

## Research Article

Eduard Vatutin\*

# Comparison of Decisions Quality of Heuristic Methods with Limited Depth-First Search Techniques in the Graph Shortest Path Problem

<https://doi.org/10.1515/eng-2017-0041>

Received Oct 02, 2017; accepted Nov 13, 2017

**Abstract:** The article deals with the problem of analysis of effectiveness of the heuristic methods with limited depth-first search techniques of decision obtaining in the test problem of getting the shortest path in graph. The article briefly describes the group of methods based on the limit of branches number of the combinatorial search tree and limit of analyzed subtree depth used to solve the problem. The methodology of comparing experimental data for the estimation of the quality of solutions based on the performing of computational experiments with samples of graphs with pseudo-random structure and selected vertices and arcs number using the BOINC platform is considered. It also shows description of obtained experimental results which allow to identify the areas of the preferable usage of selected subset of heuristic methods depending on the size of the problem and power of constraints. It is shown that the considered pair of methods is ineffective in the selected problem and significantly inferior to the quality of solutions that are provided by ant colony optimization method and its modification with combinatorial returns.

**Keywords:** heuristic methods, Brute Force method, ant colony optimization, shortest path problem, discrete combinatorial optimization, BOINC

## 1 Introduction

There is a wide class of optimization problems with restrictions on discrete values of decision elements (arguments of the objective function) [1, 2]. These include many problems from combinatorics, graph theory, operations research. Mathematical apparatus of derivatives and gradi-

ents is not applicable for them, but it is successfully used in solving 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 have similar methods. To solve them, in practice, heuristic methods need to be used. They provide decisions of good quality for reasonable computing time.

Known 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 with 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 with using some rules that are specific to the problem being solved (discrete versions of random walks method and simulated annealing method, evolutionary (genetic) methods group, 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]).

Brute Force method and its modifications have 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 mentioned above, it is possible to construct hybrid multistep methods [3].

\*Corresponding Author: Eduard Vatutin: Southwest State University, Kursk, Russia; Email: [evatutin@rambler.ru](mailto:evatutin@rambler.ru)

## 2 Statement of the Problem

Different heuristic methods are characterized in practice both by the different costs of computer time, and by the different quality of the decisions obtained, in this view it is interesting to compare quality of decisions aimed to select the best of methods. For this purpose a test problem of getting shortest path  $P(G) = [a_{i_1} = a_{beg}, a_{i_2}, \dots, a_{i_Q} = a_{end}]$  in graph  $G = \langle A, V \rangle$  was selected, where  $A = \{a_1, a_2, \dots, a_N\}$  is a set of vertices,  $|A| = N$  is a number of vertices (size of the problem),  $V = \{v_1, v_2, \dots, v_M\} \subseteq A \times A$  is a set of arcs,  $|V| = M$  is a number of arcs,  $v_i = (a_{beg}^{(i)}, a_{end}^{(i)})$ ,  $a_{beg}^{(i)} \in A$ ,  $a_{end}^{(i)} \in A$ , and the arcs are weighted by the length value  $L(v_i) > 0$ ,  $i = \overline{1, M}$ . Objective function in the specified problem is a length of path  $L(P) = \sum_{j=1}^{R-1} L(a_{i_j}, a_{i_{j+1}}) \rightarrow \min$ , density of graph  $d(G) = \frac{M}{N(N-1)} \in [0.0; 1.0]$  has a role of restriction as in graphs with low density a large number of decisions are prohibited. This problem has exact optimal decision (*abbr.* O) that can be obtained polynomially using the Dijkstra algorithm [19] in quadratic time, that allows to use it as a test problem for estimating deviations of decisions quality from optimum for heuristic methods. In a number of combinatorial optimization problems it was shown [20, 21] that heuristic methods are characterized by significantly different behavior during solving problems with different size of a problem and different power of restrictions. For the purpose of careful analysis of this feature a computer experiment was organized. During this experiment it was formed a sample  $\Lambda = \{G_1, G_2, \dots, G_K\}$  of  $K = 1000$  graphs with pseudorandom structure, given number of vertices  $N$ , density  $d$  and pseudo-random values of arcs lengths  $0 < L(v_i) \leq 1$ . For the obtained decisions (paths  $P$  with length  $L(P)$ ) the following average parameters were calculated: average length of paths (*abbr.* AVG)

$$\overline{Q} = \frac{\sum_{i=1}^K Q(G_i) \phi(G_i)}{K}, \quad (1)$$

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 function that has value 1 if the decision was found with selected method and 0 otherwise; average deviation of quality of the best decision from optimum (*abbr.* DIFF)

$$\Delta \overline{Q} = \frac{\sum_{i=1}^K (Q(G_i) - Q^*(G_i)) \phi(G_i)}{K}, \quad (2)$$

where  $Q^*(G_i)$  is a quality of optimal decision (length of the shortest path) given by Dijkstra algorithm; probability of

finding decision (*abbr.* PFD)

$$\overline{p} = \frac{\sum_{i=1}^K \phi(G_i)}{K} \quad (3)$$

and probability of finding optimal decision (*abbr.* PFOD)

$$\overline{p}_{opt} = \frac{\sum_{i=1}^K \theta(G_i)}{K} \quad (4)$$

where

$$\theta(G_i) = \begin{cases} 0, & Q(G_i) > Q^*(G_i), \\ 1, & Q(G_i) = Q^*(G_i). \end{cases} \quad (5)$$

To compare methods for selected values of vertices number  $N$  and density  $d$  of a graph it is required from several minutes to several hours of computational time. To analyze the behavior of methods under different conditions of use there was organized a computing experiment based on the distributed computing project Gerasim@Home<sup>1</sup> on BOINC platform [22]. Within this experiment we studied the behavior of methods for  $2 \leq N \leq 500$  and  $0 < d \leq 0.2$  with steps  $\Delta N = 1$  and  $\Delta d = 0.001$ , the obtained results are presented in Section 4. For each set of source data for iteration methods there was organized a formation of  $C_{max} = 1000$  decisions with selection the best of them.

## 3 Brief Classification of Heuristic Methods with Limited Depth-First Search Techniques

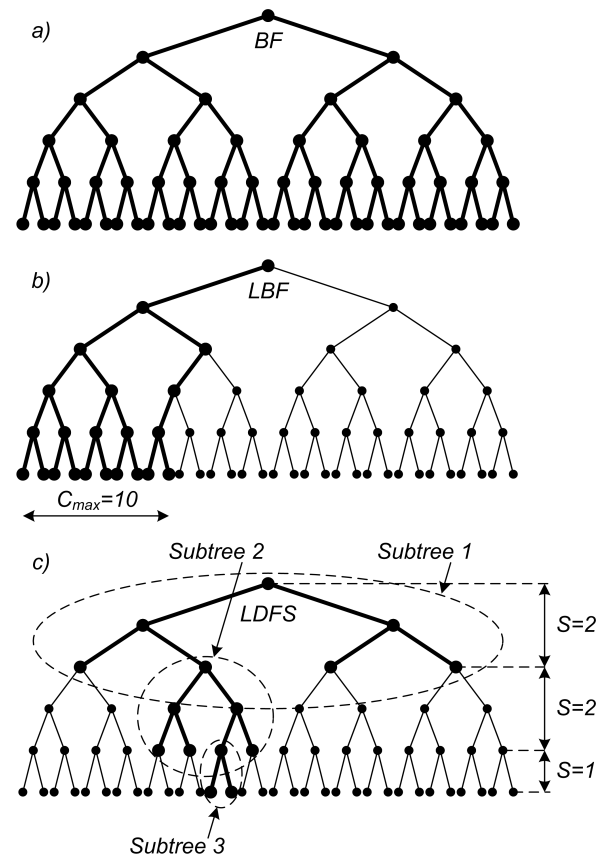
In this section we give a brief description of the heuristic methods (a detailed description is given in the works [2, 23]). As it has already been shown above, for optimal decisions with quality  $Q^*(G_i)$  the Dijkstra algorithm was used. With using Brute Force method (*abbr.* BF) traverse of all branches of combinatorial search tree was organized. In most cases, as a rule, Depth-First Search (*abbr.* DFS) order of combinatorial search tree nodes traverse is used, although in some cases Breadth-First Search (*abbr.* BFS) order [24] is used (see Figure 1a). As a result of the combinatorial search tree traversing each of its branches corresponds to one of the problem decisions. When using the strategy of branches and boundaries, early exclusion from consideration of some subtrees is allowed if the current quality of the formed decision at this stage is worse

<sup>1</sup> <http://gerasim.boinc.ru>

than the record found earlier and the further steps in the depth of the recursion will only degrade it further. By definition the Brute Force method (may be both with branches and boundaries strategy) ensures the obtaining of optimal decisions to the problem. Asymptotic time complexity of the corresponding algorithms and the computational costs of the corresponding software implementations are, as a rule, unacceptably high, which limits the scope of application of these methods in practice to cases of small size of the problem ( $N \leq 30$ ). One way to reduce the complexity of Brute Force is to introduce a limit on the maximum number  $C_{max}$  of branches of the analyzed combinatorial search tree (see Figure 1b), as a result the decision search time increases linearly with the selected value  $C_{max}$  (we will call this approach as Limited Brute Force, *abbr.* LBF). In this case, the decisions are usually close to each other (for example, by analog of the Hamming distance [25]), the search is sufficiently local and does not provide a significant diversification of solutions. Another way to limit the Brute Force method is to enter the value  $S$  of maximum depth, which is used to analyze the subtree within the combinatorial search tree with the given root vertex (we will call this approach Limited Depth-First Search, *abbr.* LDFS) [2, 23]. In this case, among the branches of the subtree of height  $S$  the best of them is selected that provides in this step suboptimal decision with the best quality criterion value, after which the process of constructing and analyzing subtrees iteratively continues until the maximum possible depth is reached (see Figure 1c). Time complexity of the corresponding algorithm is  $O(N^{S+1})$  [23] and can be changed in a wide range depending on the available time resources.

## 4 Computing Experiments and Analysis of the Experimental Results

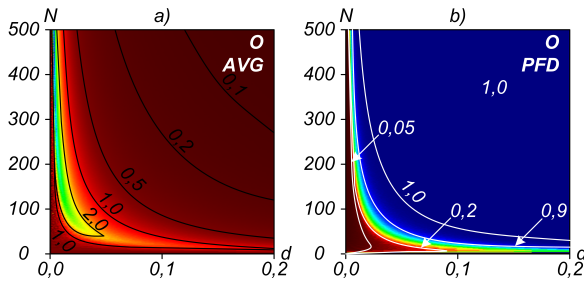
The problem of analyzing a space with coordinates  $(d; N)$  considered above is weakly coupled due to the fact that analysis of its various points can be performed independently. This feature allows the efficient use of volunteer-computing-based grid systems within the BOINC platform for its solution, which is successfully applied in a number of projects [10, 26, 27]. In this case, within one work unit a client machine receives information about the initial parameters  $N$  and  $d$  of the sample  $\Lambda$  of graphs with a pseudo-random structure and the name of the heuristic method, which is used for getting decisions, correctness



**Figure 1:** A schematic illustration of a combinatorial search tree traverse: a – within Brute Force method, b – within Brute Force method with restricted number of analyzed branches, c – within Brute Force with restricted depth of analyzed subtrees. Analyzed branches are bolded.

verifying and quality estimating. Standard Pascal/Delphi pseudo-random generator (linear congruential generator) was used in these experiments. It has period  $2^{32}$  that is suitable for the experiments with samples of pseudo-random graphs with length about 1000 - 10000 test examples. Resulting quality values are transmitted to the server of the project. During post processing of the given experimental results calculations aimed to getting values of criteria (1)-(4) for one selected point  $(d; N)$  of analyzed space, their comparison (if necessary) with optimal decisions and the construction of two-dimensional diagrams presented below are performed.

The computing unit was developed using Delphi programming language, standard BOINC wrapper was used to integrate it with the BOINC Manager software due to the fact that using BOINC sources it isn't possible to statically link a computing unit computing unit into single executable file within Delphi. The computing experiment took 4 months in the grid with real performance about 2



**Figure 2:** Average path length (a) and probability of finding decision (b) depending on the number of vertices  $N$  and graph density  $d$  (optimal decisions).

TFLOP/s. Figure 2 shows the experimental results for optimal decisions provided by the Dijkstra algorithm.

First of all it should be noted that the level lines in Figure 2a are very similar to hyperbolas in coordinates  $(d; N)$ . Minimal length of paths corresponds to the graphs with high density and big number of vertices (top right corner in Figure 2a). The greatest length of the paths is observed for graphs with low density with big number of vertices (top left corner in Figure 2a). 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 2b can be treated as probability of getting connected graph for selected values of  $N$  and  $d$ . During decreasing of graph density  $d$  for each  $N$  such value  $d'$  can be chosen that probability of finding decision is  $\bar{p} \approx 1$ , but  $\bar{p} < 1$ , and also value  $d''$ , that  $\bar{p} \approx 0$ , but  $\bar{p} > 0$ . For example, for  $N = 100$  these values are  $d' \approx 0.063$  and  $d'' \approx 0.002$ . If in the selected range of densities  $d' \leq d \leq d''$  one of methods  $X$  provides the probability of obtaining a decision lower than the Dijkstra algorithm ( $\bar{p}_X < \bar{p}_D$ ), then this indicates that for some test cases decisions exist but have not been found.

Figures 3 and 4 show dependences of average deviation of decision quality and probabilities of finding decision and finding optimal decision for LBF and LDFS methods.

Analysis of the experimental results obtained makes it possible to formulate a number of conclusions. First of all, the greatest deviation of the average path length from the optimum is observed for low-density graphs with a large number of vertices (top left corner of two dimensional diagrams). The LBF method demonstrates a much greater deterioration (more than an order of magnitude) in the quality of decisions compared with the LDFS method. The likelihood of finding solutions is comparable for the pair of methods under consideration: for an absolute majority of cases methods provide solutions that are their advantage. For high-density graphs with a small number of vertices

(bottom right corner of the diagrams), the LBF and LDFS methods demonstrate a high probability of obtaining optimal solutions  $\bar{p}_{opt} = 0.7-0.9$ , during increasing the size of the problem this probability substantially decreases, this process is most rapidly observed for LBF method.

In order to identify the advantage of the probability of finding the optimal decision, a comparison of the values  $\bar{p}_{LBF}$  and  $\bar{p}_{LDFS}$  for different values of  $N$  and  $d$  was organized, as a result, regions were obtained where one of two methods have the advantage (see Figure 5).

Analysis of obtained results shows that for high-density graphs  $d > d'$  LDFS method has the advantage, and during decreasing the density of graph (increasing power of the restriction) LBF method gets the advantage. In the area of small-density graphs ( $d < d''$ ) LDFS method has a small advantage. The boundary between the areas of primary use for the considered pair of methods can be empirically approximated by a hyperbola  $N = \frac{A}{d}$ , where  $A$  – some constant, from which it is possible to obtain the following relation:

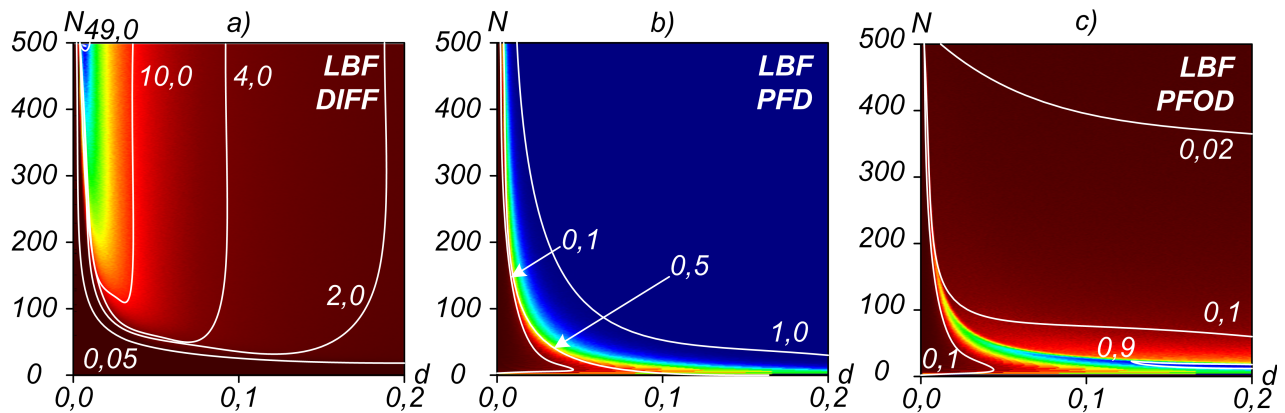
$$N = A \frac{N(N-1)}{M} \Rightarrow 1 = A \frac{N-1}{M} \Rightarrow M = A(N-1). \quad (6)$$

For the pair of methods under consideration  $A \approx 2$ . Using this relation, we can conclude that when  $M > A(N-1)$  the LDFS method has the advantage, and when  $M < A(N-1)$  – LBF method is preferable.

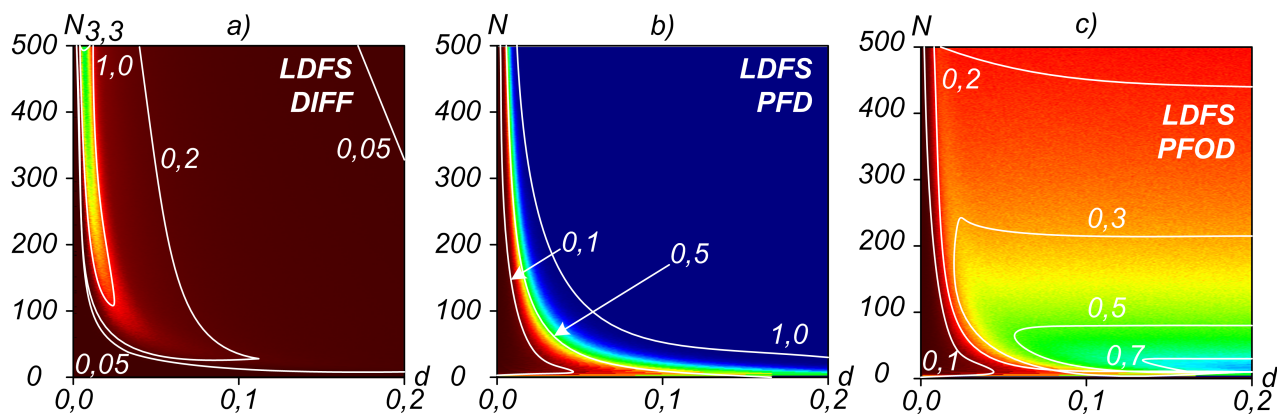
In the article [28] during comparing heuristic methods with sequential formation of the decisions it was shown that in the selected problem the best methods are ant colony optimization method (*abbr.* AC) and its modification with combinatorial returns (ant colony with returns, *abbr.* ACR) [13, 29]. Figure 6 shows results of comparison of described above LDFS and LBF methods with described in [28] results for AC and ACR methods by criterion of maximum probability of finding the optimal decision  $\bar{p}_{opt}$ .

An analysis of comparison results for the group of methods under consideration allows us to conclude that ant colony optimization method has a significant advantage in comparison with the methods of limited Brute Force. During decreasing density of graphs starting from  $d < d' \approx \frac{4.5}{N}$  modified ant colony optimization method with combinatorial returns turns out to be more effective than classical method of ants colony and also surpasses the quality of decision for LBF and LDFS methods. Taking into account the results of the computational experiments, it can be concluded that in the problem under consideration the methods based on the limited Brute Force are ineffective and inferior to the group of methods based on the ant colony heuristic.

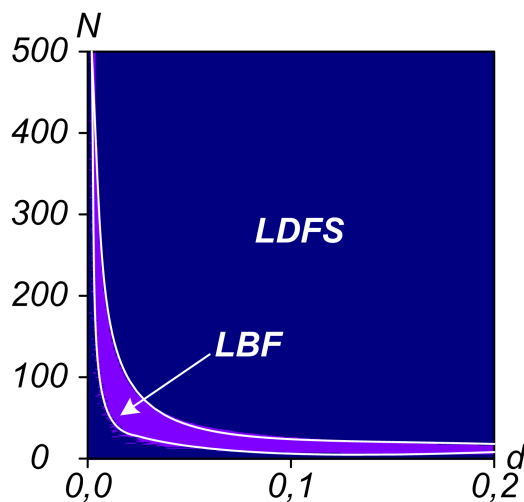




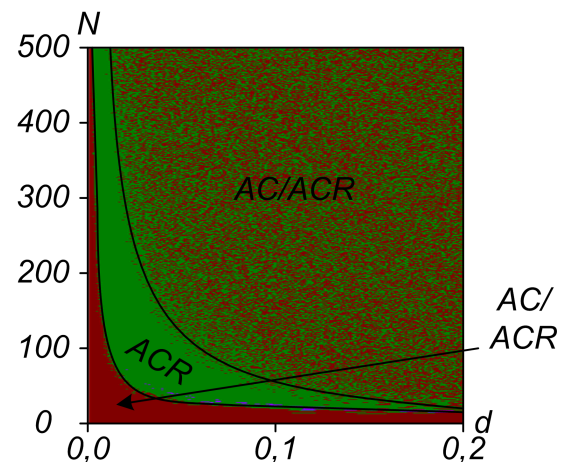
**Figure 3:** Dependencies of average deviation of the best decision quality from optimum (a), probability of finding decision (b) and probability of finding optimal decision (c) for depth-first search with limited number of analyzed branches.



**Figure 4:** Dependencies of average deviation of the best decision quality from optimum (a), probability of finding decision (b) and probability of finding optimal decision (c) for depth-first search with limited height of subtrees within combinatorial search tree.



**Figure 5:** Comparison of methods LBF and LDFS decisions quality by the criterion of the probability of obtaining a better decision.



**Figure 6:** Comparison of decisions quality for described above LBF and LDFS methods with AC and ACR methods from [28] by criterion of maximum probability of finding the optimal decision.

## 5 Prospects for the Future Investigations

The current series of experiments performed within Gerasim@Home project has several objectives and aimed to subset of heuristic methods with limited depth-first search techniques and sequential formation of the decision only. In addition, it would be interesting to study the quality of solutions obtained using different iterative methods such as group intelligence approaches [16, 17], modifications of simulated annealing [15], random walks, particle swarm optimization [18] and so on. Also all of heuristic methods requires testing in different combinatorial optimization problems. Fine tuning of variable parameters of methods during meta-optimization or adaptation is another one important problem that needs to be investigated.

**Acknowledgement:** 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 technical part of Gerasim@Home project. The article was partially supported by RFBR (grant 17-07-00317-a) and by council on grants of the President of the Russian Federation (grant MK-9445.2016.8).

## References

- [1] K. H. Rosen. Handbook of discrete and combinatorial mathematics. Chapman and Hall/CRC, 2000.
- [2] E. I. Vatutin, V. S. Titov, and S. G. Emelyanov. Basics of discrete combinatorial optimization (in Russian). Argamac-Media, 2016.
- [3] A. P. Karpenko. Modern algorithms of search optimization. Algorithms inspired by nature (in Russian). Bauman Moscow State Technical University, 2014.
- [4] R. C. Prim. Shortest connection networks and some generalizations. Bell System Technical Journal, 36(6):1389–1401, 1957.
- [5] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society, 7:48–50, 1956.
- [6] H. W. Kuhn. Variants of the hungarian method for assignment problems. Naval Research Logistics Quarterly, 3:253–258, 1956.
- [7] J. Munkres. Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics, 5(1):32–38, 1957.
- [8] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. Econometrica, 28:497–520, 1960.
- [9] C. J. Colbourn and J. H. Dinitz. Handbook of combinatorial designs, second edition (Discrete mathematics and its applications). Chapman & Hall/CRC, 2006.
- [10] O. S. Zaikin, S. E. Kochemazov, and A. A. Semenov. SAT-based search for systems of diagonal Latin squares in volunteer computing project SAT@home. In 39th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2016, Opatija, Croatia, May 30 – June 3, 2016, pages 277–281, 2016.
- [11] E. I. Vatutin, E. N. Dremov, I. A. Martynov, and V. S. Titov. Weighted random search method for discrete combinatorial problems solving (in russian). Proceedings of Volgograd State Technical University. Series: Electronics, Measuring Equipment, Radiotechnics and Communication, 10(137):59–64, 2014.
- [12] M. Dorigo and T. Stützle. Ant colony optimization. MIT Press, 2004.
- [13] E. I. Vatutin and V. S. Titov. 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, 12(161):111–120, 2014.
- [14] F. W. Glover and G. A. Kochenberger. Handbook of Metaheuristics. Springer US, 2003.
- [15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. Science, 220:671–680, 1983.
- [16] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, October 2005.
- [17] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi. The bees algorithm, a novel tool for complex optimisation problems. In Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2006), pages 454–459. Elsevier, 2006.
- [18] J. Kennedy and R. Eberhart. Particle Swarm Optimization. In Proceedings of IEEE International Conference on Neural Networks, volume 4, pages 1942–1948, Washington, DC, USA, November 1995. IEEE Computer Society.
- [19] E. W. Dijkstra. A note on two problems in connexion with graphs. Numer. Math., 1(1):269–271, December 1959.
- [20] E. I. Vatutin, S. Yu. Valyaev, and V. S. Titov. Comparison of sequential methods for getting separations of parallel logic control algorithms using volunteer computing. In Second International Conference BOINC-based High Performance Computing: Fundamental Research and Development (BOINC:FAST 2015), Petrozavodsk, Russia, September 14-18, 2015, volume 1502 of CEUR-WS, pages 37–51, 2015.
- [21] E. I. Vatutin and V. S. Titov. Superiority areas analysis of sequential heuristic methods for getting separations during logical multicontrollers design. Journal of Instrument Engineering, 58(2):115–122, 2015.
- [22] D. P. Anderson. Boinc: A system for public-resource computing and storage. In Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, GRID '04, pages 4–10, Washington, DC, USA, 2004. IEEE Computer Society.
- [23] E. I. Vatutin, I. A. Martynov, and V. S. Titov. Analysis of results of depth first search with fixed depth method at the shortest path problem (in russian). In Multicore processors, parallel programming, FPGA, signal processing systems, pages 120–128. Altay State University, 2015.
- [24] C. Y. Lee. An algorithm for path connections and its applications. IRE Trans. Electronic Computers, 10(3):346–365, 1961.

- [25] R. W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 26(2):147–160, 1950.
- [26] I. I. Kurochkin. Determination of replication parameters in the project of the voluntary distributed computing netmax@home. *Science. Business. Society*, 2:10–12, 2016.
- [27] N. N. Nikitina and E. E. Ivashko. The price of anarchy in a game for drug discovery. In *Networking Games and Management Extended abstracts*, pages 24–25, 2016.
- [28] E. I. Vatutin. Comparison of decisions quality of heuristic methods with sequential formation of the decision in the graph shortest path problem. In *Third International Conference BOINC-based High Performance Computing: Fundamental Research and Development (BOINC:FAST 2017)*, Petrozavodsk, Russia, August 28 – September 1, 2017, CEUR-WS.
- [29] E. I. Vatutin, I. A. Martynov, and V. S. Titov. Method of workaround deadlocks for solving discrete combinatorial optimization problems with constraints (in russian). In *Perspective information technologies*, pages 313–317, 2014.