Journal of Physics: Conference Series **1441** (2020) 012152 doi:10.1088/1742-6596/1441/1/012152

Fuzzy cultural algorithm for solving optimization problems

D T Muhamediyeva

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi, Amir Temur street, 108, 100200, Tashkent, Uzbekistan

Abstract. Cultural algorithm demonstrates incompetence in solving problems of multi-extremal optimization. This is due to the large dimensionality of the data and causes premature convergence. To solve this problem, a fuzzy cultural algorithm is proposed.

1. Introduction

The cultural algorithm (CA) is the improvement of population optimization algorithms by taking into account the experience gained in solving the problem. The founder of this algorithm is Reynolds [1, 2].

In the process of social development, people accumulate information about the world around them. This information can be considered a knowledge base of society, which its members can exploit in order to optimize their behavior. In a cultural algorithm, this knowledge base is formalized in the form of a beliefspace and is used as another element of the evolutionary impact on the population.

Any optimization problem without loss of generality can be defined as a global conditional minimization problem with inequality constraints.

$$\min_{X \in D \cap P} f(X) = f(X^*) = f^*,$$
$$D = \left\{ X \middle| \, f(X) \geq 0, P = X \middle| \, pl_j \leq x_j \leq pu_j, j \in [1:N] \right\}.$$

Let the set $D \cap P$ is convex, and everywhere in this set is the fitness function $f(X)$ takes non-negative values where the objective function should be optimized. The complexity of the optimization problem depends on the mathematical nature $f(X)$, hich makes the use of traditional optimization methods impractical for such a task [3, 4].

Given these considerations, algorithms based on nature, which are known as evolutionary algorithms, were developed to solve such problems. The task of the evolutionary algorithm is to simulate the processes of the natural process, such as reproduction and selection, to find good solutions using their earlier success in the previous stages of the quest. For this, many evolutionary algorithms have been developed over the years, such as: genetic algorithm (GA) [5], particle swarm optimization (PSO) [6], bee colony algorithm (ABC) [7], differential evolution (DE) [8] and the cultural algorithm [9].

For any evolutionary algorithm, there is a population space that is initialized by a population set at the beginning of the quest, and later individuals evolve using replay operations. Any evolutionary algorithm has various replay operations that can be called mutational and crossover operations and the purpose of creating a new generation of better populations. Analyzing the above-mentioned evolutionary algorithms, we can conclude that the cultural algorithm is the only evolutionary algorithm that uses various sources of knowledge, called the space of beliefs, to explicitly control the quest [10]. This space uses knowledge gathered over generations to help the population create new good solutions for optimal quest based on evolutionary algorithms. To coordinate the interaction between the population space and the space of beliefs, an algorithm is needed to connect these spaces. Evolutionary algorithms often demonstrate incompetence in solving a multi-extremal optimization problem, especially for large dimensions. Consequently, it causes stagnation and premature convergence [11].

Stagnation occurs when there is no improvement, and the solution to the best solution is repeated for successive generations, while premature convergence occurs when the population converges to the same solution, and the quality of this solution is not as expected [12].

To solve the above problems, which exist in almost every evolutionary algorithm, the researchers used three basic concepts to improve the efficiency of the cultural algorithm. The first of these is the use of hybridization methods in which a cultural evolutionary algorithm hybridizes with another method or local quest. Local quest algorithms are stochastic strategies based on population. Their main goal is to use existing knowledge in the field of quest. The researchers from [13] have integrated the cultural algorithm with repeated local quest to help cultural quest generate better solutions and ease stagnation and premature convergence, especially in subsequent generations. The second concept is to use another evolutionary algorithm as a population space of a cultural algorithm. For this purpose, many soft computing algorithms were used [11, 13].

Hybridization primarily relates to combining the best two methods [14]. The goal is to form a new agent who is expected to surpass their ancestors compared to a common optimization standard or application-specific problems. In [15], the PSO and CA algorithms are used to form a hybrid algorithm for solving limited optimization problems. They used a common population space, which consists of many swarms, and the belief space is used by collecting information about swarms. The renewed belief space was then used to influence the population and influence their evolution in order to generate better decisions. Another hybrid CA approach for solving problems with limited optimization is introduced in [16]. The authors used the differential evolution in the population space of a cultural algorithm with a well-designed influence function, which uses four sources of knowledge, normative, spatial, historical, and situational knowledge to help create new good quality solutions. Another well-balanced differential evolution algorithm was introduced in [17]. In this algorithm, a new influence function was introduced in order to use the recently updated knowledge to adjust the space of beliefs and to direct it to new promising regions. In [18] combined a local quest algorithm with CA to solve the traveling salesman problem.

In [13], an iterative local quest method is combined with CA to solve a variety of multi-extremal optimization problems. The authors used three steps to create good quality solutions. At the first stage a repeated local quest was used. Then the knowledge gained from the best people is used to update the space of beliefs. Finally, they performed a global quest to further improve the best generated solutions. All of these algorithms used the capabilities of SA, which use different knowledge sources to quest for regions of quest and collect additional information about the quest landscape. On the other hand, local quest methods are built in to increase the operational ability to customize the quest around newly located zones using CA. However, a suitable and well-designed structure should be developed for such hybrid algorithms that can balance quest modes, behave smarter with stagnation and cases of premature convergence, and can better use the knowledge generated at different stages of the quest.

Another multiplayer cultural algorithm was introduced in [19]. The algorithm was proposed to discover communities of agents on social networks by using different subgroups to extract knowledge in different ways, and then choosing the best knowledge to spread to the next generation. Three improved niche-based CAs were proposed in [20]. The population is divided into different niches in order to support different subgroups to determine the set of optimal options, where they were tested and compared to a number of problems with limited and engineering optimization. Consider a fuzzy cultural algorithm with a custom belief space. Based on the fuzzification operator, the relationship between the population space and the influence space and the selection of a high-quality population that influence the creation of knowledge sources through an updated process are estimated.

Cultural algorithms use cultural knowledge that is presented in the belief space component. This component consists of different sources of knowledge, any of which is responsible for a particular kind of cultural knowledge.

2. Problem Statement

Main components of the cultural algorithm are population space and belief space.

The space of a population and the space of beliefs interact through the function of acceptance and the influence function. With the help of the first of these functions, many best agents are identified who have the ability to correct the space of belief. The second function sets the rights by which beliefs are capable of influencing the evolution of population agents.

The pseudocode of the cultural algorithm is presented below.

Begin
g =0;
Initialization of the population S (g);
Initialization of the belief space B (g);
Population estimation with the help of fitness function S (g);
repeat
Interaction between the population space and the space of belief through the adoption functions (S (g), B (g));
Correction of the space of belief *B(g);*
The interaction between the population space and the space of persuasion through influence functions $(B(g), S(g))$;
$g = g + 1$;
Population evolution at the current iteration *S(g);*
Until *(termination condition)*
End

3. Correction of the space of belief

Knowledge relates to maintaining the performance of individual agents. This will lead to the movement of other agents towards the elite (generally best in this context) among the populations.

Knowledge is a set of interval information for any of N – measurements. Any interval of any parameter x_j of task N – dimension is represented as a set $< I_j, SL_j, SU_j >$, I_j is an interval of real numbers $[pl_j, pu_j]$, where the estimates are initialized using the specified values. SL_j and SU_j represent lower bound estimates pl_j and upper bound pu_j for parameter x_j, respectively. These performance metrics are initialized to a fairly large value. The exaggeration of this knowledge is progressive with the expansion of the interval and conservatism with the narrowing of the interval for any of the parameters. The parameter values for individuals taken by the fuzzy acceptance function will be used to calculate the allowable intervals for any of the parameters N. Spacing control is a way to control the speed of the evolution process during a quest.

Two membership functions of a fuzzy set, $R(x)$ and $L(x)$, will be used to update the upper and lower bounds, respectively, in the process of evolution. Performance evaluation will be the key to initiating a lower bound update and an upper bound of the intervals for any of the parameters N during the quest. Upper bound pu_j ill be shifted from the right side towards the core. In addition, the lower bound pu_j will be shifted to the left in the direction of the core. This update is performed to ensure that both boundaries fall within the core interval, which is defined as $[l_j, u_j]$ – center of parameter interval x_j.

Adjusted update process for belief space, individual X_b^g, $b = 1, 2, ..., NB$, where NB – the number of agents used to update the space of beliefs occurs according to the following algorithm:

Initialization of the lower bound and upper bound for the parameter x_j using the by lower and upper bound estimates for this parameter, as indicated in the task: $pl_j^g = pLD_j$, $pu_j^g = pUD_j$, where pLD_j и pUD_j – specified lower and upper boundary of the area for the parameter x_j.

Calculation of the fuzzy functions L (x) and R (x) using the membership functions that were adopted from [19] will be used to update the lower bound and the upper bound of the interval.

For Gaussian membership function $\mu(x) = \mathrm{L(x)} = \mathrm{R(x)} = \exp\left(-\left(\frac{x-a}{b}\right)^2\right)$ taking into account the values of partial derivatives

$$\frac{\partial \mu_j}{\partial a_j} = \frac{(x_j^* - a_j)\cdot \mu_j(x_j^*)}{(b_j)^2},\ \frac{\partial \mu_j}{\partial b_j} = \frac{(x_j^* - a_j)^2 \cdot \mu_j(x_j^*)}{(b_j)^3},$$

parameter values on $t+1$ are defined as follows:

$$a_j(t+1) = a_j(t) - \alpha\varepsilon_2\varepsilon_3\varepsilon_4 \cdot \frac{(x_j^* - a_j)\cdot \mu_j(x_j^*)}{(b_j)^2},$$

$$b_j(t+1) = b_j(t) - \beta\varepsilon_2\varepsilon_3\varepsilon_4 \cdot \frac{(x_j^* - a_j)^2\cdot \mu_j(x_j^*)}{(b_j)^3}.$$

For bell-shaped membership function $\mu(x) = \mathrm{L(x)} = \mathrm{R(x)} = \frac{1}{1+\left(\frac{x-a}{b}\right)^2}$ taking into account the values of partial derivatives

$$\frac{\partial \mu_j}{\partial a_j} = \frac{2b_j(x_j^* - a_j)^2}{\left((b_j)^2 - (x_j^* - a_j)^2\right)^2},\ \frac{\partial \mu_j}{\partial b_j} = \frac{2(b_j)^2(x_j^* - a_j)}{\left((b_j)^2 + (x_j^* - a_j)^2\right)^2},$$

parameter values on $t+1$ are defined as follows:

$$a_j(t+1) = a_j(t) - \alpha\varepsilon_2\varepsilon_3\varepsilon_4 \cdot \frac{2b_j(x_j^* - a_j)^2}{\left((b_j)^2 - (x_j^* - a_j)^2\right)^2},$$

$$b_j(t+1) = b_j - \beta\varepsilon_2\varepsilon_3\varepsilon_4 \cdot \frac{2(b_j)^2(x_j^* - a_j)}{\left((b_j)^2 + (x_j^* - a_j)^2\right)^2}.$$

For parabolic membership functions $\mu(x) = \mathrm{L(x)} = \mathrm{R(x)} = 1 - \left(\frac{x-a}{b}\right)^2$ taking into account the values of partial derivatives

$$\frac{\partial \mu_j}{\partial a_j} = \frac{2(x_j^* - a_j)^2}{(b_j)^2}, \ \frac{\partial \mu_j}{\partial b_j} = \frac{2(x_j^* - a_j)^2}{(b_j)^3},$$

parameter values on *t+1* are defined as follows:

$$a_j(t+1) = a_j(t) - \alpha\varepsilon_2\varepsilon_3\varepsilon_4 \cdot \frac{2(x_j^* - a_j)^2}{(b_j)^2},$$

$$b_j(t+1) = b_j - \beta\varepsilon_2\varepsilon_3\varepsilon_4 \cdot \frac{2(x_j^* - a_j)^2}{(b_j)^3}.$$

Here

$$\varepsilon_2 = \frac{\partial y}{\partial \mu^{d_j}(y)} = \frac{\overline{d_j}\sum_{j=1}^{m}\mu^{d_j}(y) - \sum_{j=1}^{m}\overline{d}_j\mu^{d_j}(y)}{\left(\sum_{j=1}^{m}\mu^{d_j}(y)\right)^2},$$

$$\varepsilon_3 = \frac{\partial\mu^{d_j}(y)}{\partial\left(\prod_{i=1}^{n}\mu^{jp}(x_i)\right)}, \ \varepsilon_4 = \frac{\partial\left(\prod_{i=1}^{n}\mu(x_i)\right)}{\partial\mu(x_i)} = \frac{1}{\mu(x_i)}\prod_{i=1}^{n}\mu(x_i).$$

3. Calculation $pc_j^g = a_j(t+1)$.

4. Updating the lower and upper bounds of the interval for the j-th agent, on generation g, is as follows:

$$pl_j^{g+1} = \begin{cases} pl_j^g + L(x_j^g)[x_j^g - pl_j^g], & pl_j^g \le x_j^g \le pc_j^g - b_j^g, \\ pl_j^g, & pc_j^g - b_j^g \le x_j^g \le pc_j^g + b_j^g, \end{cases}$$

$$pu_j^{g+1} = \begin{cases} pu_j^g + R(x_j^g)[x_j^g - pu_j^g], & pc_j^g + b_j^g \le x_j^g \le pu_j^g, \\ pu_j^g, & pc_j^g - b_j^g \le x_j^g \le pc_j^g + b_j^g. \end{cases}$$

4. Fuzzy acceptance function

This method is implemented in terms of a new fuzzy modification as follows. The formula is based on performance evaluation and is used to build fuzzy equivalence relations, and the generation number is used to determine the level of refinement**.** β . In the early stages of evolution, a smaller value is used to get more elites from a population of individuals. Then gradually reducing the value β , fewer agents will be used at the end of the quest.

Basic adoption functions are to find a fuzzy similarity matrix. This matrix represents a fuzzy similarity between any pair of individuals with respect to their attributes. Such attributes can be indicators such as age or distance from elites as a set of binary relations. To create this matrix, an evaluation of the effectiveness of the agent is used. This matrix is used as an equivalence relation in a fuzzy classification process, since a fuzzy similarity matrix is an equivalence relation.

Any element in this matrix is calculated by the following formula:

$$m_{r,c} = 1 - \frac{|f(X_r) - f(X_c)|}{\sum_k f(X_k)}, \quad 1 \le r, c, k \le NP.$$

Similarly, the original matrix definition (M_β) is used to use the matrix of fuzzy similarity in order to classify the population into two categories of performers: elite and inferior personalities. The elements of the matrix are defined as follows:

$$\beta_{m_{r,c}} = \begin{cases} 1, & m_{r,c} > \beta^{p_m} \\ 0, & m_{r,c} \le \beta^{p_m} \end{cases},$$

where $\beta^{p_m} \in [0,1]$ is the value of the refinement. In this notation, $\beta_{m_{r,c}} = 1$ means both individuals X_i and X_j are similar and, therefore, can be selected in the space of beliefs as an elite subset, and therefore, the elite agent will be used to update the space of beliefs.

The main stages of fuzzy adoption in the cultural algorithm are shown below:

1. Creating an equivalence relation M (matrix of fuzzy similarity) for the population space.
2. Using the generation number to calculate the level (β)refinement.

Using the generation number to calculate the level () refinement.

3. Use the value of refinement in M to perform a fuzzy classification.
4. Classification of a new generation of individuals based on their performance indicators. If the completion condition is ranyed (with a certain accuracy), then the end of the work. Otherwise, go to the next step.
5. Using the linear reduction function to update the value.

$$\beta = \begin{cases} random(\overline{v}(\beta), v_\sigma(\beta)), & p^m \ge \varepsilon, \\ 1 - \varphi(p^m - \varepsilon) / g, & 0 \le p^m \le \varepsilon, \end{cases}$$

where $p^m, \varepsilon, \varphi$ specified data.

6. Calculate the number of elite agents (the number of agents that will be accepted) using the matrix M.

During the initialization phase creating a refinement level (β) uses only for individuals with the current best performance from the elite set. For this purpose, linear reduction will be used. For this reduction function $\overline{v}(\beta)$ there is an average of the normal distribution at $p^m \ge \varepsilon$, and in this context, the set value is the standard deviation from such a distribution. The elite kit contains at least one agent in the first pass and will be used as a seed at the beginning of the optimization process. $v_\sigma(\beta)$ – amount of elements. These elements are those that will affect the population space. Any element in M is sorted in ascending order. The matrix M is symmetric.

5. Fuzzy influence function

The source of uncertain knowledge is the control of the direction and size of the mutation step. The motivation for this function depends on the idea that a well-working agent is more likely to stay longer in the population than an individual with a lower level.

This means that the probability of applying a mutation operator to an older agent is compared to a younger age. Accordingly, the upper bound of the fuzzy function is described as:

$$\beta = \begin{cases} 1 - g_{\max}(x_j^s)^k, & x_j^s < 1/g_{\max}, \\ 0, & x_j^s \geq 1/g_{\max}. \end{cases}$$

β will be used in the influence function as follows:

$$\tilde{x}_{i,j} = \begin{cases} x_{i,j} + \beta\left|(pu_j^g - pl_j^g)random(0,1)\right|, & x_{i,j} < pl_j^g, \\ x_{i,j} - \beta\left|(pu_j^g - pl_j^g)random(0,1)\right|, & x_{i,j} > pl_j^g, \\ x_{i,j} - \beta(pu_j^g - pl_j^g)random(0,1), & otherwise. \end{cases}$$

Knowledge in the space of belief will be used to regulate the direction and size of the step. Using this approach, if the parameter value in the parent is in the allowable range, then a small perturbation in the random direction will be applied. Otherwise, the current range of beliefs will be used to perturb the right or left boundaries of the current range found in the space of beliefs. If value $x_{i,j}$ less than pl_j^g, then the mutation operator will use the fuzzy adjustment coefficient to adjust in the negative direction. On the other hand, if the value $x_{i,j}$ is more than pl_j^g, then the mutation operator will use the fuzzy adjustment factor to update only in the positive direction. If the disturbance of the offspring violates the restrictions, then the adjustment method will be used.

The violation will be fixed by stochasticity values in the range of restrictions as follows:

$$\tilde{x}_{i,j} = \begin{cases} pLD_j + random(0,1), & \tilde{x}_{i,j} < pLD_j, \\ pLD_j - random(0,1), & \tilde{x}_{i,j} > pLD_j, \\ \tilde{x}_{i,j}, & otherwise. \end{cases}$$

To predict the occurrence of premature convergence, criteria are defined for such cases:

$$v = (\mu_{\max}^g - \mu_{\min}^g) / (f_{\max}^g(X) - f_{\min}^g(\mathrm{X})),$$

where $f_{\max}^g(X)$, and $f_{\min}^g(\mathrm{X})$ and also are the maximum and minimum values of the suitability of the g-th generation. $\mu_{\max}^g$ is the average fitness value, which is greater than the average fitness of the gth generation, $f_{\min}^g(\mathrm{X})$ this is the average availability of values that are less than the average fitness of the gth generation and the average value of fitness is defined as $\overline{f}_{\max}(X) = \sum_i f(X_i)/NP, \quad i = 1,2,...,\mathrm{NP}$.

v is an indicator of the distribution of agents. When value v s small, then the variety is acceptable. Otherwise, when the value v is large, then diversity should be taken into account by increasing the mutation rate. The success rates of all component space-population strategies will differ, allowing another quest to avoid current local optima. v defined in the range from 0 to 1 and $v \in$ { low medium high }.

6. Computational experiment

The algorithm was tested on test problems for multi-extremal optimization. This test consists of 3 complex optimization tasks:

- Ackley function (Fig. 1),
- Griewank function (Fig. 2),
- Levy function (Fig. 3).

Backley function:

$$f(x) = -a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp.$$

Recommended Variables: a=20, b=0.2 и c=2π.

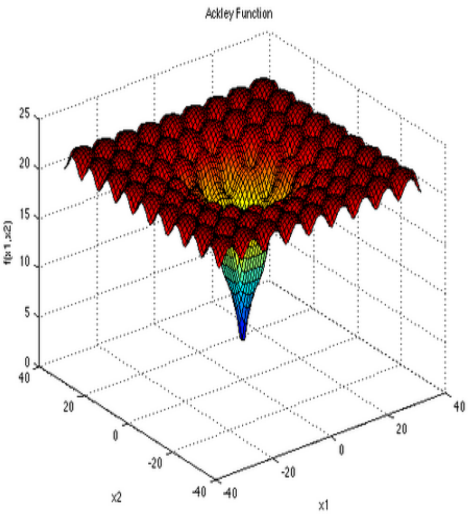

Figure 1. Function Byckley

Domain: $x_i \in [-32.768, 32.768]$, $i = 1, \ldots, d$

Global minimum: $f(x^*) = 0$, $x^* = (0,...,0)$.

Griewank function:

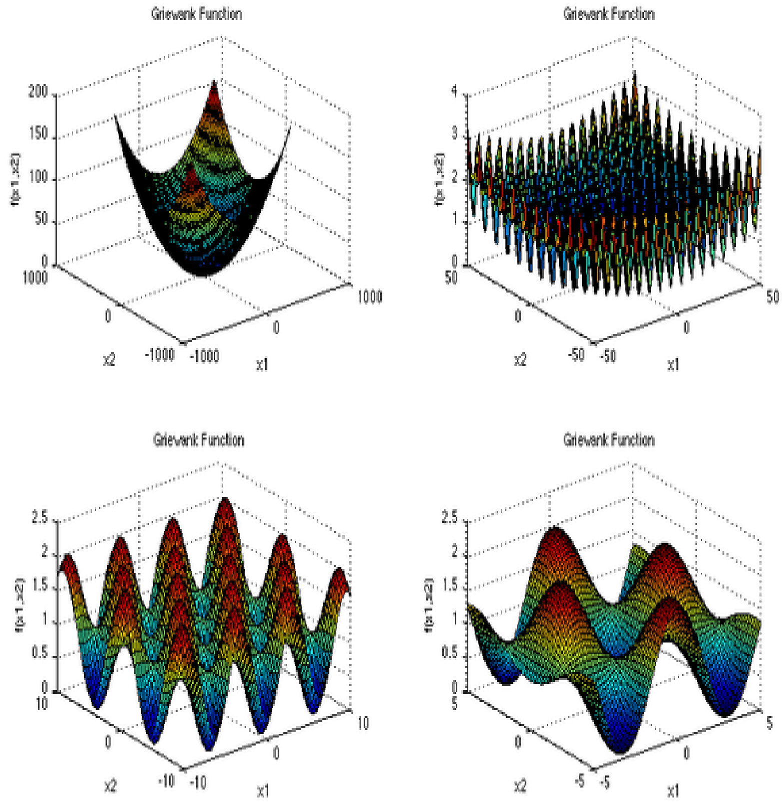

Figure 2. Griewank function

$$f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1.$$

(A) Domain: $x_i \in [-600,\ 600],\ i = 1,\ \ldots,\ d.$

(B) Global minimum: $f(x^*) = 0,\ \ x^* = (0,...,0).$

Function Levy:

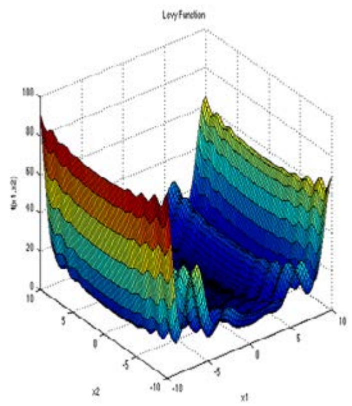

Figure 3. Function Levy

$$f(x) = \sin^2(\pi w_1) + \sum_{i=1}^{d-1}(w_i - 1)^2[1 + 10\sin^2(\pi w_i + 1)] + (w_d - 1)^2[1 + \sin^2(2\pi w_d)],$$

$$w_i = 1 + \frac{x_i - 1}{4},\ i = 1, \ldots, d$$

(C) Domain: xi∈ [-10, 10], i = 1, …, d.

(D) Global minimum: $f(x^*) = 0,\ \ x^* = (1, \ldots, 1).$

parameters to which the proposed algorithm is sensitive are: the average value of the normal distribution, which is used to generate the value β as a function of fuzzy acceptance (Fig. 4) and the standard deviation of the normal distribution, which is used in the function of fuzzy acceptance (fig. 5) for any probability of mutation.

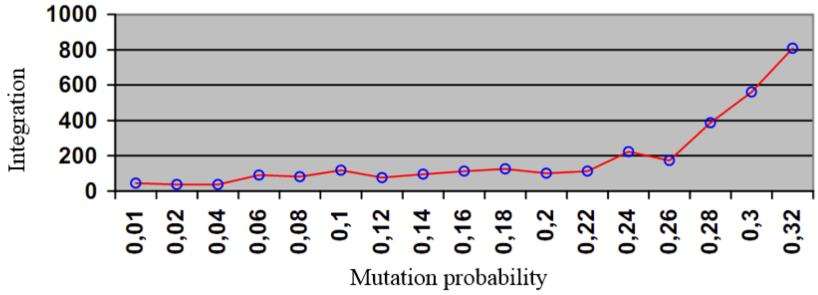

Figure 4. Average value of the normal distribution

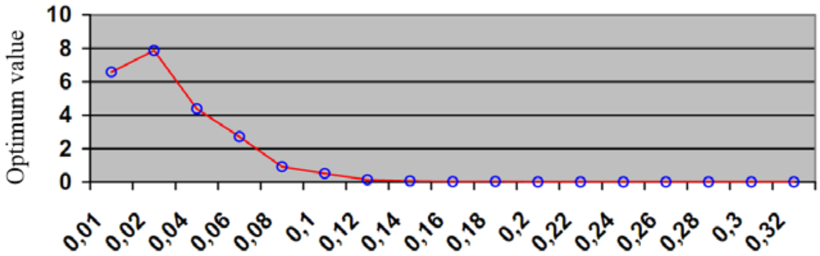

Figure 5. Standard deviation of the normal distribution

algorithm gives the best overall performance when the optimal mutation probability is from 0.17 to 0.22.

7. Conclusion

This paper presents a balanced fuzzy cultural algorithm with a customizable belief space. For the computational experiment, a new fuzzy communication algorithm was used between the population space and belief spaces using fuzzy acceptance and influence functions. The fuzzy acceptance function is responsible for selecting the best agents to create knowledge in a belief space, while the fuzzy influence function selects the best sources of knowledge to help agents in the population space.

According to the results obtained during the experiments, we can conclude that with an increase in the number of variables, the quest time for the optimum increases and the number of iterations increases. With an increase in the number of variables on average, the difference between the obtained values of the optimum for each function changes only slightly.

References

[1] Reynolds R G Chung Ch.-J.1997 *Lecture Notes on Artificial Intelligence (Springer-Verlag Press)* p 184

[2] Reynolds R G and Chung Ch-J 2000 *International Journal of Pattern Recognition and Artificial Intelligence* V 14 **1** 19
[3] Gill P E and Murray W 1986 *IMA. J. Appl. Math.* **9** 91
[4] Nelder A and Mead R1965 *Comput J* **7** 308
[5] Holland J H 1975 *University of Michigan Press, Ann Arbor, MI*
[6] Kennedy J and Eberhart R 1995 Particle swarm optimization *Proceedings of IEEE International Conference on Neural Networks* **4** 1942
[7] Karaboga D 2005 *Technical Report-TR06 Oct*
[8] Storn R Price K V 1997 *J Global Optim* **11** 341
[9] Reynolds R G 1994 *Proc. Annu. Conf. Evol. Program* 131
[10] Reynolds R G and Peng B 2005 *Journal of Mathematical Sociology* **29** 118
[11] Muhamediyeva D K 2018 *Int. J. of Mech. and Prod. Eng. Research and Develop.* **8** 555
[12] Back T 1996 *Oxford, U.K.: OxfordUniv. Press*
[13] Nguyen T T and Yao X 2008 *Int. J. Neural Syst* **18** 1
[14] Muhamediyeva D K 2018 *Int. J. of Mech. and Prod. Eng. Research and Develop.* **9** 649
[15] Daneshyari M and Yen G G 2011 *IEEE Trans. Syst., Man, Cybern. A* **18** 1
[16] Becerra R L and Coello C A C 2004 *Lect. Notes. Artif. Int.* **3315** 881
[17] Ali M Z, Awad N H, Suganthan P N and Reynolds R G 2016 *Knowledge-Based Systems* **111** 73
[18] Kim Y Cho S-B 2009 *IEEE Int. Conf. Robot. (CIRA)* 188
[19] Zadeh P M, Kobti Z 2015 *Procedia Computer Science* **52** 342
[20] Gao F and Liu H and Zhao Q 2006 *Simul. Evol. Learn. (SEAL)* V 4247 **8** 448