Application of bioinspired algorithms for solving transcomputational tasks

V V Kureichik¹, I O Kursitys¹, E V Kuliev¹ and E M Gerasimenko¹

¹Southern Federal University, Rostov-on-Don, 344006, Russia

E-mail: vkur@sfedu.ru, i.kursitys@mail.ru, elmar_2005@mail.ru, egerasimenko@sfedu.ru

Abstract. The paper investigates the pattern recognition task as one of the most important transcomputational problems. Pattern recognition is applied for statistical data analysis, signal processing, image analysis, bioinformatics, machine learning, and many other fields. The area of application is related to automated pattern detection in the data using computer algorithms and data classification in terms of different categories. The development of new ideas in the processing of large volumes of information creates a new trend in the pattern recognition area. A specific field is formed by the bioinspired algorithms as mathematical reorganization, that transform the initial information into the results based on simulating the evolution methods, natural analogies, and statistic approach. In terms of computer modelling based on the bioinspired algorithms, we can create and develop difficult concepts, for which we have no analytical description. The authors present the optimal bioinspired algorithms based on the ant, monkey, and bat behavior in nature and develop software based on the principle of pattern recognition using the uploaded photo and downloading the results into the file. The experiments demonstrate the effectiveness of the proposed approach.

1. Introduction

The popular computational algorithms are believed to work with finite objects only. The finite objects are considered as objects which can be effectively encoded with words of a certain alphabet (without generality constraints, in a binary way). The encoding effectiveness is characterized by the ability to determine the key features using the object code. For example, the code of the graph allows us to calculate the number of vertexes and connections, and the code of matrix can give us its size and number in any cell. Generally, encoding is quite a trivial process: the algorithms can work with the strings that describe the objects. Natural numbers can be represented with binary or decimal notation; vectors and matrices are encoded as sequences of numbers with dividers; graph can be denoted as a list of edges or adjacency matrix, etc. In terms of all mentioned examples the objects are encoded as words of a certain alphabet, and the object code cannot be represented separately [1].

The same task can be solved by different algorithms with different complexity. It is expected that there is the optimal algorithm for each task, and its complexity determines the task complexity. Unfortunately, in some cases, there can be no optimal algorithm: small adjustments can reduce the complexity a little, but the limit complexity cannot be achieved by any of the existing algorithms. Blum’s theorem shows that such a situation can appear for some problems of any complexity measure [2].

Thus, we always talk about task complexity in terms of high and low assessments. As a rule, if the algorithm can solve the task with the complexity $O(g(n))$, the task is included in the class of such
complexity. More rarely, it is possible to obtain a non-trivial low assessment and prove that any algorithm that solves the task is included in the class $\Omega(h(n))$ [3,4].

Transcomputational tasks are referred to as such kind of problems. The concept “transcomputational” was introduced by Bremerman. Solving the transcomputational tasks requires processing more than 1093 bits of information. The number 1093 is called the Bremermann's limit. It represents the common number of bits that can be processed by the computer of the Earth’s size (hypothetically) during the time that is equal to the Earth’s age. Any number that is larger than the Bremermann's limit is called transcomputational number [5,6].

1.1. Examples of transcomputational tasks

1.1.1. Integrated circuits testing. Comprehensive testing of all combinations in the integrated circuits having 309 inputs and 1 output requires testing 2309 combinations of the inputs. Since the number 2309 is considered transcomputational (more than 1093), the task of testing such a system of the integrated circuits involves solving transcomputational tasks. There is no method to test the circuit for all combinations of the input.

1.1.2. Pattern recognition. Let us consider the array of $\sigma \times \sigma$ size and chess type, where each square can be assigned with one of K colors. Generally, there are $K \times n$ color patterns, where $n = \sigma^2$. The task is to define the best classification of the models according to a certain criterion. The task can be solved by searching for all possible coloring. In terms of two colors, the search becomes transcomputational, where the arrays’ size is $18 \times 18$ or bigger. For matrix $10 \times 10$, the problem becomes transcomputational, when there are 9 colors and more [7].

2. Analysis of the pattern recognition task

Pattern recognition is considered as recognition of dependencies in the data. This method is applied for the statistical analysis of the data, signal processing, image analysis, information search, bioinformatics, data compaction, computer graphics, machine learning, etc. Pattern recognition is originated from the statistics and engineering; some of the modern approaches to pattern recognition include using machine learning due to the increased access to big data and new diversity of computational performance. However, these fields can be considered as two aspects of the same application area, and they have been significantly developed over the past decades.

The area of pattern recognition is related to the automated discovery of the dependencies in the data using computer algorithms. The mentioned dependencies are applied for solving such tasks as classification according to several categories. [8]

Pattern recognition is a scientific procedure that focuses on the automated discovery of the dependencies and relations in the input patterns. After that, the algorithm assigns a mark or real-valued output for each input pattern. The input patterns can be represented on the basis of the main purpose of the procedure such as events, measurements, processes, or captures of any type than need to be classified or estimated. Today, the need for the practical functioning of pattern recognition has been increased together with the advanced processors. The introduction of new concepts in the industrial systems and processing of large scopes of information produces a new branch of pattern recognition. The development and implementation of the pattern recognition systems prove the existence of the growing trend in the field of its application and research. The pattern recognition systems can be classified according to the application areas. For instance, the document image classification in terms of the optical pattern recognition; identification of personality or fingerprint in terms of biometrical diagnostics; automated diagnostics in medicine; automated target recognition in the military sphere; analysis in bioinformatics; object sorting in the industrial automatization; security of electric power system in the power electrical engineering, and so on. The pattern recognition problems have four common solutions that can be divided into four classes: statistical, structural, hybrid methods of statistical, structural, and
artificial intelligence. The last method is recommended for more complex tasks with no linear and direct relation between the patterns and the target vector.

2.1. Problem formulation
The section describes the model of pattern recognition used in the paper.

Let $\Omega$ denote the set of recognized objects (pattern space); $\omega: \omega \in W$ denote the recognized object (pattern); $g(\omega) : W \rightarrow M, M = \{1, 2, \ldots, m\}$ be the indicating function that divides the pattern space $\Omega$ into $m$ non-overlapping classes $\Omega_1, \Omega_2, \ldots, \Omega_m$. The indicating function is unknown; $X$ denotes the observation space perceived by the observer (feature space); $x(\omega) : \Omega \rightarrow X$ be the function that matches each object $\omega$ with the point $x(\omega)$ in the feature space. Vector $x(\omega)$ is the object pattern perceived by the observer [9].

In the feature space, non-overlapping sets of points $K_i \in X, i = 1, 2, \ldots, m$ corresponding to patterns from the same class.

The decision rule e.g. assessment of $g(\omega)$ based on $x(\omega)$ as $\hat{g}(x) = \hat{g}(x(\omega))$ is represented by the function $\hat{g}(x) : X \rightarrow M$.

Let us denote the available information on functions $g(\omega)$ and $x(\omega)$ as $x_j = x(\omega_j), j = 1, 2, \ldots, N$ as the functions are unknown. Thus, the set of precedents can be represented as $(g_j, x_j), j = 1, 2, \ldots, N$.

The task includes building such decision rule $\hat{g}(x)$, that can provide recognition with minimal mistakes (figure 1). The general case is to consider the feature space as Euclidean i.e. $X = R_l$. The quality of the decision rule is measured by the frequency of good solutions occurrence. It is usually estimated by assigning a probabilistic measure to the object set $\Omega$. Thus, the task is reduced to the minimization of $P\{ \hat{g}(x(\omega)) \neq g(\omega)\}$ [10].

![Figure 1. Model of the pattern recognition task.](image)

3. Algorithms for solving the pattern recognition task
Different approaches to solving the studied task assume using different methods for searching optimal decisions in a certain sequence. All methods deal with the statistical problem statement and objective i.e. maximization of the system capacity. Many optimization algorithms raise the problem of choosing the best optimization method. It seems that we still do not have the answer. If there was the ultimate optimization algorithm, all other ones would be unnecessary. No doubt that there is no universal optimization algorithm that would be able to solve all real problems. All the existing algorithms can be used only without considering the specific areas of application. In terms of the nature of a concrete problem, we can use different optimization algorithms that can give the most beneficial solutions.

In recent days, the methods of bioinspired search have become very popular for solving transcomputational tasks.

Inspired by animals’ behavior in the biological systems, the researchers in the area of computational paradigms have emulated the intelligent bioinspired processes in the form of computational algorithms. Their purpose was to simulate the natural benefits of biological systems for solving complex modeling tasks and optimization problems. Special emphasis is given to such optimization problems, which complexity has provided a rich background for developing different heuristic methods based on the populations. Each of the mentioned methods is characterized by a different mix of computational effectiveness and quality of the obtained solutions. The first contributions in the studied area were based
on observing and imitating Darwin’s evolutionary ideas. Today, the number of bioinspired researches is growing rapidly, and the ideas are based on the animals’ intelligent behavior [10].

Let us present the review of some popular bioinspired algorithms.

3.1. Ant colony optimization
Ant colony optimization (ACO) algorithm was introduced by Marco Dorigo in his Ph.D. thesis in the 90s. This algorithm is based on ant behavior while searching for the path between the colony and the food source. It was firstly used for solving the popular traveling salesman problem. Later is demonstrated its effectiveness in terms of solving other complex optimization tasks.

Ants are socialized insects that live in the colonies. Their behavior is controlled by the purpose of finding the food. During the search process, the ants move into the colonies to find the food and leave their organic compound called pheromone. They communicate with each other by these pheromone traces. When an ant finds some amount of food, it takes it as much as it can. When it arrives back, the pheromone is left on the path depending on the amount and quality of food. Other ants can feel the left pheromone and follow the corresponding path. The more pheromone is left, the higher the probability of choosing this path is. The more ants follow the path, the more pheromone is left on this path [11].

Let there be two paths to reach the food. At first, there is no pheromone on the ground, and the probability of choosing one of the paths is 50%. Two ants choose different paths to reach the food since the probability is 50/50.

The distances between the colony and food of the mentioned paths are different. After finding the food, the ant carries it back to the colony. While going back, it leaves a pheromone trace on the ground. The shorter the path is, the faster the ant will arrive.

When the third ant is going to search for the food, it will follow the shortest path, which depends on the level of pheromones. Since the shorter path has more pheromones than the longer one, the third ant will follow the path with a higher level of pheromones.

After this process is repeated several times, the shortest path will have more pheromones and a higher probability to be chosen. The next iteration, all ants will follow the shortest path [12].

3.2. Monkey algorithm
The term «behavior» is related to the models of animal movements such as jumping, running, crawling, climbing, etc. These actions also include movements in the space and are common for the animals during feeding (stillness, freezing, color-changing). The monkey search algorithm (MSA) was introduced by R. Zhao and W Tang in 2007 on the basis of the monkey behavior while searching for food and climbing up. The algorithm assumes that the higher the altitude is, the more food is at the peak. From the initial position, each monkey climbs up until it reaches the peak. Then, it makes several random local jumps hoping to find the maximal altitude and climbs up again. After researching this area, the monkey makes a global jump to investigate another part of the searching space. The described actions are repeated several times according to a predetermined number of iterations. The solutions are represented by the highest found peaks. Climbing up is interpreted as a local search and can be implemented by different methods. The number of local and global jumps is determined by the decision-maker or randomly. The intervals of the jumps (low and high limit) can also be set by the researcher [13].

The local jump is checked if it gets in the tolerance interval of the search space. If the obtained solution is located in the valid range, the algorithm is finished. Otherwise, it is returned back to the previous step. Note that if the value of the solution is positive, the agent jumps to the gravity center; if it is negative, the agent jumps on the contrary direction.

The required solution is represented by the maximal amount of food, and the branches of the tree are considered as options between the nearest solutions. The main idea of the algorithm includes the following statement: if the monkey is located at the end of any branch, it can jump to the left or right branch with an equal probability. At the search point which corresponds to the end of the branch, the algorithm calculated the value of the objective function. The moving stops when the monkey reaches the maximum peak of the tree. All the branches that were investigated by the monkey are saved in the
list. If the monkey has not studied all the branches, it reaches the peak and falls down the current best point and starts climbing up again bypassing some of the investigated branches [14].

3.3. Bat Algorithm
Bat Algorithm (BA) was introduced by X.-Sh. Yang in 2010 [15]. BA is considered as a metaheuristic algorithm which allows it not to be assigned to the specific task rather than heuristic ones.

Bat species have the perfect echolocation mechanism that helps them to find the prey and obstacles and sit on the blocks in the complete darkness. They determine the distance to the object by calculating the time between the pulse and its reflection.

Bat algorithm works according to the following rules:

- Bats use echolocation to determine the distance, find the prey and obstacles.
- Bats move randomly with a certain velocity \( v_i \), near the position \( S_i \), with a fixed frequency from \([f_{max}, f_{min}]\). The regulated parameters are wavelength \( \lambda \), pulse rate \( r \) \([0;1]\), and loudness \( a \).
- The loudness is changed from the maximum value \( a_{max} \) to the minimal (constant) value \( a_{min} \).

The algorithm work can be described using the following steps:

1. To initialize the bat population at the position \( S_i \) \((i = 1, 2 \ldots n)\) and their velocity \( v_i \).
2. To initialize the frequency \( f_i \) at the position \( S_i \).
3. To determine the initial pulse rate \( r_i \) and loudness \( a_i \).
4. To calculate the distance between the initial and final values:
   \[ d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \]  
   where \( x_i, y_i \) denote the start coordinates; \( x_2, y_2 \) denote the finish coordinates.
5. To calculate the frequency for new solutions according to the following formula:
   \[ f_i = f_{min} + (f_{max} - f_{min}) \cdot U(0,1), \]  
   where \( U \) is a random value.
6. To update the velocity and global best solution.
7. To calculate the local best solution for each bat species.

At the stage of initializing the bat population, the initial values of the loudness \( a \), frequency \( f \), and pulse rate \( r \) are represented as the values from the corresponding intervals: \([a_{min}, a_{max}], [f_{max}, f_{min}], [r_{min}, r_{max}]\). A new solution is obtained by the random flight of the bats near the best solution. After finding the obstruction, the algorithm decreases the loudness and increases the frequency of the ultrasonic pulse [16].

4. Experimental research.
Based on the considered bioinspired algorithms, the authors developed software for pattern recognition to implement the proposed bioinspired techniques [17-19].

The developed software is based on the pattern recognition idea on the uploading photo and downloading the results in the form of the file. The program work is to compare the uploaded photos and output the solution in the resulting file.

As the problem is solved, we compare the estimated areas of the images and analyze them (figures 2 and 3).
Figure 2. Conditions for meeting the association rules:
(a) - similarity; (b) - difference; (c) – common context.

1. Monkey algorithm
2. Bat algorithm
3. Ant algorithm

Figure 3. Histogram of the objective function values.

All considered algorithms are good at solving the stated task. In terms of a bigger number of studied elements, the bat algorithm shows more accurate solutions. However, its execution time is quite big. Conversely, the ant colony algorithm can demonstrate better speed but much worse results.

5. Conclusion
In conclusion, we can say that today there is no single perfect bioinspired algorithm for the optimization tasks. Thus, bioinspired algorithms are actively developed, analyzed, and researched today. In the paper, we compared different popular algorithms and discovered their benefits and shortcomings. It is reasonable to combine the algorithms to use the advantages of several of them simultaneously. This can help obtain the most accurate results at an acceptable time spent. Such a method is called “hybridization” or “combination”. Our future researches will be related to the development and analysis of the hybrid bioinspired algorithms.
Acknowledgements.
The reported study was funded by RFBR according to the research project 19-01-00559 and №19-07-00099.

References
[1] Kacprzyk J, Kureichik V M, Malioukov S P, Kureichik V V and Malioukov A S 2009 General questions of automated design and engineering Studies in Computational Intelligence 212 1-22
[2] Alpert C J, Mehta D P and Sapatnekar S S 2009 Handbook of Algorithms for Physical Design Automation (Auerbach Publications) 1044
[3] Amin A 2000 Recognition of printed Arabic text based on global features and decision tree learning techniques, Pattern Recognition 33 1309–1323
[4] Anagun A S 1998 A neural network applied to pattern recognition in statistical process control. Computers Industrial Engineering 35 185–188
[5] Yegananarayana B 1999 Artificial neural network (Prentice-Hall India)
[6] Demuth H and Beale M 1998 Neural Network Toolbox User’s Guide (Natick, MA: Math Works)
[7] Hassan A, Nabi Baksh M S, Shaharoun A M and Jamaluddin H 2003 Improved SPC chart pattern recognition using statistical features International Journal of Production Research 41(7) 1587–1603
[8] Montgomery D C 2001 Introduction to Statistical Quality Control (4th ed.) (New York: Wiley)
[9] Indra Kiran N V N, Pramila Devi M and Vijaya Lakshmi G 2010 Effective control chart pattern recognition using artificial neural networks International Journal of Computer Science and Network Security 10(3) 194-199
[10] Ruiz L R 2010 Interactive Pattern Recognition applied to Natural Language Processing (Thesis)
[11] Zaporozhets D, Zaruba D and Kuliev E 2018 Ant Algorithm for Determining of Critical Connections in VLSI Proceedings of 2018 IEEE East-West Design and Test Symposium, EWDTS 2018 10.1109/EWDTS.2018.8524709
[12] Gladkova L A, Gladkova N V and Legebokov A A 2015 Organization of knowledge management based on hybrid intelligent methods Advances in Intelligent Systems and Computing 349 107-112
[13] Kuliev E, Kureichik V and Kureichik V 2018 Monkey search algorithm for ECE components partitioning Journal of Physics: Conference Series Cep. "International Conference Information Technologies in Business and Industry 2018 - Enterprise Information Systems" DOI: 10.1088/1742-6596/1015/4/042026
[14] Yang X S 2010 Nature-Inspired Metaheuristic Algorithms 2nd Edition (Luniver Press)
[15] Tsai P W, Pan J S, Liao B Y, Tsai M J and Istanda V 2012 Bat algorithm inspired algorithm for solving numerical optimization problems Applied Mechanics and Materials 134–137
[16] Kar A K 2016 Bio inspired computing - A review of algorithms and scope of applications Expert Systems with Applications 59 20-32
[17] Kuliev E, Zaporozhets D, Kravchenko Y and Kursitisi I 2021 A Combined Bioinspired Algorithm for Big Data Processing Advances in Intelligent Systems and Computing
[18] Kureichik L, Kureichik V, Kureichik VI, Leschanov D and Zaruba D. 2018 Hybrid approach for VLSI fragments placement Advances in Intelligent Systems and Computing 679 349-358
[19] Kacprzyk J, Kureichik V M, Malioukov S P, Kureichik V V and Malioukov A S 2009 Experimental investigation of algorithms developed Studies in Computational Intelligence 212 211-223, 227-236