

# On the issue of using intuitionistic fuzzy sets for describing the expediency of solving optimization problems by genetic algorithms with given parameters

O Agibalov<sup>1</sup>, T Blinovskaya<sup>2</sup>, N Ventsov<sup>2,\*</sup>

<sup>1</sup>Kinoplan LLC, 107/1 Nansena st., Rostov-on-Don, 344038, Russia

<sup>2</sup>Don State Technical University, 1 Gagarina sq., Rostov-on-Don, 344003, Russia

**Abstract.** The paper analyses a possible option for preparing data on the results of the genetic algorithm for transfer to another subject area. It was shown that the complexity of modern target functions requires the development of new approaches to determining the parameters of search procedures. A set of experiments, each stage of which consisted of performing 100 runs of the genetic algorithm on a CPU or GPU architecture, which determines the optimal solution of the Ackley's function within a given time interval, was carried out. After the specified time interval expired, the operation of the algorithm was correctly completed by fixing the results obtained at the final iteration. The values of the absolute error were set to  $\Delta = \{0.5, 0.15, 0.1, 0.05\}$ . For each error value the number of algorithm runs, as a result of which the deviation was greater than  $\Delta$ , was determined. On the basis of the experiment carried out, fuzzy estimates of the inexpediency of searching for the optimum of the Ackley's function by the genetic algorithm on the CPU architecture in a time from 100 ms ...1800 ms were determined. The possibility of using intuitionistic fuzzy sets for describing the expediency of solving optimization problems by genetic algorithms with given parameters was shown.

## 1 Introduction

The complication of modern optimization procedures occurs both in the direction of increasing the dimensionality of the problems being solved, and due to the expansion of intervals containing potentially optimal solutions for each of the search space dimensionalities. At the same time, the introduction of even one additional dimensionality can lead to a significant disproportionate change in the structure of the search space, the emergence of new patterns inherent in the areas of the definition of target functions. In practice, such situations are possible when moving from the design of 2D circuits to the design of 3D printed circuit boards.

---

\* Corresponding author: [vencov@list.ru](mailto:vencov@list.ru)

Currently, technologies for transferring knowledge between contextually related areas of knowledge and consequently the problems solved within the boundaries or at the junctions of adjacent areas are actively used. Transfer learning (TL) [1-3] is an example of such technologies. The issues of formalizing the experience gained in analyzing the processes of solving complex problems are becoming more and more urgent. At the same time, in the context of the possible use of the experience gained in solving problems from a related subject area, it is necessary to record and investigate not only positive, but also negative experience in terms of the effectiveness of obtaining an acceptable result of knowledge.

In homogeneous transfer learning, the transfer of data features spaces in the source and target areas are represented by the same attributes ( $X_s = X_t$ ) and labels ( $Y_s = Y_t$ ), while space itself has the same dimensionality ( $d_s = d_t$ ). Another category of transfer learning is heterogeneous transfer learning. In this scenario, the spatial objects between the source and target object are not equivalent and usually do not overlap. In this case,  $X_s \neq X_t$  and/or  $Y_s \neq Y_t$  as source and target domains may not have common features and/or labels, while the sizes of feature spaces may also differ. Most methods for heterogeneous knowledge transfer are very complex and scale very poorly. Based on this, there is a need to research and develop highly efficient computing systems, as well as methods that can be applied to large data sets [2].

The work [1] considers situations related to the differences in the distributions of the initial and target data. A general Minimax Game based model for selective Transfer Learning (MGTL) was proposed. The model includes three modules: selector, discriminator and TL. These modules play a Minimax Game to help select useful inputs for transmission.

One of the distinctive features of the approach [3] is the removal of noise inherent in the original data in the process of transferring knowledge.

It is important to organize the transfer of knowledge not only within the framework of training neural networks, but also in the development of other intelligent systems, for example, genetic algorithms.

## 2 Research Issue

When solving complex optimization problems, a multilevel decomposition procedure is used – a hierarchical division of the original problem into fragments. As a rule, a prerequisite for starting the process of solving a problem is to obtain the results of solving the problems of the previous level of the hierarchy.

Thus, it is desirable that all problems of the same level of decomposition be solved in approximately the same time, while each problem must be solved with an error not exceeding a given value.

Each problem solved at the current level of the hierarchy is characterized by a set of parameters: the search space dimensionality, the sampling frequency of the domains and values, the complexity of the objective function, etc. When iteratively solving a set of problems, their conditions can be changed, associated, for example, with the search space dimensionality.

Table 1 shows the dependences of the averaged values of the absolute errors of the algorithm, which performed the search for the optimal value on CPU and GPU architectures within 1000 ms on the dimensionality of the optimized Ackley's function.

**Table 1.** Dependences of the averaged values of the absolute errors of the algorithm, which performed the search for the optimal value on CPU and GPU architectures within 1000 ms on the dimensionality of the optimized Ackley's function.

No.	Architecture	Dimensionality of the Ackley's function				
		10	15	20	25	30
1	CPU	0.02332	5.071183	38.21831	141.01	430.1684
2	GPU	0.143666	0.535208	11.81442	19.6131	68.07105

From the data presented in Table 1, it follows that the error in solving the problem using the CPU architecture in 1000 ms increases by more than 20,000 times with an increase in the dimensionality from 10 to 30, and the error in solving the problem using the GPU architecture is more than 480 times.

Due to the sharp increase in the error, questions like the following arise:

- which dimensionality of the problem on a given architecture with a given accuracy is it 1000 ms enough to solve?
- if a multiprocessor program that implements CAD functions has 1000 ms, the computational capabilities of which hardware architecture will be sufficient to obtain an acceptable solution?

### 3 Proposed Approach

Most likely, due to the stochasticity of genetic algorithms, answers to questions similar to those formulated will be not only vague, but also contradictory. Let us consider intuitionistic fuzzy sets as a data structure to formalize the answers to the formulated questions [4]. The concept of intuitionistic fuzzy sets was introduced by K. Atanassov [4] as a generalization of fuzzy sets [5]. Intuitionistic fuzzy set on the universe  $X$  is an object of the form  $\tilde{A} = \{x, \mu_{\tilde{A}}(x), \nu_{\tilde{A}}(x) | x \in X\}$ , where  $\mu_{\tilde{A}}(x) \in [0, 1]$  is called the degree of membership of  $x$  in the  $\tilde{A}$  and  $\nu_{\tilde{A}}(x) \in [0, 1]$  is the degree of non-membership of  $x$  in  $\tilde{A}$ , while the functions  $\mu_{\tilde{A}}(x)$  and  $\nu_{\tilde{A}}(x)$  satisfy the following condition [4,5]:

$$(\forall x \in X)[(\mu_{\tilde{A}}(x) + \nu_{\tilde{A}}(x)) \leq 1].$$

Then the results of a computational experiment, reflecting the feasibility of using various hardware architectures, can be specified in the form of intuitionistic fuzzy sets in the simplest case. Let  $X$  be the set of lengths of time intervals describing the time of solving the problem. Then we define an intuitionistic fuzzy set:

$$\tilde{C}^E = \{x, \mu_{\tilde{C}^E}(x), \nu_{\tilde{C}^E}(x) | x \in X\}$$

where  $\tilde{C}^E$  is a set of elements of which are tuples containing information about the time  $x$  allocated to the genetic algorithm for finding the optimum of the Ackley's function on the CPU architecture;  $\mu_{\tilde{C}^E}(x)$  is a fuzzy assessment of the expediency of searching for the optimum of the Ackley's function by the genetic algorithm on the CPU architecture in time  $x$ ;  $\nu_{\tilde{C}^E}(x)$  is a fuzzy assessment of the in expediency of searching for the optimum of the Ackley's function by the genetic algorithm on the CPU architecture in time  $x$ .

A set of experiments, each stage of which consisted of performing 100 runs of the genetic algorithm on a CPU or GPU architecture, which determines the optimal solution of the Ackley's function within a given time interval, was carried out. After the specified time interval expired, the operation of the algorithm was correctly completed by fixing the results obtained at the final iteration. The values of the absolute error were set  $\Delta = \{0.5; 0.15; 0.1; 0.05\}$ . For each error value from  $\Delta$ , the number of algorithm runs, as a result of which the deviation was greater than  $\Delta$ , was determined.

For the values of the absolute error  $\Delta = \{0.5; 0.15; 0.1; 0.05\}$ , Table 2 shows the estimates of the results of the search for the optimal value of the Ackley's function by the genetic algorithm performed on the CPU architecture are given.

**Table 2.** Dependence of the number of solutions exceeding a given error on the search time obtained when searching for the optimal value of the Ackley's function in a 10-dimensional space on a CPU architecture.

#	$\Delta$	100 ms	200 ms	400 ms	600 ms	800 ms	1000 ms	1200 ms	1400 ms	1600 ms	1800 ms
1	0.5	100	100	99	73	5	1	0	0	0	0
2	0.15	100	100	100	88	20	4	0	0	0	0
3	0.1	100	100	100	95	26	6	0	0	0	0
4	0.05	100	100	100	99	53	8	0	0	0	0

Table 2 shows the absolute deviation of the obtained solution from the optimal one. Having normalized the values from table 2, and, by interpreting the values of the elements as  $v_{\mathcal{C}^E}(x)$ , where  $x$  is the running time of the algorithm, we will get table 3.

**Table 3.** Fuzzy estimates of the inexpediency of searching for the optimum of the Ackley's function by a genetic algorithm on a CPU architecture in time of  $x$  ms.

#	$\Delta$	$v_{\mathcal{C}^E}(100)$	$v_{\mathcal{C}^E}(200)$	$v_{\mathcal{C}^E}(400)$	$v_{\mathcal{C}^E}(600)$	$v_{\mathcal{C}^E}(800)$	$v_{\mathcal{C}^E}(1000)$	$v_{\mathcal{C}^E}(1200)$	$v_{\mathcal{C}^E}(1400)$	$v_{\mathcal{C}^E}(1600)$	$v_{\mathcal{C}^E}(1800)$
1	0.5	1	1	0.99	0.73	0.05	0.01	0	0	0	0
2	0.15	1	1	1	0.88	0.2	0.04	0	0	0	0
3	0.1	1	1	1	0.95	0.26	0.06	0	0	0	0
4	0.05	1	1	1	0.99	0.53	0.08	0	0	0	0

Within the framework of the results shown in Table 3, the rationality of allocating the 1200...1800 ms algorithm is the same. However, from the point of view of saving computing resources, allocating more than 1200 ms to the algorithm is not advisable.

For this reason, we introduce additional terms of  $0 < \varepsilon_1 < \varepsilon_2 < \varepsilon_3 < 1$  that increase the estimate of the irrationality of using time intervals exceeding 1200 ms. When adding  $\varepsilon_1, \varepsilon_2, \varepsilon_3$  to the columns corresponding to work intervals of 1400...1800 ms, we get table 4.

It is more reasonable and flexible to express your judgments using two or more linguistic terms when the expert is not sure of his confidence [6-8]. To date, significant research has been carried out on hesitant fuzzy linguistic term sets (Hflts) [9]. In accordance with the Hflts approach, the result can be specified by the set of  $H_s = (1200, 1400)$ , if a system with increased reliability requirements is being built, or  $H_s = (1000, 1200)$  if the speed of the system is important.

**Table 4.** Fuzzy estimates of the inexpediency of searching for the optimum of the Ackley's function by a genetic algorithm on a CPU architecture in time of x ms added by terms of  $\varepsilon_1, \varepsilon_2, \varepsilon_3$ .

No	$\Delta$	$v_{\zeta^E}$ (100)	$v_{\zeta^E}$ (200)	$v_{\zeta^E}$ (400)	$v_{\zeta^E}$ (600)	$v_{\zeta^E}$ (800)	$v_{\zeta^E}$ (1000)	$v_{\zeta^E}$ (1200)	$v_{\zeta^E}$ (1400)	$v_{\zeta^E}$ (1600)	$v_{\zeta^E}$ (1800)
1	0.5	1	1	0.99	0.73	0.05	0.01	0	$\varepsilon_1$	$\varepsilon_2$	$\varepsilon_3$
2	0.15	1	1	1	0.88	0.2	0.04	0	$\varepsilon_1$	$\varepsilon_2$	$\varepsilon_3$
3	0.1	1	1	1	0.95	0.26	0.06	0	$\varepsilon_1$	$\varepsilon_2$	$\varepsilon_3$
4	0.05	1	1	1	0.99	0.53	0.08	0	$\varepsilon_1$	$\varepsilon_2$	$\varepsilon_3$

## 4 Conclusion

1. It was shown that the complexity of modern target functions requires the development of new approaches to determining the parameters of search procedures.
2. A set of experiments, each stage of which consisted of performing 100 runs of the genetic algorithm on a CPU or GPU architecture, which determines the optimal solution of the Ackley's function within a given time interval, was carried out. After the specified time interval expired, the operation of the algorithm was correctly completed by fixing the results obtained at the final iteration. The values of the absolute error were set  $\Delta = \{0.5; 0.15; 0.1; 0.05\}$ . For each error value from  $\Delta$ , the number of algorithm runs, as a result of which the deviation was greater than  $\Delta$ , was determined.
3. The possibility of using intuitionistic fuzzy sets for describing the expediency of solving optimization problems by genetic algorithms with given parameters was shown.

## Acknowledgments

This study was supported by the Russian Fundamental Research Fund, projects Nos 19-01-00357, 18-01-00314.

## References

1. Wang B, Qiu M, Wang X, Li Y, et al. 2019 *KDD 2019 - Proceedings of the 25th ACM SIGKDD on Knowledge Discovery and Data Mining* p 34
2. Day O, Khoshgoftaar T M 2017 *Journal of Big Data* **4(1)** 29
3. Zhang L, Yang J, Zhang D 2017 *Information Sciences* **418-419** 242
4. Atanassov K T 1983 *Proceedings of VII ITKR's Session, Sofia, Central Science and Technical Library, Bulgarian Academy of Sciences* **1697/84** 6-24
5. Bozhenyuk A V, Belyakov S L, Kosenko O V, Alekhina O M 2019 *Engineering Bulletin of the Don* 3 ivdonru/ru/magazine/archive/n3y2019/5821
6. Liao H et al 2014 *Knowl Based Syst* <http://dx.doi.org/10.1016/j.knosys.2014.12.009>
7. Song Y *PLoS ONE* **13(12)** e0208855 <https://doi.org/10.1371/journal>
8. Liu Hongbin, Rodríguez R M 2014 *Information Sciences* **258** 220
9. Wang Y, Tian L, Wu Z 2020 *Expert Systems with Applications* doi: <https://doi.org/10.1016/j.eswa.2020.11381>