(max(fit));
    end if
else
    if pp = 2 then
        sort pop according to descending fitness values;
        sort fit according to descending fitness values;
        nn – number of fitness values in fitness vector;
        norm_fit = (1 : nn) /nn;
    else
    end if
end if
count = 0;
```

4 EXPERIMENTS

The right setup of any evolution algorithm is not an easy task due to many variables needed to be set and our method provides 3 more additional parameters, and that could be seen as a disadvantage. On the other hand, it provides a possibility of very precise setting of a selection.

The influence of 3 new parameters is shown on 6 different GA (Table 1), which differ in the variation operators setup. We chose different GA setup in the meaning of a different exploration and exploitation rate. The results for different combinations of 3-selection method parameters on the 6 different GA setups are compared to the tournament and the SUS selection methods.

```
while n > count do
    j = round to the nearest integer towards infinity
      (random number ∗ population size);
    if φ ≤ 1 then
        if random number ∗ (1 + σ) ≤ σ + (norm_fit(j)^(φ)) then
            count = count + 1;
            save individual j from pop to popout to position given by the
              value of variable count;
            save fitness of individual j from fit to fitout to position given by
              the value of variable count;
        end if
    end if
    if φ > 1 then
        if random number ∗ (1 + σ) ≤ σ + (norm_fit(j)^(0.5/(1−(φ/2)))) then
            count = count + 1;
            save individual j from pop to popout to position given by the
              value of variable count;
            save fitness of individual j from fit to fitout to position given by
              the value of variable count;
        end if
    end if
end while
```

In the experiments for all tested functions we used a simple panmictic GA whose algorithm was:

1. Generate initial population of 50 individuals (chromosomes) – each individual (chromosome) consists of 5 genes and each gene generate randomly from the considered range of values (searching space) for particular test function.
2. Fitness evaluation of new or modified individuals – minimization problem – the most fit (fittest) individual has minimum value of the fitness function.
3. Selection of 3 groups of individuals:
    - Best – one best individual,
    - Old – 15 random selected individuals,
    - Work – 34 individuals selected by 3-selection method.
4. Crossover of the Work individuals.
5. Global mutation of the crossed individuals.
6. Local mutation of the mutated individuals.
7. Merging of groups Best, Old and Work to the new population.

8. If end condition is satisfied then end, else go to step 2.

The termination condition was set as the stagnating convergence of GA for 200 generations with 0 difference of best fitness. This allows to get the best result for a GA setup. The global mutation is a mutation which can modify a gene with a value from the whole searching space. The mutation probability is defined as (number of genes) $*$ (number of individuals) $*$ (mutation rate) $*$ (uniform random number).

For the local mutation an additive mutation was used, which adds a value from a space $\langle -\text{amps}, +\text{amps}\rangle$ to the mutated gene. The *amp* value is a maximal amplitude of the additive mutation, usually taken as a very small part of the searching space.

GA-1: low-div-LOCAL	*GA-2: mid-div-LOCAL*
50 individuals	50 individuals
5 genes	5 genes
One-point crossover	One-point crossover
Uniform global mutation 5 % Local (additive) mutation 5 % Amplitude of local (additive) mutation – 0.1 % from the space range	Uniform global mutation 20 % Local (additive) mutation 20 % Amplitude of local (additive) mutation – 0.1 % from the space range

GA-3: high-div-LOCAL	*GA-4: low-div-GLOBAL*
50 individuals	50 individuals
5 genes	5 genes
One-point crossover	One-point crossover
Uniform global mutation 50 % Local (additive) mutation 50 % Amplitude of local (additive) mutation – 0.1 % from the space range	Uniform global mutation 5 % Local (additive) mutation 5 % Amplitude of local (additive) mutation – 10 % from the space range

GA-5: mid-div-GLOBAL	*GA-6: high-div-GLOBAL*
50 individuals	50 individuals
5 genes	5 genes
One-point crossover	One-point crossover
Uniform global mutation 20 % Local (additive) mutation 20 % Amplitude of local (additive) mutation – 10 % from the space range	Uniform global mutation 50 % Local (additive) mutation 50 % Amplitude of local (additive) mutation – 10 % from the space range

Table 1. Genetic algorithm settings for different exploration and exploitation rate

In the experiments, we used 5 different test functions, namely Eggholder function, Quadratic function, Fnc1 function, Rastrigin function and Sgu2 function. Each of them has a different geometrical landscape and therefore a different degree of difficulty. In addition, each of the test function needs a totally different setup of decision parameters, namely the degree of a selection pressure, the degree of a randomness, the global and the local mutation rate and the amplitude of local mutation. For all of the test functions a common parameter was the number of searching space dimensions and it was set to 5.

The combinations of 3-selection method parameters were made as full factorial of selection φ and random σ parameters with 0.05 step and with both order and proportional type of selection. All of the provided results show the average of 100 runs for every combination.

In our article we provide only the best results (SR, MBF) compared to the results from other applied selection methods – SUS, tournament selection. For each used test function we only state the GA setup in which the best result is found (see Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11). The best results of success rate (SR) and mean best fitness (MBF) function are marked bold.

There are 5 292 different combinations of parameters values and GA variation operators setups for each test function. Concretely, 21 selection pressure setting options, 21 randomness setting options, 2 selection methods (proportional, order-based) and 6 different GA setups ($21 * 21 * 2 * 6 = 5\,292$). Our sel3 selection method gradually goes through all possible combinations of its parameter values and is looking for an optimal solution for each combination (therefore, the parameters of the selection method are not pre-set, all of them are passed sequentially).

The most interesting results for each of the 6 different GA setups (due to 5 292 of combinations per test function) compared to results for tournament and SUS selection for each test function we could present due to the large scale of tables in the Annex. As can be seen from these tables (listed in the Annex) for each test function, (for the indicator for SR and also for the indicator for MBF) the best results through the proposed 3-selection method were achieved.

The exception is evaluating the indicator of success rate (SR) for functions Rastrigin and Quadratic (less difficult test functions) – here the best result (100 %) was reached not only using selection method sel3 but also using the conventional standard selection methods SUS and tournament selection. For more difficult test functions (Eggholder, Fnc1, Sgu2) this is no longer true while the best results for the indicator of success rate (SR) and mean best fitness (MBF) are achieved only using the method sel3.

Of course, each of the test functions is, in general, achieving the best results by certain specific settings of variation operators of GA, which directly influence diversity and the level of degree of searching. Thus, in a certain specific (for given function the most suitable or a number of the most suitable) setting(s) of variation operators each of the used and tested selection methods shows better results for given test function in comparison with other settings of variation operators.

If we compare 3-selection method with the proportional type and with the order-based type using our five test functions, we can see that their effectiveness is relatively balanced. Whether it is, in a particular case, more effective to apply the proportional or order-based selection method always depends on the type of the test function and partly also on the setting of variation operators of genetic algorithm.

4.1 Eggholder Function

This test function has a strong multimodal character. The variables of this function are not linearly dependent, and it increases the difficulty of finding a solution. Global extreme of this function is unknown. By now the best reached minimal value of this function for 5 variables of the considered range $\langle -500; 500\rangle$ is $-3\,719.7$.

$$f(X) = \sum_{i=1}^{n-1}\left(-x_i \sin\left(\sqrt{|x_i - (x_{i+1}+47)|}\right)\right) - \sum_{i=1}^{n-1}(x_{i+1}+47)\sin\left(\sqrt{\left|x_{i+1}+47+\frac{x_i}{2}\right|}\right). \tag{11}$$

success rate [%] **SR -3650**	SUS	sel3 proportional	Tournament	**sel3** **order-based**
high-div-local	29 %	64 % p-sel = 0.65 p-rand = 0 and p-sel = 0.9 p-rand = 0.3	61 %	**67 %** **p-sel = 0.65** **p-rand** = 0.15

Table 2. Eggholder function – SR

mean best fitness **MBF**	SUS	sel3 proportional	Tournament	**sel3** **order-based**
high-div-local	-3 274.18	-3 515.68 p-sel = 0.65 p-rand = 0.1	-3 502.99	**-3 538.92** **p-sel = 0.65** **p-rand = 0.15**

Table 3. Eggholder function – MBF

4.2 Quadratic Function

Unimodal function. This function has only one extreme, but, for testing purposes, it is very useful. The value of extreme of this function is 0, where $x_i = 0$ for $i = 1, 2, 3, \ldots, 5$; $x_i \in \langle -500; 500\rangle$.

$$f(X) = \sum_{i=1}^{n} x_i^2. \tag{12}$$

success rate [%] SR 0,1	**SUS**	**sel3 proportional**	**Tournament**	**sel3 order-based**
low-div-local	**100 %**	**100 %** **all parameters** **p-sel, p-rand**	**100 %**	**100 %** **all parameters** **p-sel, p-rand**

Table 4. Quadratic function – SR

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
mid-div-local	0.0000126	0.0001259 p-sel = 1 p-rand = 0.15	0.0000653	**0.00000147** **p-sel = 0.95** **p-rand** = 0

Table 5. Quadratic function – MBF

4.3 Fnc1 Function

Fnc1 function is designed so that every position and value of extreme is known. It consists of one declining hyper-space of x^3 and randomly generated Gaussian functions. For this test function, it is characteristic that for finding a solution a high degree of randomness is needed. The randomness rate should be at least 0.45. It seems that for this test function and for the proportional selection (when randomness rate ≥ 0.5), to find the best results, the size of the selection parameter does not matter much. We would like to discuss this phenomenon in more detail in the following article.

$$\sum_{j=1}^{ex} \prod_{k=1}^{dim} \frac{-\sqrt{\frac{s_1(j)}{\Pi}}}{e^{s_2(j)(x(k)-o(k,j))^2}} + \left(\sum_{i=1}^{dim} 0.002 x_i \right)^3 . \tag{13}$$

The parameters s_1, s_2, o were once randomly generated, $-5 < x_k < 5$ and $k = 1, 2, \ldots, 5$. The value of global minimum (global extreme) is -56.4176 and global extreme position is $x_1 = 4.0587$; $x_2 = -2.9964$; $x_3 = 2.7314$; $x_4 = -4.7486$ and $x_5 = -1.0560$. The function has 50 extremes randomly distributed in the space and one corresponding to the minimum of the hyperspace x^3.

4.4 Rastrigin Function

Rastrigin function is multimodal function that shows strong periodical character with the regular occurrence of extremes. It belongs to the separable test functions. The value of global minimum (global extreme) of this test function is 0, where $x_i = 0$ for $i = 1, 2, 3, \ldots, 5$; $x_i \in \langle -500; 500 \rangle$.

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cos(2\Pi x_i) \right) . \tag{14}$$

success rate [%] **SR -50**	SUS	**sel3** **proportional**	Tournament	sel3 order-based
high-div-local	2 %	**41 %** **p-sel = 1** **p-rand = 0.5** **and** **p-sel = 0.05** **p-rand = 0.55** **and** **p-sel = 0.15** **p-rand = 1**	0 %	36 % p-sel = 1 p-rand = 0.45 and p-sel = 0.9 p-rand = 0.65 and p-sel = 1 p-rand = 0.8

Table 6. Fnc1 function – SR

mean best fitness **MBF**	SUS	**sel3** **proportional**	Tournament	sel3 order-based
high-div-local	−25.256	**−37.568** **p-sel = 0.05** **p-rand = 0.55**	−16.243	−36.636 p-sel = 0.15 p-rand = 0.45

Table 7. Fnc1 function – MBF

success rate [%] **SR 0,1**	SUS	sel3 proportional	Tournament	**sel3** **order-based**
mid-div-local	100 %	100 % p-sel= 1 p-rand = 0 and several others	99 %	**100 %** **p-sel = 1** **p-rand = 0** **and many others**
low-div-local	100 %	100 % p-sel = 1 p-rand = 0 and several others	100 %	**100 %** **p-sel = 1** **p-rand** = 0 **and many others**

Table 8. Rastrigin function – SR

4.5 Sgu2 Function

Test function Sgu2 has a strong multimodal character and belongs to non-separable functions. The location and the value of global extreme is unknown for $x_i \in \langle -500; 500 \rangle$, $i = 1, 2, 3, \ldots, 5$. So far the best value achieved by different experiments was −46.2655.

$$f(x) = \sum_{i=1}^{n-1} - \left| \ln \left(\left| \arctan(x_{i+1}) - \arccos(x_i) \right| - \Pi \left(\sin(x_{i+1}) - \cos(x_i) \right) \right) \right| . \quad (15)$$

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
mid-div-local	0.0005	0.0033 p-sel = 1 p-rand = 0	0.013	**0.0003** **p-sel = 1** **p-rand = 0**

Table 9. Rastrigin function – MBF

success rate [%] SR -25	SUS	**sel3 proportional**	Tournament	sel3 order-based
mid-div-local	61 %	**85 %** **p-sel = 0.7** **p-rand = 0**	73 %	78 % p-sel = 0.85 p-rand = 0.05

Table 10. Sgu2 function – SR

mean best fitness MBF	SUS	**sel3 proportional**	Tournament	sel3 order-based
mid-div-local	−27.209	**−28.189** **p-sel = 0.8** **p-rand = 0**	−27.181	−28.163 p-sel = 0.75 p-rand = 0

Table 11. Sgu2 function – MBF

success rate [%] SR -3650	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-global	28 %	49 % p-sel = 0.8 p-rand = 0	15 %	48 % p-sel = 0.85 p-rand = 0
high-div-local	29 %	64 % p-sel = 0.65 p-rand = 0 and p-sel = 0.9 p-rand = 0.3	61 %	**67 %** **p-sel** = 0.65 **p-rand** = 015
mid-div-global	30 %	53 % p-sel = 0.6 p-rand = 0.1	46 %	54 % p-sel = 0.6 p-rand = 0.1
mid-div-local	25 %	61 % p-sel = 0.25 p-rand = 0.2 and p-sel = 0.55 p-rand = 0.7	45 %	61 % p-sel = 0.25 p-rand = 0
low-div-global	13 %	40 % p-sel = 0.6 p-rand = 0.65	18 %	37 % p-sel = 0.45 p-rand = 0.7
low-div-local	21 %	51 % p-sel = 0.15 p-rand = 0.85	28 %	46 % p-sel = 1 p-rand = 0.7

Table 12. Eggholder function – SR

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-global	−3 314.42	−3 492.89 p-sel = 0.85 p-rand = 0	−3 421.74	−3 501.17 p-sel = 0.85 p-rand = 0
high-div-local	−3 274.18	−3 515.68 p-sel = 0.65 p-rand = 0.1	−3 502.99	**−3 538.92 p-sel = 0.65 p-rand = 0.15**
mid-div-global	−3 296.37	−3 513.28 p-sel = 0.6 p-rand = 0.1	−3 456.52	−3 509.35 p-sel = 0.6 p-rand = 0.35
mid-div-local	−3 251.13	−3 509.95 p-sel = 0.25 p-rand = 0.2	−3 392.8	−3 488.4 p-sel = 0.4 p-rand = 0.8
low-div-global	−3 129.94	−3 399.69 p-sel = 0.15 p-rand = 1	−3 185.14	−3 402.85 p-sel = 0.2 p-rand = 0.65
low-div-local	−3 137.88	−3 417.39 p-sel = 0.15 p-rand = 0.65	−3 229.15	−3 389.04 p-sel = 1 p-rand = 0.7

Table 13. Eggholder function – MBF

success rate [%] SR 0,1	SUS	**sel3 proportional**	Tournament	**sel3 order-based**
high-div-global	97 %	97 % p-sel = 1 p-rand = 0	0 %	99 % p-sel = 1 p-rand = 0
high-div-local	100 %	100 % p-sel = 1 p-rand = 0 and many others	100 %	100 % p-sel = 1 p-rand = 0 and many others
mid-div-global	100 %	100 % p-sel = 1 p-rand = 0 and p-sel = 1 p-rand = 0.05	32 %	100 % p-sel = 1 p-rand = 0 and several others
mid-div-local	100 %	100 % p-sel = 1 p-rand = 0 and many others	100 %	100 % p-sel = 1 p-rand = 0 and many others
low-div-global	96 %	97 % p-sel = 1 p-rand = 0.1	95 %	99 % p-sel = 0.55 p-rand = 0.05
low-div-local	100 %	**100 % all parameters p-sel, p-rand**	100 %	**100 % all parameters p-sel, p-rand**

Table 14. Quadratic function – SR

5 CONCLUSION

In this paper we introduced new selection method called 3-selection which enables higher scalability, covers both the proportional and order-based methods, and in addition, it has a randomness parameter. The examples show that for different settings of variation operators, which directly influence the diversity and the degree of searching, the meaning of selection method is changing.

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-global	0.0269	0.0272 p-sel = 1 p-rand = 0	19.867	0.0195 p-sel = 1 p-rand = 0
high-div-local	0.0000412	0.000283 p-sel = 1 p-rand = 0	0.0064	0.00000369 p-sel = 1 p-rand = 0
mid-div-global	0.0092	0.0126 p-sel = 1 p-rand = 0	0.2481	0.0097 p-sel = 1 p-rand = 0
mid-div-local	0.0000126	0.0001259 p-sel = 1 p-rand = 0.15	0.0000653	**0.00000147** **p-sel = 0.95** **p-rand = 0**
low-div-global	0.0309745	0.0335 p-sel = 1 p-rand = 0.1	0.0375027	0.0284 p-sel = 0.75 p-rand = 0
low-div-local	0.0000123	0.0000512 p-sel = 1 p-rand = 0.1	0.00000746	0.00000448 p-sel = 0.95 p-rand = 0.05

Table 15. Quadratic function – MBF

success rate [%] SR -50	SUS	**sel3 proportional**	Tournament	sel3 order-based
high-div-global	12 %	31 % p-sel = 0.6 p-rand = 0.3	0 %	23 % p-sel = 1 p-rand = 0.15
high-div-local	2 %	**41 %** **p-sel = 1** **p-rand = 0.5** **and** **p-sel = 0.05** **p-rand = 0.55** **and** **p-sel = 0.15** **p-rand = 1**	0 %	36 % p-sel = 1 p-rand = 0.45 and p-sel = 0.9 p-rand = 0.65 and p-sel = 1 p-rand = 0.8
mid-div-global	12 %	27 % p-sel = 0.15 p-rand = 0.6	0 %	32 % p-sel = 1 p-rand = 0.3
mid-div-local	5 %	35 % p-sel = 0 p-rand = 0.4	0 %	38 % p-sel = 0 p-rand = 0.35
low-div-global	4 %	21 % p-sel = 0 p-rand = 0.2	0 %	20 % p-sel = 0 p-rand = 0.15
low-div-local	2 %	20 % p-sel = 0 p-rand = 0.65	0 %	20 % p-sel = 0 p-rand = 0.65

Table 16. Fnc1 function – SR

Compared with the most used fitness-proportionate selection method the SUS and the most used order-based selection method – the tournament selection, it was also shown that the proposed 3-selection method is able to provide better results because this method enables more precise setting of GA, and consequently, more precise results are obtained. Experiments used binary tournament selection with

mean best fitness MBF	SUS	**sel3 proportional**	Tournament	sel3 order-based
high-div-global	−28.972	−34.569 p-sel = 0.9 p-rand = 0.75	−17.274	−33.686 p-sel = 0.95 p-rand = 0.3
high-div-local	−25.256	**−37.568** **p-sel = 0.05** **p-rand = 0.55**	−16.243	−36.636 p-sel = 0.15 p-rand = 0.45
mid-div-global	−17.323	−34.621 p-sel = 0.05 p-rand = 0.5	−18.479	−34.4 p-sel = 0 p-rand = 0.6
mid-div-local	−15.175	−36.171 p-sel = 0 p-rand = 0.4	−18.121	−36.191 p-sel = 0 p-rand = 0.35
low-div-global	−11.042	−31.647 p-sel = 0 p-rand = 0.2	−17.926	−29.317 p-sel = 0 p-rand = 0.25
low-div-local	−9.029	−28.439 p-sel = 0 p-rand = 0.2	−9.615	−27.72 p-sel = 0 p-rand = 0.45

Table 17. Fnc1 function – MBF

success rate [%] SR 0,1	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-global	0 %	0 %	0 %	0 %
high-div-local	100 %	100 % p-sel = 1 p-rand = 0	1 %	100 % p-sel = 1 p-rand = 0 and several others
mid-div-global	1 %	1 % p-sel = 1 p-rand = 0.05	0 %	3 % p-sel = 1 p-rand = 0
mid-div-local	100 %	100 % p-sel = 1 p-rand = 0 and several others	99 %	**100 %** **p-sel = 1** **p-rand = 0** **and many others**
low-div-global	0 %	1 % p-sel = 1 p-rand = 0.15 and several others	0 %	1 % p-sel = 0.45 p-rand = 0 and several others
low-div-local	100 %	100 % p-sel = 1 p-rand = 0 and several others	100 %	**100 %** **p-sel = 1** **p-rand = 0** **and many others**

Table 18. Rastrigin function – SR

arity = 2 and selection probability (for $k = 2$):

$$p(f_i) = \frac{(i_{\min}(f_i) + z(f_i))^k - (i_{\min}(f_i) + z(f_i) - 1)^k}{n^k} \tag{16}$$

mean best fitness MBF	SUS	sel3 proportional	Tournament	**sel3 order-based**
high-div-global	2.21	2.137 p-sel = 1 p-rand = 0	49.616	2.41 p-sel = 1 p-rand = 0
high-div-local	0.00107	0.018 p-sel = 1 p-rand = 0	3.864	0.0009 p-sel = 1 p-rand = 0
mid-div-global	1.488	1.403 p-sel = 1 p-rand = 0	6.971	1.385 p-sel = 1 p-rand = 0
mid-div-local	0.0005	0.0033 p-sel = 1 p-rand = 0	0.013	**0.0003 p-sel = 1 p-rand = 0**
low-div-global	2.582	2.6504 p-sel = 1 p-rand = 0.15	2.868	2.4198 p-sel = 0.9 p-rand = 0
low-div-local	0.0013	0.0027 p-sel = 1 p-rand = 0.2	0.0015	0.0008 p-sel = 1 p-rand = 0

Table 19. Rastrigin function – MBF

success rate [%] SR -25	SUS	**sel3 proportional**	Tournament	sel3 order-based
high-div-global	35 %	52 % p-sel = 0.85 p-rand = 0	3 %	49 % p-sel = 0.95 p-rand = 0
high-div-local	66 %	76 % p-sel = 0.8 p-rand = 0	16 %	79 % p-sel = 0.95 p-rand = 0
mid-div-global	47 %	57 % p-sel = 0.9 p-rand = 0	33 %	57 % p-sel = 0.8 p-rand = 0 and p-sel = 0.8 p-rand = 0.1
mid-div-local	61 %	**85 % p-sel = 0.7 p-rand = 0**	73 %	78 % p-sel = 0.85 p-rand = 0.05
low-div-global	28 %	38 % p-sel = 0.85 p-rand = 0.1	20 %	33 % p-sel = 0.55 p-rand = 0.3
low-div-local	52 %	66 % p-sel = 0.95 p-rand = 0.8 and p-sel = 0.5 p-rand = 0.7	43 %	64 % p-sel = 0.3 p-rand = 0.25

Table 20. Sgu2 function – SR

where

- $p(f_i)$ is selection probability of the i^{th} individual (fitness),
- $z(f_i)$ is number of repetitions of the i^{th} individual (fitness) in the population,
- $i_{\min}(f_i)$ is number of individuals from which the i^{th} individual has better fitness (for minimization tasks)
- k is arity of the tournament selection,

mean best fitness MBF	SUS	**sel3 proportional**	Tournament	sel3 order-based
high-div-global	−23.801	−25.212 p-sel = 0.85 p-rand = 0	−20.472	−24.949 p-sel = 0.9 p-rand = 0
high-div-local	−26.282	−27.837 p-sel = 0.9 p-rand = 0	−22.263	−27.952 p-sel = 0.95 p-rand = 0
mid-div-global	−25.141	−25.555 p-sel = 0.9 p-rand = 0	−24.234	−25.871 p-sel = 0.8 p-rand = 0
mid-div-local	−27.209	**−28.189 p-sel = 0.8 p-rand = 0**	−27.181	−28.163 p-sel = 0.75 p-rand = 0
low-div-global	−23.005	−24.0403 p-sel = 0.85 p-rand = 0.1	−22.891	−23.893 p-sel = 0.7 p-rand = 0.6
low-div-local	−25.49	−26.5197 p-sel = 0.5 p-rand = 0.7	−24.877	−26.3822 p-sel = 0.3 p-rand = 0.25

Table 21. Sgu2 function – MBF

- n is number of individuals in the population.

The selection probability was calculated for used SUS:

$$p(X_i) = \frac{(f_{n_i} \times z(f_i))}{\sum f_{n_i}}; \quad f_{n_i} \geq 0 \tag{17}$$

where

- f_{n_i} is normalized fitness value of the i^{th} individual (minimization tasks),
- $p(X_i)$ is selection probability of the i^{th} individual,
- $z(f_i)$ is number of repetitions of the i^{th} individual (fitness) in the population.

It could be argued that the proposed 3-selection method makes the process of designing GA more difficult. By providing more degrees of freedom the process of designing GA is more difficult. However, the extra properties of selection predetermine this method to be used in some sort of adaptive algorithms where such disadvantages become useful. But the ability of continuously changing selection pressure, influence of randomness, or optionally changing behavior of selection by switching between proportional and order-based selection are definitely great merits of the 3-selection method.

Acknowledgement

This work has been supported by the Slovak Scientific Grant Agency VEGA under the grant No. 2/0155/19.

REFERENCES

[1] Vavak, F.—Fogarty, T. C.: Comparison of Steady State and Generational Genetic Algorithms for Use in Nonstationary Environments. Proceedings of IEEE International Conference on Evolutionary Computation, Nagoya University, 1996, pp. 192–195, doi: 10.1109/ICEC.1996.542359.

[2] Holland, J. H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975. ISBN 978-0-472-08460-9.

[3] Goldberg, D. E.—Deb, K.: A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. Foundations of Genetic Algorithms, Vol. 1, 1991, pp. 69–93, doi: 10.1016/B978-0-08-050684-5.50008-2.

[4] Eiben, A. E.—Smith, J. E.: Introduction to Evolutionary Computing. Springer-Verlag, Berlin, Heidelberg, Natural Computing Series, 2003, doi: 10.1007/978-3-662-05094-1.

[5] Grefenstette, J. J.—Baker, J. E.: How Genetic Algorithms Work: A Critical Look at Implicit Parallelism. In: Schaffer, J. D. (Ed.): Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, San Mateo, CA, 1989, pp. 20–27.

[6] Sivaraj, R.—Ravichandran, T.: A Review of Selection Methods in Genetic Algorithm. International Journal of Engineering Science and Technology (IJEST), Vol. 3, 2011, No. 5, pp. 3792–3797.

[7] Shodhganga, `http://shodhganga.inflibnet.ac.in/bitstream/10603/32680/16/16/_chapter/%206.pdf`.

[8] Blickle, T.—Thiele, L.: A Comparison of Selection Schemes Used in Genetic Algorithms. Technical Report No. 11, Swiss Federal Institute of Technology (ETH), Computer Engineering and Communications Networks Lab (TIK), Zurich, 1995.

[9] De Jong, K. A.: An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral Dissertation, Dissertation Abstracts International, Vol. 36, 1975, No. 10, 5140B, University of Michigan.

[10] Baker, J.: Reducing Bias and Inefficiency in the Selection Algorithm. Proceedings of the Second International Conference on Genetic Algorithms and Their Application, Hillsdale, New Jersey, 1987, pp. 14–21.

[11] Khachaturyan, A.—Semenovskaya, S.—Vainshtein, B.: Statistical-Thermodynamic Approach to Determination of Structure Amplitude Phases. Soviet Physics, Crystallography, Vol. 24, 1979, No. 5, pp. 519–524.

[12] Goldberg, D. E.: A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing. Complex Systems, Vol. 4, 1990, No. 4, pp. 445–460.

[13] Matsui, K.: New Selection Method to Improve the Population Diversity in Genetic Algorithms. 1999 IEEE Proceedings of the International Conference on Systems, Man, and Cybernetics (SMC '99), Vol. 1, 1999, pp. 265–630, doi: 10.1109/ICSMC.1999.814164.

[14] Baker, J. E.: Adaptive Selection Methods for Genetic Algorithms. In: Grefenstette, J. J. (Ed.): Proceedings of the International Conference on Genetic Algorithms. Hillsdale, 1985, pp. 101–111.

[15] Whitley, D.: The GENITOR Algorithm and Selection Pressure: Why Rank Based Allocation of Reproductive Trials Is Best. In: Schaffer, J. D. (Ed.): Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, San Mateo, CA, 1989, pp. 116–121.

[16] Bandyopadhyay, S.—Murthy, C. A.—Pal, S. K.: Pattern Classification Using Genetic Algorithms. Pattern Recognition Letters, Vol. 16, 1995, No. 8, pp. 801–808, doi: 10.1016/0167-8655(95)00052-I.

[17] Fernandes, C.—Tavares, R.—Munteanu, C.—Rosa, A.: Using Assortative Mating in Genetic Algorithms for Vector Quantization Problems. Proceedings of the 2001 ACM Symposium on Applied Computing (SAC 2001), ACM, 2001, pp. 361–365, doi: 10.1145/372202.372367.

[18] Huang, C.-F.: An Analysis of Mate Selection in Genetic Algorithms. Technical Report CSCS-2001-002, Center for the Study of Complex Systems, University of Michigan, 2001.

[19] Ochoa, G.—Mädler-Kron, C.—Rodriguez, R.—Jaffe, K.: Assortative Mating in Genetic Algorithms for Dynamic Problems. In: Rothlauf, F., Branke, J., Cagnoni, S. et al. (Eds.): Applications of Evolutionary Computing (EvoWorkshops 2005). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 3449, 2005, pp. 617–622, doi: 10.1007/978-3-540-32003-6_65.

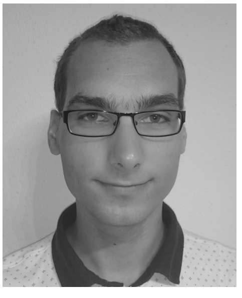
[20] Eshelman, L. J.—Schaffer, J. D.: Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. Proceedings of the 4^{th} International Conference on Genetic Algorithms (ICGA '91), Morgan Kaufmann, San Francisco, CA, 1991, pp. 115–122.

[21] Hutter, M.: Fitness Uniform Selection to Preserve Genetic Diversity. Proceedings of the 2002 IEEE Congress on Evolutionary Computation (CEC '02), 2002, pp. 783–788, doi: 10.1109/CEC.2002.1007025.

[22] Chen, Y.—Hu, J.—Hirasawa, K.—Yu, S.: Performance Tuning of Genetic Algorithms with Reserve Selection. Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC '07), 2007, pp. 2202–2209, doi: 10.1109/CEC.2007.4424745.

[23] Bäck, T.: Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms. Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence (ICEC '94), Vol. 1, 1994, pp. 57–62, doi: 10.1109/ICEC.1994.350042.

Jana Lovíšková works as a researcher at the Institute of Informatics of the Slovak Academy of Sciences in the Department of Modelling and Control of Discrete Processes. She received her Master's (2001) and Ph.D. (2010) degrees in the field of industrial management from the Faculty of Materials Science and Technology of the Slovak Technical University in Bratislava. Her research interests include optimization, search and scheduling problems solution by means of evolutionary computing, mainly genetic algorithms.



Daniel Pernecký received his Master's degree in cybernetics at Slovak University of Technology in Bratislava, Faculty of Electrical Engineering and Information Technology in 2014. His main resarch area is an optimization by metaheuristic methods algorithms.