Parallel approach for bioinspired algorithms

Dmitry Zaporozhets, Daria Zaruba, Nina Kulieva

Southern Federal University, 105/42 Bolshaya Sadovaya Str., Rostov-on-Don, 344006, Russia
E-mail: elpilasgsm@gmail.com, daria.zaruba@gmail.com, holopova@sfedu.ru

Abstract. In the paper, a probabilistic parallel approach based on the population heuristic, such as a genetic algorithm, is suggested. The authors proposed using a multithreading approach at the micro level at which new alternative solutions are generated. On each iteration, several threads that independently used the same population to generate new solutions can be started. After the work of all threads, a selection operator combines obtained results in the new population. To confirm the effectiveness of the suggested approach, the authors have developed software on the basis of which experimental computations can be carried out. The authors have considered a classic optimization problem – finding a Hamiltonian cycle in a graph. Experiments show that due to the parallel approach at the micro level, increment of running speed can be obtained on graphs with 250 and more vertices.

1. Introduction
In the mid-1980s, to increase the speed of solution of the NP-hard problems of optimization, stochastic search methods, based on natural phenomena, were investigated [1, 2]. This class of algorithms, named as bioinspired ones, processes a set of alternative solutions and is alternative to classic search algorithms working with only one solution. Bioinspired algorithms belong to a class of heuristic algorithms, for which the convergence to a global optimum has not been theoretically proved, but it has been empirically established that the probability of obtaining an optimal or quasi-optimal solution is high [1, 3].

In the paper, the authors suggest a probabilistic parallel approach on the basis of a bioinspired algorithm. To estimate the algorithm’s quality and compare obtained results with another swarm intelligence algorithms, the authors have developed software to conduct a computational experiment. On the basis of these researches, the authors can confirm that the time complexity of the developed algorithm has polynomial complexity.

2. Common scheme of bioinspired algorithms
In general case, all bioinspired approaches involve the following stages [2]:
1. Generation of initial alternative solutions. Within given search area, a set of alternative solutions, which are approximated to the desired solution, is generated. To create the initial population, the authors use deterministic and random algorithms [3]. Solutions cover the entire search area, i.e. breadth-first search is implemented. If a neighbourhood of a global extremum is known, then agents fill out the entire neighbourhood and search is executed near a global extremum of an optimized function (depth-first search). Such approach can reduce running
time of the algorithm. But during a real problem solution, a global extremum is unknown and agents are evenly distributed within the entire search space.

2. Generation of new alternative solutions. New solutions are generated with the use of a set of operators and each new solution tends to optimum iteratively. A set of operators, as mathematical transformations based on various probabilistic approaches, is a peculiarity of each bioinspired algorithm. For example, for a bacterial algorithm, it is a simulation of E.coli bacterium movement; for genetic algorithms (GA) – crossing of alternative solutions (crossingover) and mutation (sudden changes in one or more parts of the alternative solution); for ant colony optimization algorithms (ACO) – simulation of ants’ behaviour [4-6].

3. Selection of obtained solutions. After generation of new alternative solutions, a size of a population is increased several times (depending on configuration parameters). In a general case, solutions with a low objective function (OF) rating and illegitimate solutions can be included in the population. Growth of the population leads to degradation of an algorithm convergence. To solve these problems, the authors introduce a selection stage. Here, different heuristics for selection of the optimal solutions is used. A good example is a “greed” heuristics which selects only the best solutions, but this leads to population degeneration. Today, there are a lot of selection operators known, not only universal, but specialized for concrete problems.

4. Search termination. Verification of a stop criterion is performed. The best obtained solution is taken as an approximate problem solution. There are the following widespread stop criteria that are known: a number of iterations and computational time. Also, it is frequently used as a stop criterion, an algorithm stagnancy– the best obtained quazi-optimal solution is permanent during given quantity of iterations or time.

One of the main advantages of bioinspired algorithms is its modular structure which allows one to obtain a great number of new variation algorithms by development of new and a modification of existing rules for initialization and generation of new agents.

In bioinspired algorithms, all agents are characterized by the following features [7]:

- separateness – in search space, all agents can move independently of each other;
- stochastic behavior – a generation of new solutions is random to some extent;
- connectivity – agents can exchange information identified during a search process;
- limitation of conceptualization – each agent has information about an investigated part of search space;
- decentralization – absence of high level agents to control a search process.

In comparison with classic optimization methods, bioinspired optimization algorithms have advantages especially for nonformilized and high dimensional problems. In such case, bioinspired algorithms provide with high probability of finding an optimal or quazi-optimal solution in polynomial time.

Nevertheless, in case of big amount of input data, execution time increases considerably. It is connected with time complexity of these algorithms, which can vary from $O(n^2)$ to $O(n^3)$.

Figure 1 shows a general scheme of a bioinspired algorithm.
3. Architecture of parallel search

In the paper, a parallel search method is suggested to speed up bioinspired algorithms. In particular, a modified parallel genetic algorithm based on a simple genetic algorithm is considered for a classic optimization problem – a salesmen problem.

In general case, with respect to bioinspired algorithms, there are 3 levels of paralleling:

1. Microlevel – problem paralleling at the level of generation of new alternative solutions;
2. Macrolevel – paralleling of search at the level of iteration;
3. Metalevel – paralleling of search at the level of an isolated set of alternative solutions (populations). In the genetic algorithms paradigm, this approach is successfully implemented through a migration operator [7-10].

The authors suggest a realization of the parallel genetic algorithm with paralleling at the macrolevel. An architecture of the parallel search method is represented in Figure 2.

The algorithm starts with a generation of a new population of alternative solutions. Next, at each iteration, N threads are initialized that include crossover, mutation and inversion operators. For each new solutions, the OF value is calculated in the thread, in which the given solution is obtained. The algorithm waits a termination of all threads. Then, all new solutions are combined in the population and a selection is performed. The process continues iteratively until the predetermined number of iterations will be reached. As a mechanism of a new alternative solution generation, a cyclic crossover operator, a one-point mutation operator and an inversion with a choice of begin and end in the inversion block are used.
4. Experiments

To confirm the effectiveness of the suggested approach, the authors have developed the software to implement the parallel genetic algorithm. A classic optimization problem – finding Hamiltonian cycle in a graph – was considered as an experimental problem [14]. The main aim of experiments is to determine a genetic algorithm work, and to compare it with a classic “sequential” genetic algorithm. Research was carried out on two different platforms: Intel I7 – 4 cores, 2.7 GHz and AMD 6000+, - 2 cores – 2.5 GHz. Experimental results are summarized in Table 1.

Comparison of the running time of parallel and classic genetic algorithms is represented in Figures 3 and 4. Figure 5 shows a histogram of speed increment of the parallel genetic algorithm with respect to the classic genetic algorithm.

Figure 5 shows that due to the parallel approach at the micro level, increment of running speed is on graphs with 250 and more vertices. So, this approach can be effectively applied for high dimension optimization problems.
Table 1. Experimental results

| AMD 6000+, 2 Cores, 3GHz | Core i7, 4 Cores, 2.2GHz |
|-------------------------|--------------------------|
| **Parallel GA** | **Classic GA** | **Parallel GA** | **Classic GA** |
| Running time, ms | OF value, c.u. | Running time, ms | OF value, c.u. | Running time, ms | OF value, c.u. | Running time, ms | OF value, c.u. |
| 3436 | 11274 | 666 | 11987 | 1412 | 10703 | 428 | 11113 |
| 3370 | 26378 | 1229 | 25877 | 1589 | 26975 | 844 | 26490 |
| 5766 | 88347 | 5743 | 91281 | 2559 | 89157 | 3461 | 91041 |
| 13662 | 206367 | 17841 | 202756 | 5702 | 205833 | 12492 | 206268 |
| 46885 | 446725 | 82719 | 430594 | 17075 | 441331 | 48029 | 446612 |
| 190723 | 927196 | 412912 | 926242 | 57052 | 936829 | 189073 | 932780 |
| 357058 | 1182746 | 727041 | 1186145 | 91067 | 1192587 | 298450 | 1187945 |
| **Figure 3.** Experimental results on AMD 6000+, 2 Cores, 3GHz |

**Figure 4.** Experimental results on Core i7, 4 Cores, 2.2GHz.

**Figure 5.** Increment of running speed of the parallel GA to the classic GA.
5. Conclusion
The authors described the new approach for parallel bioinspired algorithms. The bioheuristic general structure was suggested, as well as different parallelization levels were described. As an example, the classic GA and its parallel modification were considered. For the salesman problem (Hamiltonian cycle in the graph), the parallel GA with parallelization at the micro level was realized. Experiments were performed and showed a significant increase in the speed of the algorithm with a number of vertices in the graph more than 250. The speed of the algorithm increased by 100 - 200% in comparison with the classical implementation, depending on the used processor.

References
[1] Kureichik V V, Kureichik V M, Malioukov S P, Malioukov A S 2009 Algorithms for Applied CAD Problems (Berlin Heidelberg: Springer-Verlag)
[2] Alpert C J, Dinesh P M, Sachin S S 2009 Handbook of Algorithms for Physical design Automation (USA: Auerbach Publications Taylor & Francis Group)
[3] Zaruba D, Zaporozhets D, Kureichik V 2016 Artificial bee colony algorithm—A novel tool for VLSI placement Advances in Intelligent Systems and Computing 450 433-442
[4] Zaruba D, Zaporozhets D, Kureichik V 2016 VLSI placement problem based on ant colony optimization algorithm Advances in Intelligent Systems and Computing 464 127-133
[5] Gladkov L A, Kureichik V V, Kravchenko Y A 2013 Evolutionary algorithm for extremal subsets comprehension in graphs World Applied Sciences Journal 27 (9) 1212-1217
[6] Kureichik V, Kureichik V, Bova V 2016 Placement of VLSI fragments based on a multilayered approach Advances in Intelligent Systems and Computing 464 181-190
[7] Kureichik V V, Kravchenko Y A 2013 Bioinspired algorithm applied to solve the travelling salesman problem World Applied Sciences Journal 22 (12) 1789-1797
[8] Kar A K 2016 Bio inspired computing - A review of algorithms and scope of applications Expert Systems with Applications 59 20-32
[9] Lim S K 2008 Practical Problems in VLSI Physical Design Automation (Germany: Springer Science + Business Media B.V)
[10] Yang C, Ji J, Liu J, Yin B 2016 Bacterial foraging optimization using novel chemotaxis and conjugation strategies Information Sciences 363 72-95
[11] Zhao W, Wang L 2016 An effective bacterial foraging optimizer for global optimization Information Sciences 329 719-735
[12] Kureichik Jr V, Kureichik V, Bova V 2016 Placement of VLSI fragments based on a multilayered approach Advances in Intelligent Systems and Computing 464 181-190
[13] Kacprzyk J, Kureichik V M, Malioukov S P, Kureichik V V, Malioukov A S 2009 General questions of automated design and engineering Studies in Computational Intelligence 212 1-22
[14] Hernández-Ocaña B, Mezura-Montes E, Pozos-Parra M D P 2016 Evolutionary Bacterial Foraging Algorithm to solve constraint numerical optimization problems CEUR Workshop Proceedings 1659 58-65