Eduard I. Vatutin*[1]*, Vladimir S. Panishchev*[2]*,
***Svetlana N. Gvozdeva*[3]*, Vitaly S. Titov*[4]**
Southwest State University, Kursk, Russia
[1]evatutin@rambler.ru, [2]gskunk@yandex.ru, [3]svetka-gvozdeva@yandex.ru, [4]titov-kstu@rambler.ru

DOI: 10.25045/jpit.v11.i1.01

COMPARISON OF DECISIONS QUALITY OF HEURISTIC METHODS BASED ON MODIFYING OPERATIONS IN THE GRAPH SHORTEST PATH PROBLEM

Received: 24.01.2019 Revised: 26.03.2019 Accepted: 24.06.2019

The article deals with the problem of the analysis of effectiveness of the heuristic methods based on the modification of earlier found decisions in the test problem for getting the shortest path in graph. The article briefly describes the selected group of methods used to solve the problem. The methodology considers the experimental comparison for estimating the quality of solutions based on the performance of computational experiments with the samples of pseudo-randomly structured graphs that uses the BOINC platform. It also presents the description of obtained experimental results which allow to identify the areas of the preferable usage of selected subset of methods depending on the size of the problem and power of constraints. It is shown that the particle swarm optimization, random walks, simulated annealing and bee colony methods are ineffective in the selected problem and significantly inferior to the quality of solutions that are provided by ant colony optimization method and genetic algorithms.

Keywords: *heuristic methods, genetic algorithms, particle swarm optimization, random walks, simulated an-nealing, bee colony method, shortest path problem, discrete combinatorial optimization, BOINC.*

1. Introduction

There is a wide class of optimization problems with restrictions on discrete values of solution elements (arguments of the objective function) [1, 2]. These include many problems from combinatorics, graph theory and operations research. Mathematical apparatus of derivatives and gradients is not applicable for them, but it is successfully used in solving the problems of continuous optimization [3]. To solve some of them, exact methods with polynomial time asymptotic (for example, methods of Prim and Kruskal [4, 5] for getting minimal spanning tree, Kuhn–Munkres method for assignment problem solving [6, 7], etc.) are well known, while other problems belonging to the *NP* class don't include similar methods. In order to solve them, heuristic methods are successfully used in practice. They provide decisions of good quality for reasonable computing time.

Available methods for solving discrete optimization problems can be divided into the following classes:

- limited-search methods (Brute Force, branch and bound method [8], modifications of the Brute Force with limited number of branches or limited depth of the analyzed subtree within the combinatorial search tree, SAT-based methods [9, 10]);
- methods for sequential formation of the decision (greedy methods, random and weighted random search methods, ant colony optimization method [11–13]);
- methods based on the modification of the current decision or set of decisions using some rules that are specific to the problem being solved (random walks method, simulated annealing method, evolutionary (genetic) methods, bee colony method [14–17]);
- methods based on the movement in arguments space with subsequent mapping of the current agent position to one of the decisions of discrete optimization problem (particle swarm optimization method, firefly method, fish school search method, gravitational search method, etc. [3, 14, 18, 19]).

Limited Brute Force methods' cluster has exponential or factorial asymptotic time complexity in the common case, while other methods have polynomial asymptotic time complexity. By combining the elementary methods that are shown above, it is possible to construct hybrid multistep methods [3].

The article [20] discusses in detail the features of using heuristic methods with sequential formation of the decision in the graph's shortest path problem. The article [21] describes the application of methods based on limited Brute Force strategies. This article aims at the analysis of the decisions' quality for methods that are based on the set of modifying operations for selected problem.

2. Statement of the Problem

Different heuristic methods are characterized in practice both by the different costs of computer time and by varying quality of the decisions obtained. In this view, it is interesting to compare the quality of decisions aiming to select the best methods. For this purpose a test problem of getting shortest path $P(G)=\left[a_{i_1}=a_{beg}, a_{i_2}, ..., a_{i_Q}=a_{end}\right]$ in graph $G=\langle A,V\rangle$, is selected where $A=\{a_1, a_2, ..., a_N\}$ is a set of vertices, $|A|=N$ is a number of vertices (size of a problem), $V=\{v_1, v_2, ..., v_M\}\subseteq A\times A$ is a set of arcs, $|V|=M$ is a number of arcs, $v_i=\left(a_{beg}^{(i)}, a_{end}^{(i)}\right)$, $a_{beg}^{(i)}\in A$, $a_{end}^{(i)}\in A$, and the arcs are weighted by the length value $L(v_i)>0$, $i=\overline{1,M}$. Target function in the specified problem is a length of path $L(P)=\sum_{j=1}^{R-1}L\left(a_{i_j}, a_{i_{j+1}}\right)\to\min$; the density of graph $d(G)=\frac{M}{N(N-1)}\in[0;1]$ has a restrictive role, as a large number of decisions are prohibited in the graphs with low density. Selected problem has exact optimal decision (*abbr.* O) that can be obtained polynomially using Dijkstra algorithm [22] for a quadratic time that allows to use it as a test problem for estimating the deviations of the quality of decisions from optimum for heuristic methods.

In several combinatorial optimization problems, it is shown [23, 24] that heuristic methods are characterized by significantly different behavior while solving the problems with different size of a problem and different power of restrictions. For the purpose of careful analysis of this feature, a computer experiment is organized. During this experiment, a sample $\Lambda=\{G_1, G_2, ..., G_K\}$ of $K=1\,000$ graphs with pseudorandom structure is formed with the given number of vertices *N*, density *d* and pseudorandom values of arcs lengths $0<L(v_i)\le 1$. The following average parameters are calculated for the obtained decisions (paths *P* with length $L(P)$): average length of paths (*abbr*. AVG) $\bar{Q}=\frac{\sum_{i=1}^{K}Q(G_i)\phi(G_i)}{K}$, where $Q(G_i)=L(P\in G_i)$ is the quality of the best decision (path with minimal length) for selected method, $\phi(G_i)\in\{0,1\}$ is a function that takes the value of 1 if the decision is found with selected method and 0 otherwise; average deviation of quality of the best decision from optimum (*abbr*. DIFF) $\Delta\bar{Q}=\frac{\sum_{i=1}^{K}\left(Q(G_i)-Q^*(G_i)\right)\phi(G_i)}{K}$, where $Q^*(G_i)$ is a quality of optimal decision (length of the shortest path) given by Dijkstra algorithm; probability of finding a decision (*abbr*. PFD) $\bar{p}=\frac{\sum_{i=1}^{K}\phi(G_i)}{K}$ and the probability of finding an

optimal decision (*abbr*. PFOD) $\overline{p}_{opt} = \frac{\sum_{i=1}^{K} \theta(G_i)}{K}$, where $\theta(G_i) = \begin{cases} 0, & Q(G_i) > Q^*(G_i), \\ 1, & Q(G_i) = Q^*(G_i). \end{cases}$

To compare the methods for selected values of vertices number *N* and density of graph *d* it is required the computational time from several minutes to several hours. To analyze the behavior of methods under different conditions of use, a computing experiment is organized based on distributed computing project Gerasim@Home on BOINC platform [25]. Within this experiment, we studied the behavior of the methods for $2 \le N \le 500$ and $0 < d \le 0{,}2$ with steps $\Delta N = 1$ and $\Delta d = 0{,}001$; obtained results are presented in this article. The formation of $C_{\max} = 1\,000$ decisions are organized for each set of source data for iteration methods with the selection of the best of them.

We give a brief description of the heuristic methods (a detailed description is given in the works [2, 19, 26–28]). As it has already been shown above, for optimal decisions with quality $Q^*(G_i)$, Dijkstra algorithm is used.

3. Brief Classification of Heuristic Methods Based on Modifying Operations

Before describing the details of the method's implementation, it is necessary to introduce the concepts of modifying operations. In most methods as an initial solution some initial path (for example, $P = \left[a_{beg}, a_{end}\right]$ or some path from different heuristic method) was chosen which within the processing of the algorithm is subjected to following modifications (modifying operations): inserting the random vertex $a_j \notin P$ to the random position *R* of the current path, deleting the random vertex $a_j = a_{i_R} \in P$ from the current way with getting a new path, replacing the random vertex a_{i_R} in the current path to the random and unvisited vertex $a_j \notin P$ and rearranging the pair of randomly selected vertexes $a_{i_{R_1}}$ and $a_{i_{R_2}}$ in the current path.

First two modifying operations can be considered elementary, and last two operations can be obtained by successively applying elementary operations. The choice of one of the considered modifying operations for the current path is made randomly in the proportion to the corresponding values of the parameters p_1, p_2, p_3, p_4 (value $\frac{p_i}{\sum_{j=1}^{4} p_j}$ can be perceived as a probability of applying of the *i*-th modifying operation). Choice of the values p_i performed during meta-optimization has a significant influence on the quality of the resulting decisions [2].

The random walks method (*abbr*. RW) based on the random selection of one of the modifying operations o_j for the current decision X_i with getting a new decision $X_{i+1} = o_j(X_i)$ eventually. As a result of the repetition of the above procedure, $C_{\max}$ times (iterations), $C_{\max}$ decisions are formed, for each of which a quality assessment is performed (in the problem under consideration, the path length $L(P)$) and the best decision is returned as the result of the method work.

In the simulated annealing method (*abbr*. SA), more intellectual strategy of applying the modifying operations is used. As a result of applying the modifying operation o_j to the current decision X_i, as shown above for the random walks method, decision X_{i+1} is formed, and the quality assessment $Q(X)$ is performed for both of them aimed to get the change of decision quality $\Delta Q(X_i, X_{i+1}) = Q(X_{i+1}) - Q(X_i)$. Taking into account the fact that the best solution is characterized by a lower quality value $Q(X)$ (path length) in the problem being solved, the value

$\Delta Q<0$ corresponds to the improvement and $\Delta Q>0$ – to the worsening of the quality of the newly obtained decision, respectively. In the case with $\Delta Q<0$, a new decision X_{i+1} replaces the current decision X_i. If the quality of the decision worsens after a random change $(\Delta Q>0)$, the probability is calculated, $p=e^{-\frac{\Delta Q}{T}}$, where *T* is a current temperature, and only in the case with $r_k>p$, where $r_k\in[0;1]$ – next pseudorandom value, a new decision replaces the current one. The temperature is decreased from iteration to iteration as $T^{(i)}=\alpha T^{(i-1)}$, where $\alpha\in[0;1)$ is a tuning parameter that regulates the rate of cooling starting with some initial value $T^{(0)}$ which leads to ever less allowed deterioration in the quality of the current solution ΔQ over time. During the iterative process the quality of all intermediate solutions is evaluated and the best of those is the result of the method work.

Genetic and evolutionary heuristic methods are grouped (*abbr*. GA) based on applying the genetic operators (in most cases, selection, crossover and mutation) with the aim of improving the fitness function of individuals in the population during the emulated natural selection. Each individual X_i, $i=\overline{1,W}$ in the mentioned problem within population $X=\{X_1,X_2,...,X_W\}$ corresponds to the path $P_i=\left[a_{i_1}=a_{beg},a_{i_2},...,a_{i_Q}=a_{end}\right]$ between vertexes a_{beg} and a_{end} of the path [27]. Size of the population $W=|X|$ is selected as some constant value. As the fitness function $f(X_i)$ that determines the probability of participation of the selected individual X_i in crossover with the possibility of producing offspring, the target function (length of the path $L(P)$) is used. A pair of different individuals participates in the crossing with the probability that is proportional to their fitness function.

Considering the necessity of genetic information inheritance from the parents to descendants, we perform the crossing so that the descendant contains a part of the path from each of the parents. In this case, two situations shown schematically in figure 1 can arise in practice.

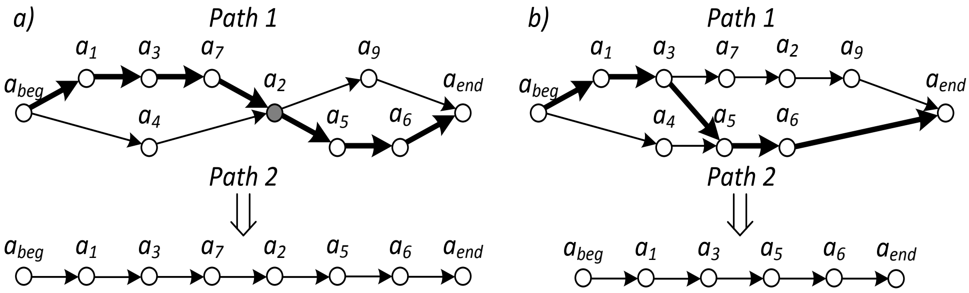

Figure 1. Examples of crossing intersecting (a) and non-intersecting (b) paths, the resulting paths are marked with bold lines

In the first case (figure 1, a), two paths that intersect in the vertex a_2 are selected that allows the trivial implementing of the genetic information copying from the parents to the descendant by choosing the first half of the path (prior to the point of intersection) from one parent and the second (after the intersection point) from the other one. In the given example, when crossing a pair of individuals identified with paths the $P_1=\left[a_{beg},a_1,a_3,a_7,a_2,a_9,a_{end}\right]$ and $P_2=\left[a_{beg},a_4,a_2,a_5,a_6,a_{end}\right]$, it is possible to get the descendants $P_1'=\left[a_{beg},a_1,a_3,a_7,a_2,a_5,a_6,a_{end}\right]$ and $P_2'=\left[a_{beg},a_4,a_2,a_9,a_{end}\right]$ with the subsequent equiprobable choice between them. In the given example path the P_1' is selected. In the second case (figure 1, b), the paths do not intersect and the crossing can be performed by selecting a pair of random vertexes in each of the paths and then connecting them with an arc (in the example the arc is added between vertexes a_3 and a_5). In this case, as in the previous example, the descendant retains a part of the genes (path fragments) from the parents which corresponds to the concept of the base blocks [3] responsible for the transfer of the genetic information from the parents to descendants.

The mutation operator in the problem under consideration can be implemented in the form of the simplest modifying operations discussed above. During the computational experiments, it is found out that only the operation of adding the vertices to the path has a statistically significant effect on the quality of solutions, the influence of the remaining operations is practically not traced.

During the implementation of crossover and mutation operators, solutions with cycles (the same vertex is visited twice) may arise. Obviously, the presence of the cycles in the graphs with the positive arcs weights does not lead to shorter paths; therefore, such solutions must be subjected to the cycle removal procedure. Example, the path $P = \left[a_{beg}, \underbrace{a_9, a_6, a_5, a_2, a_7, a_9}_{cycle}, a_4, a_{end} \right]$ with the cycle must be reduced to the path $P' = \left[a_{beg}, a_9, a_4, a_{end} \right]$ without the cycle (figure 2).

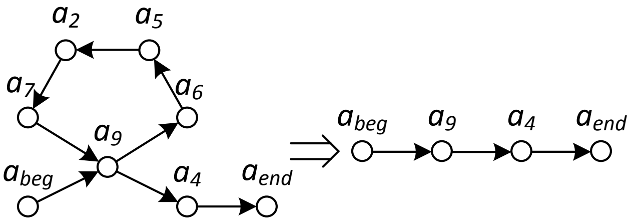

Figure 2. Example of the paths P and P'

Using the above procedure for selecting the fittest individuals followed by the use of crossover and mutation operators with subsequent removing of the cycles, it is possible to implement an iterative evolution process during which the fitness function (decision quality) is evaluated for each individual, and the current record is updated. The final number of analyzed decisions is defined as $C_{\max} = (R+1)W$, the initial population formation is possible with using any heuristic method that provides a variety of solutions (genetic information).

When solving optimization problems using the bee colony method (*abbr*. BC) only two types of bees (agents) are usually used: scouts and foragers (working bees). In nature, scouts are investigating the surrounding hive of space and reporting about promising places where the greatest amount of nectar is found (for information exchange in a hive, there is a special cunning mechanism called a bee dance). Further on the most promising areas indicated by the scouts are the departure of working bees who are engaged in the collection of nectar and, in passing, refine the information of scouts about the amount of nectar in some neighborhood from the area indicated by the scout.

These social roles of the agents can be used to solve the problems of discrete optimization [28]. At the same time, the role of a scout is transformed into an exploration of the space of permissible decisions $\Re$ with selecting one of its points $X_i \in \Re$ where target function has some value *a*. This choice in the simplest case can be implemented using the random search method. Due to the impossibility of storing information about all paths in the electronic analogue of the hive, the information about the best *W* places $S = \{X_1, X_2, ..., X_W\} \subseteq \Re$ is stored. These places are characterized by the best values of the objective function $f(X_1) \le f(X_2) \le ... \le f(X_W)$ during the method's implementation (which is sufficiently similar to the restriction on the population size using the genetic method, but in this case, the value of *W* mostly characterizes the information capacity of the hive "memory", rather than the immediate number of bees). Working bee takes as the direction of motion one of the values $X_i \in S$, $i = \overline{1,V}$, in the course of its work it performs the reconnaissance "around" the starting point X_i. To do this, it is necessary in some way to modify the initial solution X_i for which, it is possible to use the modifying operations considered above, applying each of them leads to a single elementary change in the solution $X_i \to X_i'$, a minimal number of corresponding operations $d\left(X_i, X_j\right)$, necessary for making a decision X_j from X_i, can be regarded as some metric, that is specific for this problem and is an analog of the widely known Hamming or Levenshtein distances [29, 30]. The number R of modifying operations that sequentially is applied to the decision X_i can be treated as the radius of the vicinity of the reconnaissance. The result of the work for the forager bee is the decision X_j and the value of target function $f\left(X_j\right)$ corresponding to it. If at least one decision $X_k \in S$, $k = \overline{1,W}$ among the previously found *W* decisions is inferior in quality to the newly found ones, the previous decision is replaced with a new one with their unchanged total size in the electronic model of the hive: $S^{(t)} = S^{(t-1)} \setminus \left\{X_k\right\} \bigcup \left\{X_j\right\}$, where t – discrete time (number of iteration). Combining the work of scouts and working bees for iterations $C_{\max}$, the quality of solutions in the composition of the selected subset *S* is monotonically increased. The choice of the best among them can be considered the result of the algorithm.

Particle swarm optimization method (*abbr*. PSO) is based on the movement of swarm agents in some space. Each position of the agent is associated with one of the decisions of the problem, and the dimension of the space is determined by the size of the problem (the number of elements within decision). While solving problems of continuous optimization, each of the agents is characterized by the current position (vector of coordinates) X_i and direction (vector of speed) V_i. The coordinates of the agent when changing to a new iteration change as $X_i^{(t+1)} = X_i^{(t)} + V_l^{(t)}$, where *t* – iteration number. Vectors of agent speed usually change in such a way that the agent moves in the vicinity of the local (better position of one agent during the computational experiment) and the global record (the best solution for the entire simulation time).

In solving problems of discrete optimization, the direct organization of the motion of agents in continuous space is impossible in view of the fact that the elements of the solution of the problem are discrete by definition. In this regard, 3 different approaches are used in practice. First of them (*abbr*. PSO1) operates with the swarm motion in a continuous space with the subsequent mapping of the real coordinates of agents into discrete elements of the solution, for example, using the rounding operation. In the problem under consideration, the probabilities p_{ij} of occurrence of the vertex a_i in the *j*-th position of the formed path are chosen as coordinates of the positions of the swarm agents. The second approach (*abbr*. PSO2) is based on rounding the coordinates of the swarm particles to obtain discrete vertex numbers in the path. The third direction of the approaches (*abbr*. PSO3) does not use the notion of speeds and coordinates in a continuous space. Instead of specifying which solutions of the problem are applied taking into account its specificity, the solution *X* is modified in order to obtain the solution X' which is more similar to the required *Y* (in the mentioned method – to global or local record). In this case, instead of the coordinates of the swarm agents, specific solutions encoded in terms of the problem being solved are used, and the speeds correspond to the probabilities p_i of applying one of modifying operation o_i in order to obtain a new position of the decision $X' = o_i\left(X\right)$ in a discrete space.

At the same time, to preserve the general idea of PSO method, the condition $d(X',Y)<d(X,Y)$ must be provided.

4. Computing Experiments and Analysis of the Experimental Data

The problem of analyzing a space with the coordinates $(d;N)$ considered above is weakly coupled since the analysis of its various points can be performed independently of each other. This feature allows the efficient use of grid systems on a voluntary basis on the BOINC platform for its solution which is successfully applied in a number of projects [10, 31, 32]. In this case, within the one workunit (*abbr*. WU), a client machine receives information about initial parameters *N* and *d* for the sample Λ of the graphs with the pseudo-random structure and a method name that is used to get the decisions for each of them checking their correctness and calculating quality assessment, the results of which are returned to the server of the project. As a result of post-processing of the obtained experimental data, the quality criteria of the decisions for the selected point $(d;N)$ of the analyzed space are calculated, and their comparison with optimal decision if performed. Experimental results of this process in a view of the two-dimensional diagrams are presented below.

The computing experiment took 5 months in the grid with real performance about 2 TFLOP/s. Figure 3 shows the experimental results for optimal decisions provided by Dijkstra algorithm.

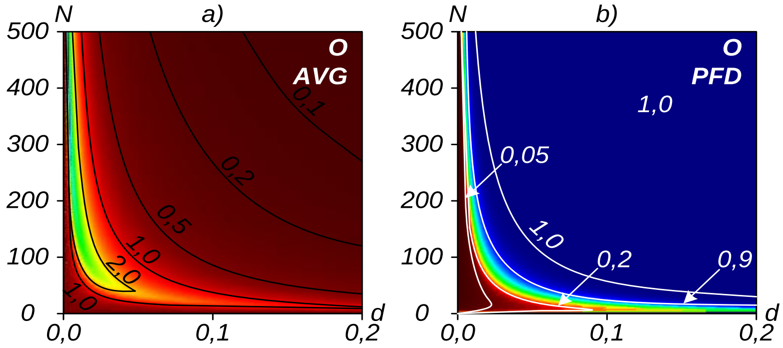

Figure 3. Average path length (a) and probability of finding decision (b) depending on the number of vertexes *N* and graph density *d* (optimal decisions)

First of all, it should be noted that the level lines in figure 3a are very similar to hyperbolas in coordinates $(d;N)$. Minimal length of the paths corresponds to the graphs with high density and big number of vertexes (top right corner in figure 3a). The greatest length of the paths is observed for the graphs with low density with big number of vertexes (top left corner in figure 3a). Dijkstra's algorithm guarantees a decision in case of its existence (if the, graph in which the path is found, is connected), therefore, the dependence in figure 3b can be treated as the probability of getting connected the graph for selected values of *N* and *d*.

Figures 4–10 show the dependences of the averaged deviation of the solution quality from the optimum and the probabilities of finding the solution and finding the optimal solution for the methods considered in the article.

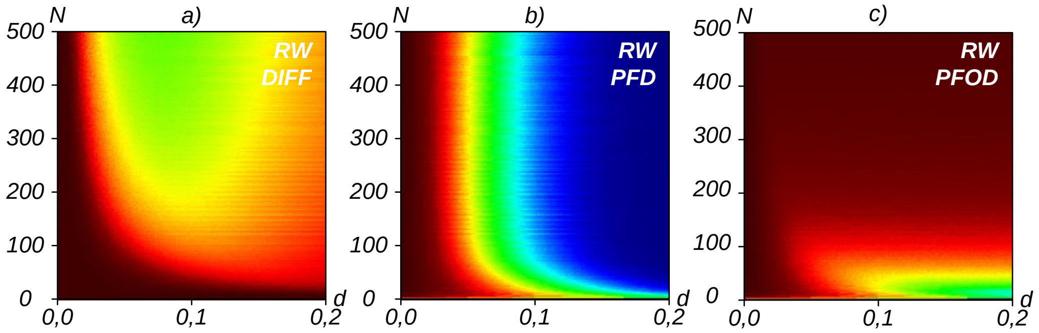

Figure 4. Dependencies of average deviation of quality of the best decision from optimum (a), probability of finding decision (b) and probability of finding optimal decision (c) for random walks method. Red color corresponds to the minimal value, blue color – to the maximal value

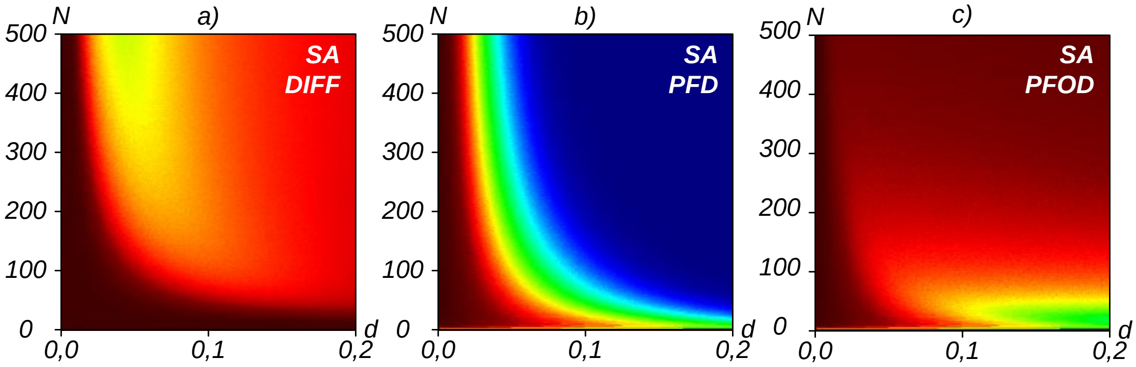

Figure 5. Dependencies of average deviation of quality of the best decision from optimum (a), probability of finding decision (b) and probability of finding optimal decision (c) for simulated annealing method

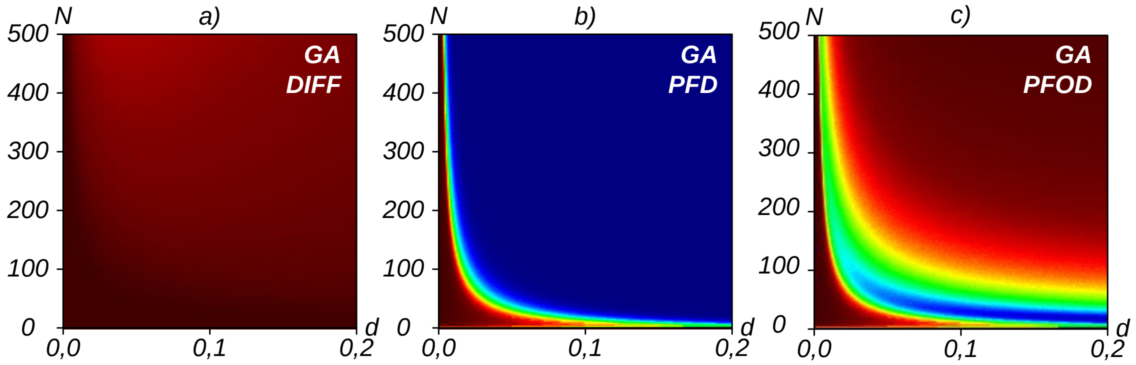

Figure 6. Dependencies of average deviation of quality of the best decision from optimum (a), probability of finding decision (b) and probability of finding optimal decision (c) for genetic algorithm

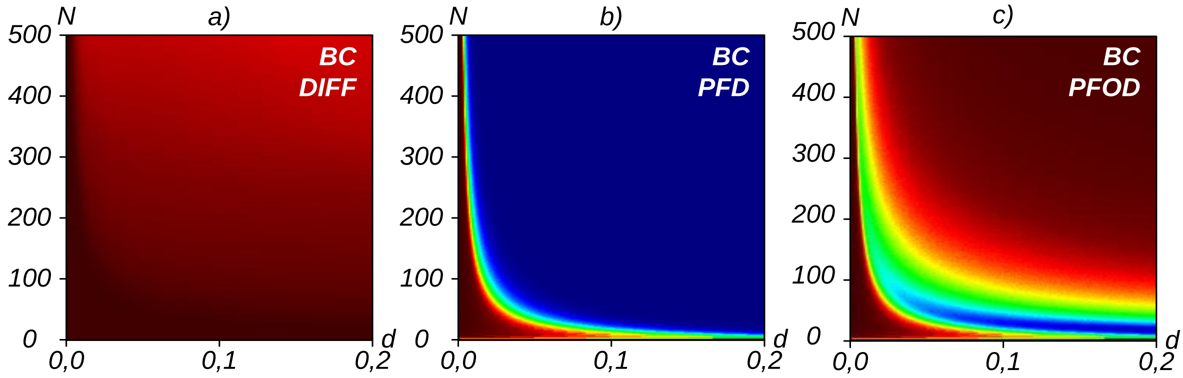

Figure 7. Dependencies of average deviation of quality of the best decision from optimum (a), probability of finding decision (b) and probability of finding optimal decision (c) for bee colony algorithm

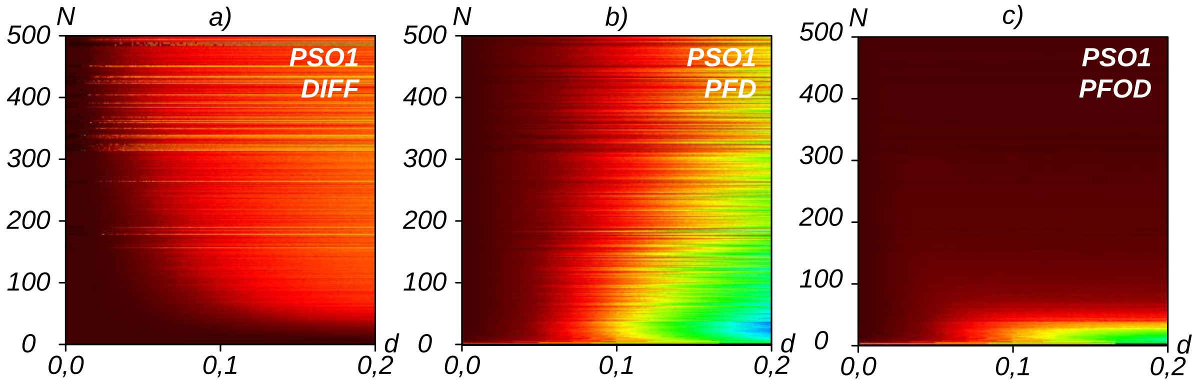

Figure 8. Dependencies of average deviation of quality of the best decision from optimum (a), probability of finding decision (b) and probability of finding optimal decision (c) for particle swarm optimization method, version 1

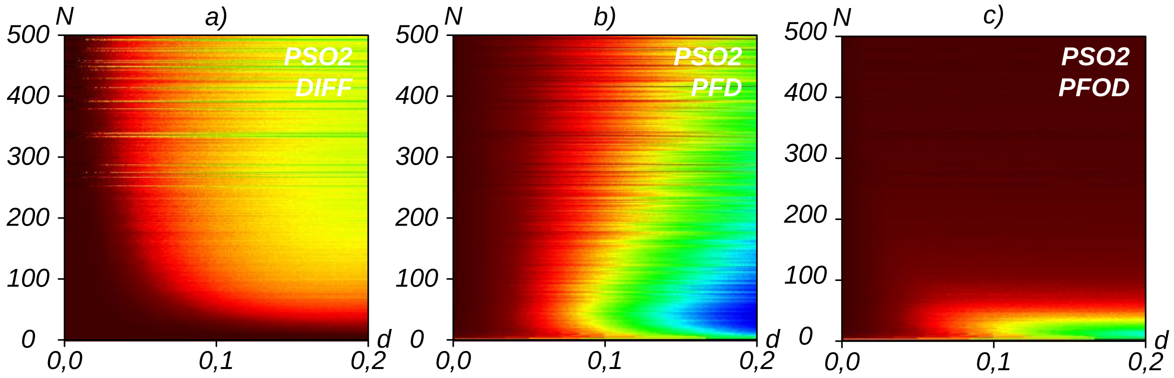

Figure 9. Dependencies of average deviation of quality of the best decision from optimum (a), probability of finding decision (b) and probability of finding optimal decision (c) for particle swarm optimization method, version 2

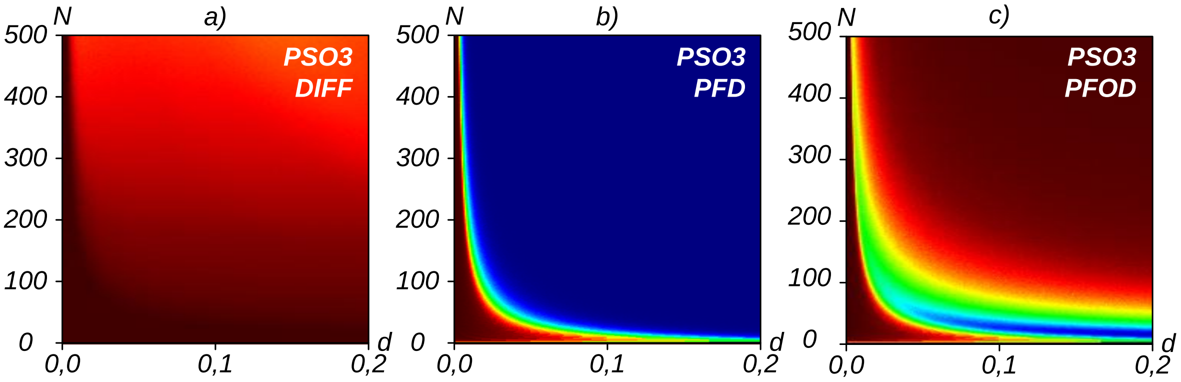

Figure 10. Dependencies of average deviation of quality of the best decision from optimum (a), probability of finding decision (b) and probability of finding optimal decision (c) for particle swarm optimization method, version 3

The analysis of obtained experimental results makes it possible to formulate a number of conclusions. First, the greatest deviation from the quality of the optimal decision is observed for the methods RW, PSO1 and PSO2 which is a consequence of their low efficiency in this problem. Methods RW, SA, PSO1 and PSO2 are characterized by a low probability of finding decisions if the density of graphs *d* decreases. The probability of finding optimal decisions close to 1 is provided by the methods GA, BC and PSO3 for the graphs with a small number of vertexes; if the number of vertexes increases, this probability decreases significantly (in contrast to the method of the colony in [20]).

In order to identify the advantage of the probability of obtaining the optimal decision, we compared the corresponding probabilities for the methods considered above for different values of *N* and *d*, resulting in the areas where one of the two algorithms has the advantage shown in figure 11a.

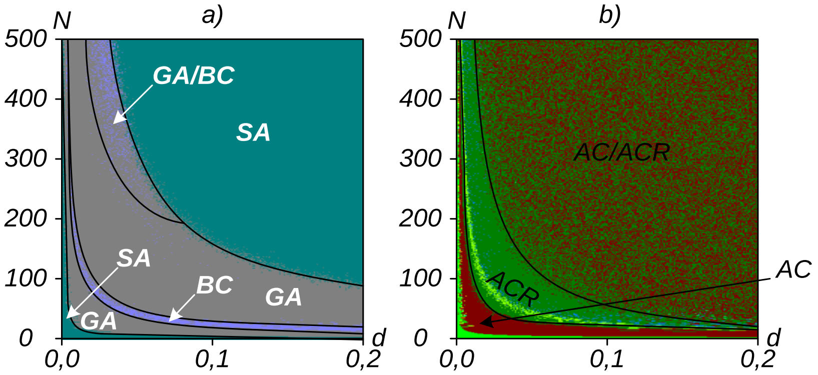

Figure 11. Comparison of the methods based on modifying operations decisions quality by the criterion of the probability of obtaining a better decision (a) and methods AC and ACR from [20] (b)

The analysis of obtained results allows us to conclude that there is a zone dependence of the quality of the decisions obtained from the selected heuristic method in the problem under consideration.

As a result of comparing decisions quality of heuristic methods with the sequential formation of the decision, the article [20] has shown that the best method in this problem is the method of the ant colony optimization (*abbr*. AC) and its modification based on combinatorial returns (abbr. ACR) [33]. Figure 11b presents the results of the comparison of group of methods discussed above with the results described in [20] for the AC and ACR methods by the criterion of the maximum probability of obtaining the optimal decision $\overline{p}_{opt}$.

5. Conclusion

The analysis of the comparison results of group of methods based on modifying operations allows to conclude that the particle swarm optimization, random walks, simulated annealing and bee colony methods are ineffective in the selected problem (excluding dense graphs with low number of vertexes *N*). Among selected group of methods, the most perspective method is the simulated annealing method that provide good quality decisions for dense graphs with different number of vertexes. The comparison of this results with the ant colony and its modification with support for combinatorial returns allow to conclude that this group of methods have a significant advantage in the selected test problem in comparison with the methods considered above.

6. Acknowledgements

The author would like to thank all volunteers who took part in the calculation within the distributed computing project Gerasim@Home. The author also wish to thank Anna Vayzbina for assistance in preparing the English version of the article. The author also wish to thank Sergey Valyaev for developing and supporting the technical part of Gerasim@Home project. The article was partially supported by RFBR (grants 17-07-00317-a, 16-07-00214-a).

References

1. Rosen K.H. et al. Handbook of discrete and combinatorial mathematics. CRC Press, New York, 2000, 768 p.
2. Vatutin E.I., Titov V.S., Emelyanov S.G. Basics of discrete combinatorial optimization (in Russian). M.: Argamac-Media, Moscow, 2016, 270 p.
3. Karpenko A.P. Modern algorithms of search optimization. Algorithms inspired by nature (in Russian). M.: Bauman Moscow State Technical University, 2014, 446 p.
4. Prim R.C. Shortest connection networks and some generalizations // Bell System Technical Journal, 1957, vol. 36, no 6, pp. 1389–1401. DOI: 10.1002/j.1538-7305.1957.tb01515.x
5. Kruskal J.B. On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society, 1956, vol. 7, pp. 48–50. DOI: 10.1090/S0002-9939-1956-0078686-7
6. Kuhn H.W. Variants of the Hungarian method for assignment problems. Naval Research Logistics Quarterly. 1956. pp. 253–258. DOI: 10.1002/nav.3800030404
7. Munkres J. Algorithms for the Assignment and Transportation Problems // Journal of the Society for Industrial and Applied Mathematics, 1957, vol. 1, no. 5, pp. 32–38. DOI: 10.1137/0105003
8. Land A.H., Doig A.G. An Automatic Method of Solving Discrete Programming Problems. Econometrica, 1960, vol. 28, pp. 497–520. DOI: 10.2307/1910129
9. Colbourn C.J., Dinitz J.H. Handbook of Combinatorial Designs, 2nd ed. (Discrete Mathematics and Its Applications). New York: Chapman & Hall/CRC, 2006, 1026 p.
10. Zaikin O.S., Kochemazov S.E., Semenov A.A. SAT-based search for systems of diagonal Latin squares in volunteer computing project SAT@home / 39th international convention on information and communication technology, electronics and microelectronics (MIPRO 2016). 2016, pp. 277–281. DOI: 10.1109/MIPRO.2016.7522152
11. Vatutin E.I., Dremov E.N., Martynov I.A., Titov V.S. Weighted random search method for discrete combinatorial problems solving (in Russian) // Proceedings of Volgograd State Technical University. Series: Electronics, Measuring Equipment, Radiotechnics and Communication, 2014, № 10 (137). 9, pp. 59–64.
12. Dorigo M. Optimization, Learning and Natural Algorithms. PhD thesis. Milan: Politecnico di Milano, 1992.
13. Vatutin E.I., Titov V.S. Analysis of results of ant colony optimization method in the problem of getting shortest path in graph with constraints (in Russian) // Proceedings of South State

University. Technical sciences. 2014, № 12 (161), pp. 111–120.

14. Glover F., Kochenberger G.A. Handbook of Metaheuristics. New York: Kluwer Academic Publishers, 2003, 756 p.
15. Kirkpatrick S., Gelatt C.D., Vecchi M.P. Optimization by Simulated Annealing. Science, 1983, vol. 220, no. 4598, pp. 671–680. DOI: 10.1126/science.220.4598.671
16. Karaboga D.D. An Idea Based On Honey Bee Swarm for Numerical Optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
17. Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., Zaidi M. The Bees Algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, 2005.
18. Kennedy J., Eberhart R. Particle Swarm Optimization / Proceedings of IEEE International Conference on Neural Networks. 1995, pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968
19. Vatutin E.I., Titov V.S. Investigation of Features of Particle Swarm Optimization Method in Graph Shortest Path Problem with Constraints (in Russian) // Herald of computer and information technologies, 2018, no. 5 (167), pp. 26–34. DOI: 0.14489/vkit.2018.05.pp.026–034
20. Vatutin E.I. Comparison of Decisions Quality of Heuristic Methods with Sequential Formation of the Decision in the Graph Shortest Path Problem / CEUR Workshop Proceedings. Proceedings of the Third International Conference BOINC-based High Performance Computing: Fundamental Research and Development (BOINC: FAST 2017). Technical University of Aachen, Germany, 2017, pp. 67–76.
21. Vatutin E.I. Comparison of Decisions Quality of Heuristic Methods with Limited Depth-First Search Techniques in the Graph Shortest Path Problem // Open Engineering, 2017, vol. 7, no 1, pp. 428–434. DOI: 10.1515/eng-2017-0041
22. Dijkstra E.W. A note on two problems in connexion with graphs. Numerische Mathematik, 1959, vol. 1., pp. 269–271.
23. Vatutin E.I., Valyaev S.Yu., Titov V.S. Comparison of Sequential Methods for Getting Separations of Parallel Logic Control Algorithms Using Volunteer Computing / CEUR Workshop Proceedings. Proceedings of the Second International Conference BOINC-based High Performance Computing: Fundamental Research and Development (BOINC: FAST 2015). 2015, vol. 1502, pp. 37–51. urn:nbn:de:0074-1502-3
24. Vatutin E.I., Titov V.S. Analisys of the areas of qualitative superiority of the sequential heuristic methods for getting separation during logic multicontrollers design (in Russian). Proceedings of the Higher Educational Institutions. Instrument Making, 2015, vol. 58, no 2, pp. 115–122. DOI: 10.17586/0021-3454-2015-58-2-115-122
25. Anderson D.P. BOINC: a system for public-resource computing and storage / Fifth IEEE/ACM International Workshop on Grid Computing, 2004. DOI: 10.1109/GRID.2004.14
26. Vatutin E.I., Titov V.S. Parametric optimization of simulated annealing algorithm at the shortest path problem (in Russian) // Proceedings of Cherepovets State University, 2015, no 6 (67), pp.13–16.
27. Vatutin E.I., Titov V.S. The features of use genetic algorithm in the problem of finding the shortest path in the graph with graph density constraint (in Russian) // Multicore processors, parallel programming, FPGA, signal processing systems. Barnaul: Altay State University, 2016. pp.152–159.
28. Vatutin E.I., Titov V.S. Features of meta-optimization of the bee colony method in the shortest path problem with constraints on the graph density (in Russian) // Proceedings of Southwest State University. Series: Control, Computer Science, Informatics. Medical Devices, 2016, no 2 (19), pp. 52–65.
29. Hamming R.W. Error detecting and error correcting codes // Bell System Technical Journal. 1950, vol. 29, pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.
30. Levenstein V.I. Binary codes with correction of fallouts, inserts and substitutions of symbols

(in Russian) // Proceedings of Academies of Sciences of the USSR, 1965, vol. 163, no 4. pp. 845–848.

31. Kurochkin I.I. Determination of Replication Parameters in the Project of the Voluntary Distributed Computing Netmax@Home // Science. Business. Society, 2016, vol. 2, pp.10–12.
32. Nikitina N.N., Ivashko E.E. The Price of Anarchy in a Game for Drug Discovery. Networking Games and Management Extended abstracts. 2016, pp. 24–25.
33. Vatutin E.I., Martynov I.A., Titov V.S. Method of Workaround deadlocks for solving discrete combinatorial optimization problems with constraints (in Russian) // Perspective information technologies. Samara, Samara scientific center of RAS, 2014, pp. 313–317.

UOT 004.023

Vatutin E.İ.^{1}, Panişev V.S.^{2}, Qvozdeva S.N.^{3}, Titov V.S.^{4}
Cənub-Qərb Dövlət Universiteti, Kursk, Rusiya
[1]evatutin@rambler.ru, [2]gskunk@yandex.ru, [3]svetka-gvozdeva@yandex.ru, [4]titov-kstu@rambler.ru

Qrafda ən qısa yolun axtarışı məsələsində dəyişən əməliyyatlar əsasında evristik metodların həlli keyfiyyətinin müqayisəsi

Məqalə qrafda ən qısa yolun axtarışı məsələsində mövcud həllərin modifikasiyası əsasında evristik iterasiya metodlarının effektivliyinin öyrənilməsinə həsr edilmişdir. Məqalədə baxılan metod qrupunun qısa təsviri verilmişdir. Təşkil olunmuş hesablama sınağı verilmiş parametrlərlə psevdotəsadüfi qrafların seçiminin formalaşdırılmasına və BOINC platformasında könüllü paylanmış hesablamalardan istifadə etməklə həllərin axtarışının təşkilinə əsaslanmışdır. Məqalədə məsələnin ölçüsündən və məhdudiyyət gücündən asılı olaraq seçilmiş metod qrupundan alınmış eksperimental nəticələrin ətraflı təsviri verilmişdir. Alınmış eksperimental nəticələr belə bir nəticəyə gəlməyə imkan verir ki, hissəcik sürüsü metodları, təsadüfi dolaşma, yanma təqlidi və arı sürüsü metodları seçilmiş məsələnin həllində səmərəli deyil və həllərin keyfiyyəti üzrə qarışqa sürüsü və genetik alqoritm metodlarından əhəmiyyətli dərəcədə geridə qalır.

Açar sözlər: *evristik metodlar, genetik alqoritmlər, hissəcik sürüsü metodu, təsadüfi dolaşma metodu, yanma təqlidi metodu, arı sürüsü metodu, qrafda ən qısa yolun axtarışı, diskret kombinator optimallaşdırma, BOINC.*

УДК 004.023

Ватутин Э.И.^{1}, Панищев В.С.^{2}, Гвоздева С.Н.^{3}, Титов В.С.^{4}
[1]Юго-Западный государственный университет, Курск, Россия
[1]evatutin@rambler.ru, [2]gskunk@yandex.ru, [3]svetka-gvozdeva@yandex.ru, [4]titov-kstu@rambler.ru

Сравнение качества решений эвристических методов на базе модифицирующих операций в задаче поиска кратчайшего пути в графе

Статья посвящена изучению эффективности эвристических итерационных методов на базе модификации существующих решений в задаче поиска кратчайшего пути в графе. В статье приведено краткое описание рассматриваемой группы методов. Организованный вычислительный эксперимент базируется на формировании выборок псевдослучайных графов с заданными параметрами и организации поиска решений с использованием добровольных распределенных вычислений на платформе BOINC. В статье приведено подробное описание полученных экспериментальных результатов выбранной группы методов в зависимости от размерности задачи и силы ограничений. Полученные экспериментальные результаты позволяют сделать вывод, что методы роя частиц, случайных блужданий, имитации отжига и пчелиной колонии неэффективны при решении выбранной задачи и существенно уступают по качеству решений методам муравьиной колонии и генетическим алгоритмам.

Ключевые слова: *эвристические методы, генетические алгоритмы, метод роя частиц, метод случайных блужданий, метод имитации отжига, метод пчелиной колонии, поиск кратчайшего пути в графе, дискретная комбинаторная оптимизация, BOINC.*