VOL. 13, NO. 7, APRIL 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DESIGN OF LIBRARY OF ARTIFICIAL INTELLIGENCE ALGORITHMS FOR SOLVING THE OPTIMIZATION PROBLEMS

Vladislav A. Chekanin and Alexander V. Chekanin
Department of Theoretical Mechanics and Strength of Materials, Moscow State University of Technology «STANKIN»
3a Vadkovsky lane, Moscow, Russian Federation
E-Mail: vladchekanin@rambler.ru

ABSTRACT

The paper contains description of a unified library of artificial intelligence intended for optimization the combinatorial problems via metaheuristic algorithms. This library can be useful for practical applications in various sectors of industry and economy. The structure of the library allows creating a large number of optimization population algorithms including the genetic algorithms, the ant colony algorithms, the bees algorithms and many other algorithms based on bionic optimization models. All necessary requirements for realization such algorithms are described in details. In the paper is given the architecture of designed unified library of artificial intelligence algorithms, providing the ability of modeling a variety of intelligent algorithms and creating on its basis the combined and hybrid optimization algorithms for control problems and decision-making in organizational systems.

Keywords: library, artificial intelligence, optimization, metaheuristic algorithm, packing problem.

1. INTRODUCTION

A large number of problems arising in the automation of design and management processes belong to the wide class of optimization problems, consisting in finding of optimal solutions. Many optimization problems are combinatorial optimization problems characterized by a variety of solutions of different quality [1]. Among the classic problems of combinatorial optimization we can distinguish the traveling salesman problem, the knapsack problem, the resource allocation problem, the portfolio selection problem, the scheduling problem, the clique problem, the assignment problem, the graph coloring problem, and many other important problems. These algorithmically complex problems have a wide range of practical applications and they are relevant in automation of design and management processes in various industries and economies [1-10].

All the combinatorial optimization problems belong to the class of NP-complete problems as a result it is required to use the resource-intensive optimization algorithms to obtain the optimal solutions of the problems and it proves to be inefficient in practice due to the large expenditure of time resources [1]. One of the most prospective methods used to solve NP-complete problems of combinatorial optimization is the application of approximate metaheuristic algorithms of artificial intelligence which do not guarantee obtain of an optimal solution, while they allow to obtain a set of suboptimal solutions of acceptable quality in a relatively short period of time. At present the most efficient and interesting are the metaheuristic optimization algorithms based on bionic models borrowed from nature. Such algorithms include the genetic algorithms, the simulated annealing algorithm, the ant colony algorithm, the bees algorithm, the particle swarm algorithm, the artificial immune systems [10-18]. The effectiveness of bionic algorithms is confirmed by a number of examples of their work in nature.

In this article are presented the most important principles for constructing a unified algorithmic base in the form of a class library which will provides the possibility of realization the well known metaheuristic algorithms of artificial intelligence and it will be able to create a variety of new hybrid and modified population algorithms based on the bionic optimization ideas.

2. REQUIREMENTS TO THE LIBRARY OF ARTIFICIAL INTELLIGENCE ALGORITHMS

The library of artificial intelligence algorithms should serve as a basis for designing and researching new algorithms based on bionic optimization models. The process of creating a new bionic optimization algorithm includes the following steps:

- search in nature for an optimization idea;
- construction of a mathematical model, which is based on the found idea;
- investigation of the obtained model and its revision;
- formation of an metaheuristic and its implementation in the form of a bionic algorithm.

This process is shown on Figure-1.

As a result of designing hybrid and modified algorithms of artificial intelligence can be created new optimization algorithms which have no analogs in nature. Such algorithms will be obtained by combining and adaptive adjustment of algorithms based on different bionic models.

All bionic algorithms can be classified to the class of evolutionary algorithms, because they iteratively modify the sets of encoded solutions with forming of successive generations of solutions.

The design of a unified library of artificial intelligence algorithms requires defining of common parameters inherent for various metaheuristic optimization algorithms. On Figure-2 is shown the basic diagram which demonstrates the process of solution of optimization problems using metaheuristic algorithms.

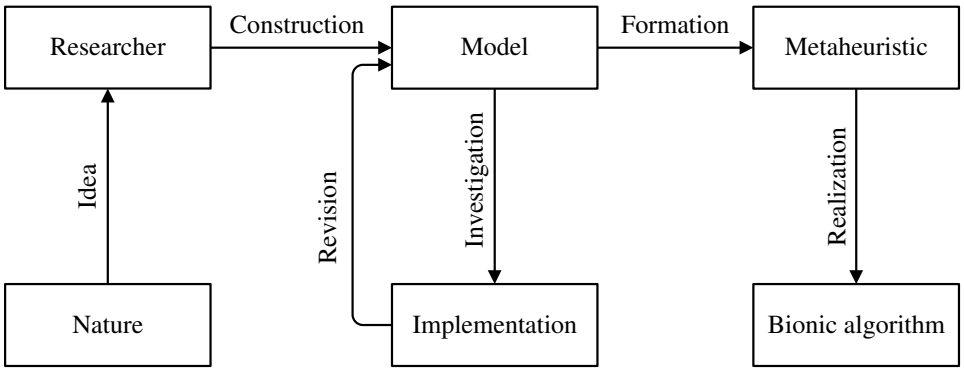

Figure-1. The process of creating a new bionic algorithm.

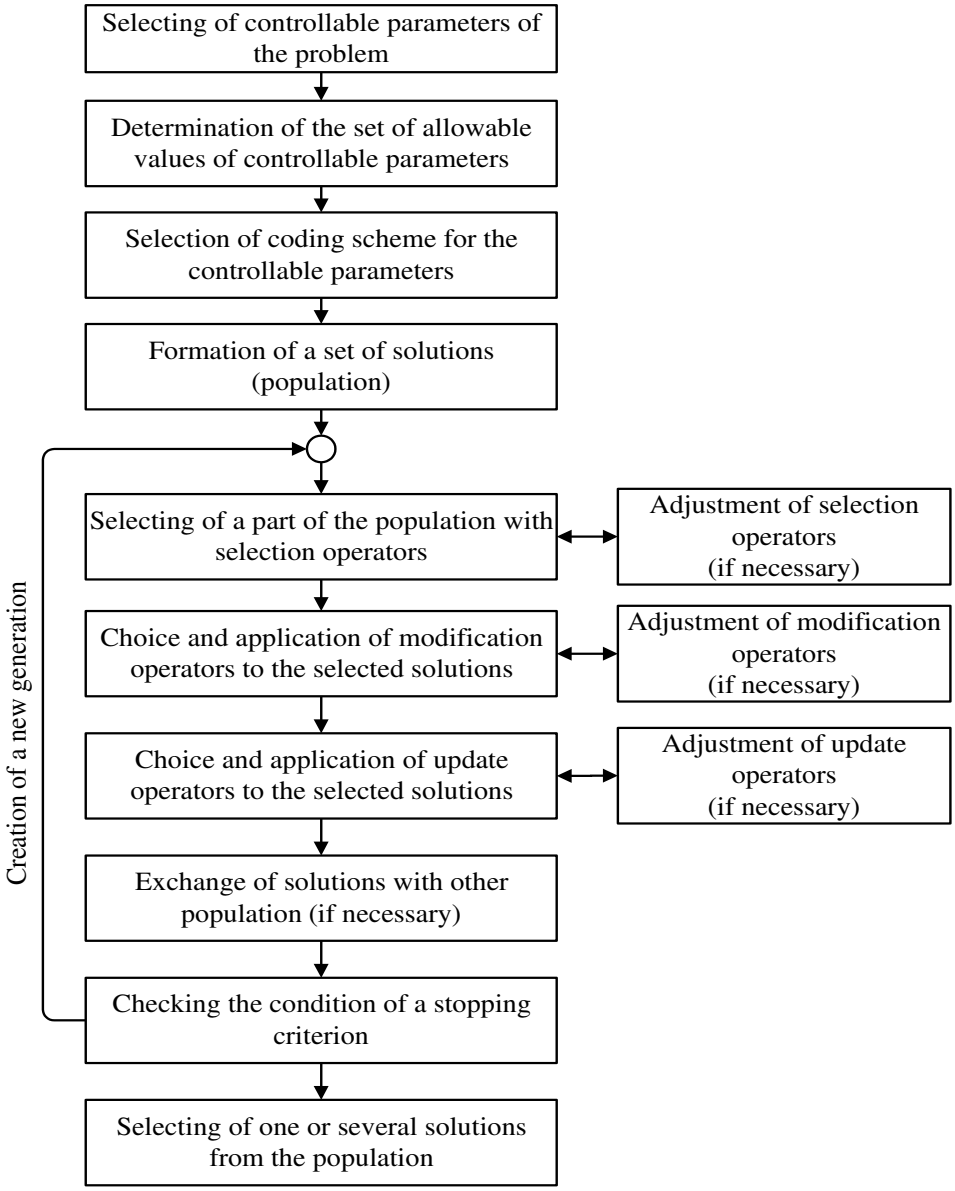

Figure-2. The process of solving an optimization problem using a metaheuristic algorithm.

The application of the metaheuristic optimization algorithm requires selecting of the following adjustable parameters:

1) a set of controllable parameters of the problem that most significantly affect the quality of the optimized solution (for example, in the packing problem one of

such parameters is the sequence of selection of the objects to be placed);

2) a coding scheme for the controllable parameters, it is possible to use the following codes:

- a binary code (the most effective in solving problems of continuous optimization);
- natural numbers (this method is most effective in solving problems of combinatorial optimization);
- other codes (for example, Gray code);

3) an optimization model which includes the following parameters:

- a set of together simulated solutions, called by a population;
- a set of together simulated populations of solutions;
- rules for creating initial populations;
- a form of fitness function as a significant factor for estimation of the quality of solutions;
- rules for selecting subpopulations for modification the solutions (the rules are given in the form of selection operators);
- operators of modification (differentiation) and updating, providing the creation of new solutions and their inclusion in the population;
- rules for changing sets of the applied modification and update operators;
- a stopping criteria for the optimization process (among the most commonly used stopping criteria can be distinguished the following: obtaining of the optimal solution, obtaining of a suboptimal solution of satisfactory quality as well as expiration of the time allotted for the solution of the problem).

Below are listed the main requirements that are defined as mandatory at the practical implementation of the universal library of artificial intelligence algorithms for solving optimization problems.

1. Requirements to the coded solutions:

- the ability of generation of coded solutions of arbitrary length;
- the ability to change the lengths of the coded solutions (particularly, it is necessary for realization of the multimethod genetic algorithm which optimizes the sequence of applied heuristics - the elementary methods of solving the problem;
- the ability of specifying a maximal lifetime for each solution.

2. Requirements to the population of solutions:

- the ability to use together solutions encoded by various methods;
- the ability to change the size of the population;
- the ability of setting the maximum number of generations of the population;
- the ability of setting the various rules for creating the initial population;
- the ability of setting the various rules for selecting solutions for its modification;
- the ability of creating the arbitrary subpopulations;
- the ability of creation the individual sets of modification and update operators for each generation;
- the ability of estimating the average fitness of the population and analyzing the dynamics of its change.

3. Requirements to the sets of populations:

- the ability of exchanging the solutions between several populations;
- the ability of including a new population into the set;
- the ability of deleting a population from the set;
- the ability of analyzing the dynamics of changes of the average fitness of the set.

4. Requirements to the selection operators:

- the ability of selecting the random set of solutions for its modification;
- the ability of applying of various standard selection methods (roulette selection method, tournament selection method, rank selection method, panmixia, genotype and phenotypic inbreeding, outbreeding, selection based on a given scale);
- the ability of adding new selection methods;
- the ability to use together several selection operators.

5. Requirements to the modification and update operators:

- the ability of application the single-point, two-point and multi-point crossover operators to the coded solutions;
- the ability of uniform crossing of the coded solutions by the given scheme;
- the ability of mutation of various parts of the coded solutions;
- the ability of mutation by a rule given in the form of a function;
- the ability of recalculation the fitness function of the selected solutions by some rule;
- the ability of generation a new random solution;
- the ability of changing the probability of any operator at any time.
- the ability of adding new modification and update operators.

6. Requirements to the optimization algorithms (choice of strategies for finding the solutions):

- support of standard evolution models (the models based on evolution theories proposed by Charles Darwin, Jean-Baptiste Lamarck, Hugo de Vries, Karl Popper, Motoo Kimura);
- the ability of modification of the standard evolution models in the solving process;
- the ability of adaptive adjustment of the used strategies of searching the optimal solution.

3. DESIGN OF LIBRARY OF ARTIFICIAL INTELLIGENCE ALGORITHMS

In the program version of the designed library of artificial intelligence algorithms should be implemented the proposed idea of setting rules for distribution and exchange of solutions between populations in the process of optimization. These rules can be specified in the form of schemes showing the direction of exchange of solutions and the percentage composition of the populations participating in the exchange. On Figure-3 is shown an example of scheme for copying 10% of the best solutions from the population A in the populations B, C, and D.

Figure-4 shows a scheme for copying the two best solutions from each population to the neighboring populations. Figure-5 demonstrates an example of an arbitrary exchange of solutions between populations. On Figures 3-5 the following notation is used:

- the arrow indicates the direction of copying the solutions;
- the text on the arrow corresponds to the number or proportion of solutions copied to another population;
- the line type of the arrow corresponds to the method of choosing the solutions for copying (solid line means choosing the best solutions, while dotting line means the random selection of solutions).

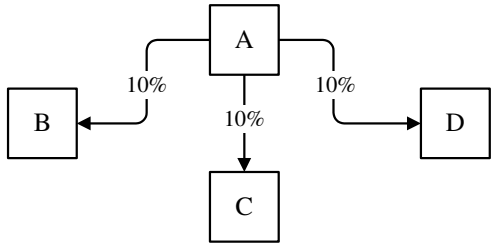

Figure-3. Scheme of uniform distribution of the best solutions.

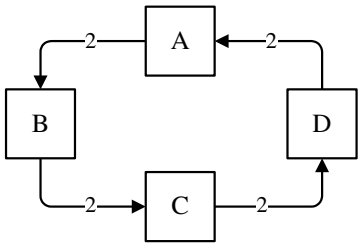

Figure-4. Scheme of uniform exchange of the best solutions between populations.

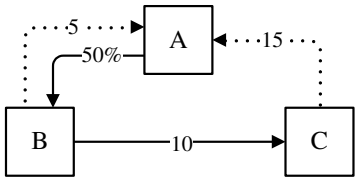

Figure-5. Scheme of random exchange of solutions between populations.

A designed universal library of artificial intelligence algorithms will support the following requirements for rules of exchange of solutions between populations:

- the ability of exchange the solutions;
- the ability of copying only best solutions which are absent in the receiver's population;
- the ability of replacement the worst solutions in the receiver's population;
- the ability of uniform mixing of all solutions in populations;
- the ability of copying of all solutions into another population;
- the ability of ignoring exchange and distribution rules under certain conditions;
- the ability of changing the exchange and distribution rules.

The proposed library of algorithms should provide the ability of creating new combined optimization algorithms from several basic algorithms, thus forming ensembles of algorithms. When creating an ensemble of algorithms, it is necessary to select a set of jointly working algorithms, as well as to define rules for the exchange of solutions (migration rules) between the algorithms.

The simplest example of an ensemble of algorithms constructed on the basis of one optimization algorithm is a parallel genetic algorithm. Parallel genetic algorithms are based on the partitioning of the original population of solutions into several separate subpopulations, each of which is processed by a separate genetic algorithm (GA) independently of other subpopulations. In this algorithm the exchange of solutions between subpopulations is possible only at the moment of transition to the next its generation. On Figure-6 is given an example of combining a set of genetic algorithms into a parallel genetic algorithm.

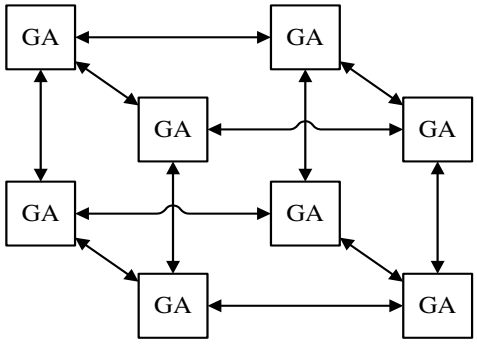

Figure-6. A parallel genetic algorithm as an example of an ensemble of algorithms.

Ensembles of algorithms provide an opportunity to exit from local optimums due to the cooperative usage of optimization algorithms that have different parameters of selection, modification and updating of their sets of populations.

The program implementation of the designed library is planned to be carried out using the object-oriented algorithmic language C++ and the standard template library (STL). When designing the library of artificial intelligence algorithms, the experience of creating a library for optimization resource allocation problems was taken into account [19, 20].

The library of artificial intelligence algorithms, designed as a result of the analysis of the described requirements, is shown in the form of the UML diagram on Figure-7. To increase the visibility of the diagram, not all fields and methods that realize the described abilities of the library are shown on it.

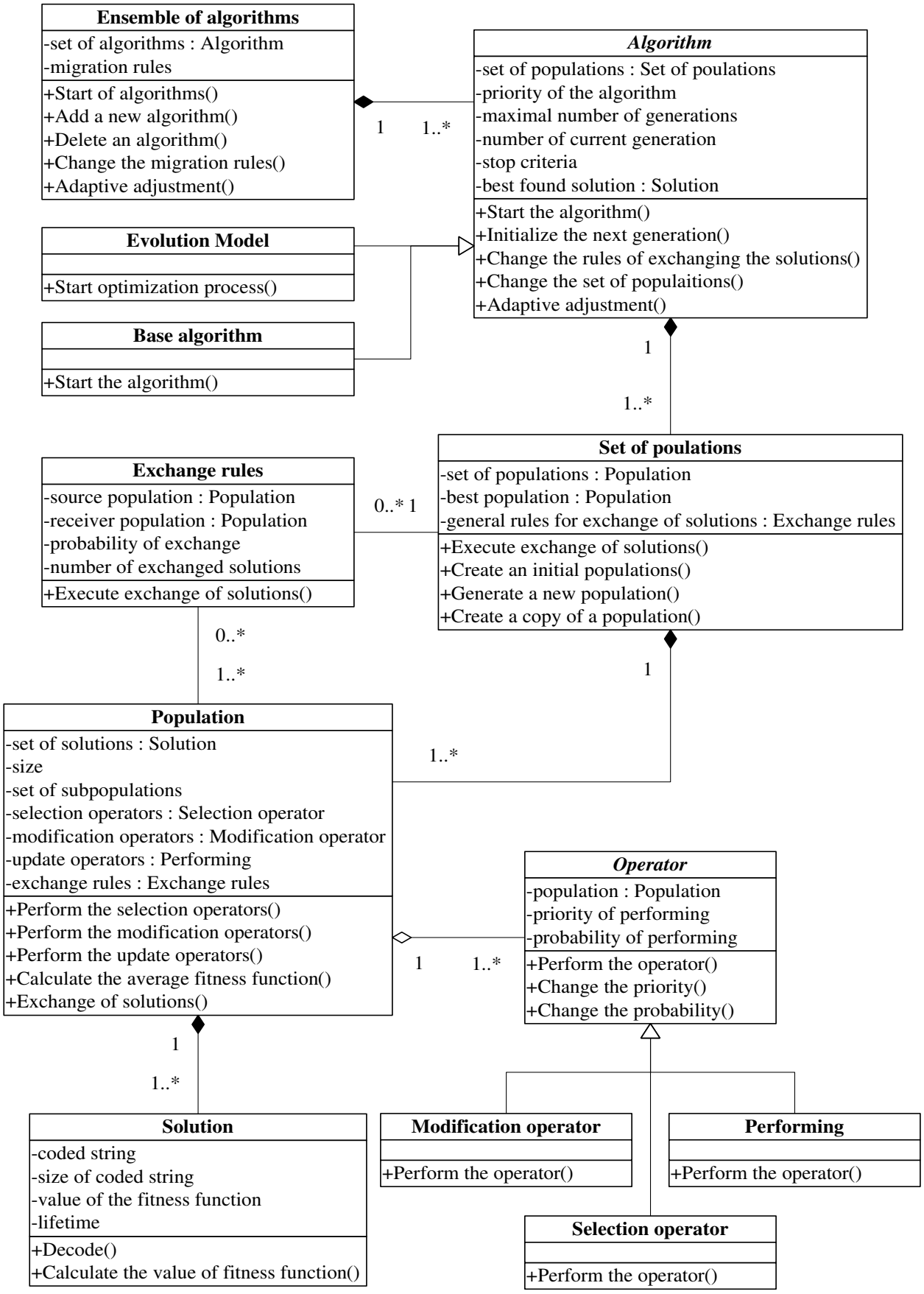

Figure-7. UML diagram of the designed library of artificial intelligence algorithms.

4. CONCLUSIONS

As a result of the research, the most significant requirements for the universal library of artificial intelligence algorithms based on the use of bionic optimization models were determined and classified. The output of the performed design is the unified library of algorithms which has been implemented on the unified modeling language UML in the form of a class diagram. The library of algorithms can be applied for searching for the rational solutions of various optimization practical problems related with automation of design and management processes in various industries and economies. Further development of this research includes the software implementation of the described library of algorithms and its usage both for solving practical problems and for creating and analyzing hybrid and combined optimization algorithms.

ACKNOWLEDGEMENTS

This work was carried out with the financial support of the Ministry of Education and Science of Russian Federation in the framework of the state task in the field of scientific activity of MSTU «STANKIN».

REFERENCES

[1] D. S. Johnson. 2012. A Brief History of NP-Completeness, 1954-2012. Documenta Mathematica. Extra Volume ISMP: 359-376.

[2] G. Wascher, H. Haubner and H. Schumann. 2007. An improved typology of cutting and packing problems. European Journal of Operational Research. 183(3): 1109-1130.

[3] A. Bortfeldt and G. Wascher. 2013. Constraints in container loading - a state-of-the-art review. European Journal of Operational Research. 229(1): 1-20.

[4] V. A. Chekanin and A. V. Chekanin. 2015. An efficient model for the orthogonal packing problem. Advances in Mechanical Engineering. 22: 33-38.

[5] V. A. Chekanin and A. V. Chekanin. 2014. Multilevel linked data structure for the multidimensional orthogonal packing problem. Applied Mechanics and Materials. 598: 387-391.

[6] V. A. Chekanin and A. V. Chekanin. 2014. Improved data structure for the orthogonal packing problem. Advanced Materials Research. 945-949: 3143-3146.

[7] V. A. Chekanin, A. V. Chekanin. 2016. Algorithms for management objects in orthogonal packing problems. ARPN Journal of Engineering and Applied Sciences. 11(13): 8436-8446.

[8] V. A. Chekanin, A. V. Chekanin. 2016. Implementation of packing methods for the orthogonal packing problems. Journal of Theoretical and Applied Information Technology. 88(3): 421-430.

[9] V. A. Chekanin, A. V. Chekanin. 2017. Deleting Objects Algorithm for the Optimization of Orthogonal Packing Problems. Advances in Mechanical Engineering. Springer International Publishing. pp. 27-35.

[10] Y. Q. Gao, H. B. Guan, Z. W. Qi, Y. Hou and L. Liu. 2013. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. Journal of Computer and System Sciences. 79(8): 1230-1242.

[11] S. C. H. Leung, D. F. Zhang, C. L. Zhou and T. Wu. 2012. A hybrid simulated annealing metaheuristic algorithm for the two-dimensional knapsack packing problem. Computers & Operations Research. 39(1): 64-73.

[12] V. A. Chekanin and A. V. Chekanin. 2013. Efficient Algorithms for Orthogonal Packing Problems. Computational Mathematics and Mathematical Physics. 53(10): 1457-1465.

[13] V. A. Chekanin and A. V. Chekanin. 2014. Development of the multimethod genetic algorithm for the strip packing problem. Applied Mechanics and Materials. 598: 377-381.

[14] Kierkosz and M. Luczak. 2014. A hybrid evolutionary algorithm for the two-dimensional packing problem. Central European Journal of Operations Research. 22(4): 729-753.

[15] D. Chen, J. Liu, Y. Fu and M. Shang. 2010. An efficient heuristic algorithm for arbitrary shaped rectilinear block packing problem. Computers & Operations Research. 37(6): 1068-1074.

[16] F. Goncalves and M. G. C. Resende. 2013. A biased random key genetic algorithm for 2d and 3d bin packing problems. International Journal of Production Economics. 145(2): 500-510.

[17] Łapa, J. Szczypta, R. Venkatesan. 2015. Aspects of structure and parameters selection of control systems using selected multi-population algorithms. Artificial Intelligence and Soft Computing. 9120: 247-260.

[18] S. K. Nseef, S. Abdullah, A. Turky and G. Kendall 2016. An adaptive multi-population artificial bee colony algorithm for dynamic optimisation problems. Knowledge-Based Systems. 104: 14-23.

[19] V. A. Chekanin and A. V. Chekanin. 2015. Development of optimization software to solve practical packing and cutting problems. Advances in Intelligent Systems Research. 123: 379-382.

[20] V. A. Chekanin, A. V. Chekanin. 2017. Algorithms and methods in multidimensional orthogonal packing problems. International Journal of Applied Engineering Research. 12(6): 1009-1019.