Modified genetic algorithm as a new approach for solving the problem of 3d packaging

Vladimir Mokshin
Kazan National Research Technical University named after A. N. Tupolev - KAI
Kazan, Russia
vladimir.mokshin@gmail.com

Darya Maryashina
Kazan National Research Technical University named after A. N. Tupolev - KAI
Kazan, Russia
maryashina.darya@yandex.ru

Nikita Stadnik
Kazan National Research Technical University named after A. N. Tupolev - KAI
Kazan, Russia
erter.live@gmail.com

Alexander Zolotukhin
Kazan National Research Technical University named after A. N. Tupolev - KAI
Kazan, Russia
avol116@yandex.ru

Leonid Sharnin
Kazan National Research Technical University named after A. N. Tupolev - KAI
Kazan, Russia
sharnin_lm@mail.ru

***Abstract*—In this paper, we proposed one of the options for developing a new evolutionary heuristic approach for solving the three-dimensional packing problem called BPP (Bin packing problem), as applied to the variation of this problem with a single container and a set of boxes of various dimensions, called the SKP (Single knapsack problem), and the comparison of 11 basic evolutionary heuristic approaches to solving the problem of three-dimensional packing of BPP (Bin packing problem) variations SKP (Single knapsack problem) with the developed new evolutionary heuristic approach to solving BPP using modi cited genetic algorithm (MGA). By performing correlation and statistical analysis using 10 randomly created sets of input data for solving BPP, the effectiveness of MGAs was proved in comparison with 11 basic evolutionary algorithms for solving BPP. Thus, it was confirmed that MGA and similar algorithms can be effectively used to solve such logistic NP-difficult problems.**

***Keywords*—modelling, genetic algorithm, 3d packing**

I. Introduction

In tough competitive market relations, each company seeks to reduce the cost of its products without compromising on quality. One of the factors affecting the cost of a product is packaging and distribution. In modern freight transportation, the packaging problem is an important applied section of transport logistics and allows you to solve many practical problems in the management, automation and optimization of cargo transportation.

The three-dimensional packing problem is a well-known NP-hard problem, both exact and evolutionary approaches are used to solve various variations of it. The most popular variations of the packing problem are RBPP (Residual bin packing problem), when it is required to pack different box sizes in different containers, and SKP (Single knapsack problem), when it is required to pack different types of boxes in one container. This article describes an algorithm for solving the problem of three-dimensional packing of the SKP variation (Single knapsack problem) from one container and N boxes.

Today, there are a number of evolutionary algorithms based on the mechanism of biological evolution and used to solve packaging problems. Let's consider each algorithm separately.

II. Evolutionary algorithms

A. Ant algorithm

Ant Algorithm (ACO) is one of the effective bionic methods and packaging algorithms. ACO is based on the principle of a multi-agent method of intellectual optimization based on modeling the behavior of an ant colony. Each ant individually represents a low-level unit, however, in general, an ant colony is a rational multi-agent system. An ant acts as an agent, looking for an optimal path between its ant hill and a food source. The following terms apply to ACO when solving sequential packaging tasks: ACO nodes are the points of space inside the container, and the paths along which ants move are the trajectories of the boxes inside the container to the places of their final packaging (nodes). Each ant begins its path from the zero node (ant nest, which is located in the lower left far corner of the container). If the ant cannot pack the boxes in the current node, then it goes to the next node other than zero. The ants move around the nodes until all the boxes are packed in a container with the maximum possible filling of each node [1].

B. Annealing simulation algorithm

The annealing simulation algorithm (SA) is associated with the methods of simulation modelling (SM) in statistical physics. The algorithm is based on the process of metal annealing in metallurgy, which consists in slow cooling of the material to increase its strength and reduce defects. The annealing process in metalworking can be described as follows: the temperature of the metal increases until it begins to melt in the heat bath, i.e. until the end of the process of complete transition of the metal into a liquid state of aggregation. After this, the metal in a liquid state is slowly cooled, i.e. its temperature is gradually and carefully reduced until the particles return to their original state of aggregation. As applied to the problems of three-dimensional packing, the main purpose of applying the annealing simulation algorithm is to minimize the free space of the container (i.e., to find the best way to pack the boxes into the container). The objective function of the algorithm can be described as follows: $(V_{total} - V_{useful}) \to \min$, where V_{total} is the total volume of the container; V_{useful} is the useful volume occupied by the boxes, i.e. the volume of the figure, consisting of the closest points of the boxes to coordinate zero and the farthest points of the boxes from coordinate zero [2], [3].

C. Tabu Search n algorithm

Tabu Search (TS) is a mathematical optimization method that uses the local search method. Unlike the local search method, TS has a higher productivity. The prohibition search process is characterized by a set of states (options for three-dimensional packing of boxes in a container), and at each stage (step), a transition is made from the current state to one of the neighboring ones. Performance improvements are achieved through the use of prohibition lists. Once a potential solution (the most optimal option from previously found options for three-dimensional packaging of boxes in a container) has been found, it is placed in the prohibition list, that is, marked as “taboo”, which reduces the search time due to the ban on “visiting” earlier discovered solutions. After this operation, the local search process continues until a new improved solution is found [4].

D. Guided Local Search

Guided Local Search (GLS) is one of the different types of searches with bans. The basis of GLS is metaheuristics, which, like TS, uses memory (a list of previously obtained solutions) to control the search process. The managed local search algorithm is a local search option in which components are searched, often leading to a local minimum of the objective function. As applied to the packing problem, the objective function of the algorithm can be written as follows: $f(x) = (V_{total} - V_{useful})$, where V_{total} is the total volume of the container; V_{useful} is the useful volume occupied by the boxes, i.e. volume of the figure consisting of the closest points of the boxes to coordinate zero and the farthest points of the boxes from coordinate zero. Then, solutions using these components are fined, thereby enhancing the research of the search space, leading to solutions leading to a global minimum of the objective function [5].

E. Fast Local Search

Fast Local Search (FLS) is an improved version of managed local search. Unlike GLS, a Fast Local Search breaks the container's fillable area into several smaller subdomains. Each such formed subdomain can be in one of two states: active or inactive. By default, all subregions are in an active state. According to a certain order (dynamic or statistical), the quick local search algorithm visits active regions of the container with the goal of packing boxes into them. Next, the subdomain is checked for subneighborhood: if there is no better option for the algorithm to move (packing boxes into neighboring subdomains), then the current subdomain is filled with boxes and becomes inactive, otherwise an improving move is performed - visiting another region. One of the advantages of FLS is the ability to reactivate subdomains. If we assume that a number of subregions previously converted to inactive status may contain improving moves, taking into account the just completed move, then such subregions can be reactivated - become active again. When all areas become inactive, the best solution to the three-dimensional packing problem will be found, i.e. the FLS algorithm will come to its global minimum [5].

F. Local search with alternating surroundings

Local search with alternating neighborhoods (VNS) is one of the methods for solving discrete optimization problems. One of the differences between the VNS method and simple local search is the systematic change in the appearance of the surrounding area during local search. The basic algorithm for local search with alternating neighborhoods can be described as follows: for some preliminary solution to the three-dimensional packing problem (the supposedly optimal variant of packing boxes in a container), its many neighborhoods are determined: other slightly different options for packing boxes in a container. Then, from this set of neighborhoods, another type of neighborhood is randomly selected. The search for an improved solution for the selected neighborhood is carried out using the local search algorithm. At the next stage, the algorithm branches: if an improved solution is found, then the preliminary solution is replaced by the value of the new solution, the search continues in the same neighborhood. Otherwise, a new neighborhood is selected from the set of neighborhoods of the preliminary solution, and the algorithm continues to run. The stopping criterion for the VNS algorithm can be the number of iterations in which no improvement of the solution was achieved, a certain number of iterations, or the fact that the optimal solution was found [6], [7].

G. Greedy randomized adaptive search procedure

The greedy algorithm with random adaptive search (GRAPS) improves combinatorial solutions obtained by constructing individual components. The GRAPS algorithm can be described in three steps. At the first step, the maximum space for filling is selected, i.e. the total volume of the container. The corners of the container are filled first, then the sides, and at the very end the interior space is filled. At the second step, from a set of boxes in a lexicographic order, boxes for packaging are selected. The choice is made taking into account two criteria: those that fit in the maximum space in the best way and which give the greatest increase in the total volume of boxes. In the third step, the list of maximum spaces is updated, since any packaging made of at least one box leads to changes in the maximum space. The change in the value of the current maximum space is calculated by the formula: $V_{\max} = V_{total} - V_{useful}$, where $V_{\max}$ is the current maximum space, V_{total} is the total volume of the container, V_{useful} the useful volume occupied by the boxes, i.e. volume of the figure consisting of the closest points of the boxes to coordinate zero and the farthest points of the boxes from coordinate zero. After that, the algorithm cycle returns to the first step and continues the process of packing boxes until all boxes are packed into a container [8].

H. Best Fit Decreasing (BFD) and First Fit Decreasing (FFD)

The Best Matching Descending (BFD) and First Matching Descending (FFD) algorithms are the simplest polynomial algorithms used to solve packing problems. Both algorithms sort objects by not increasing their volumes and sequentially put them in containers. BFD and FFD differ from each other in the way they select the container into which the boxes will be packed. According to the best descending algorithm, the boxes will be packed in the container that will have the smallest free space after loading. In the algorithm of the first descending box, the boxes are loaded into the first container into which they fit in volume [9].

I. Particle Swarm Optimization (PSO)

The Particle Swarm Optimization Method (PSO) is based on modeling the behavior of a flock of birds. The idea of PSO is that each particle is a possible solution to the optimization problem, as applied to the three-dimensional packing problem, a possible optimal option for packing boxes in a container. Particles "fly" over the solution space of the function and try to find its global minimum (the best solution to the packaging problem), taking into account their own knowledge and the experience of their neighbors. The principle of "flight" of particles is based on the gradient descent method, the method of finding a local minimum by moving the particle along the gradient, and the space of vectors indicating the path to the greatest decrease in the objective function [10].

III. Adaptive Genetic Algorithm (GA)

The genetic algorithm is based on modelling the mechanisms of natural selection in nature. Its main operators are evolutionary methods such as inheritance, mutation, crossingover, and selection. At the initial stage of the algorithm, an initial population of *X* chromosomes (individuals) *H* is formed, consisting of a set of *p* genes. Then, two chromosomes H_n and H_n are randomly selected from the original population and two new chromosomes H_n^{new} and H_k^{new} are created by crossing (mutually exchanging chromosome regions) chromosomes H_n and H_n. At the next step of the algorithm, mutations of the H_n^{new} and H_k^{new} chromosome genes with a random probability *ρ* occur. Upon completion of the mutation process, the chromosomes H_n^{new} and H_k^{new} become a new part of the *X* population. After *N* steps of this algorithm, the best chromosomes that have the set of genes describing the optimal solution necessary for solving the optimization problem are selected from the population *X* extended by the methods of inheritance, mutation, and crossing over. The main problem of packaging problems is their complexity and the impossibility of solving such problems using deterministic polynomial algorithms due to the large time and computational costs, therefore, the search and development of new methods and algorithms for solving packaging problems do not lose their importance and relevance. The article discusses the use of a new modified genetic algorithm, the practical significance of which is shown by solving the problem of packing rectangular boxes in a container with maximum compactness, taking into account the priority of unloading at delivery points. As a container, we consider a part of three-dimensional space limited by a width *W*, a depth *D*, and a height *H* having a volume *M*. As boxes, we consider *N* blocks limited by our own parameters for the width, depth, and height that must be placed in the volume *M*. We describe the location of the container in space using eight points $\{X_0, Y_0, Z_0, \ldots, X_7, Y_7, Z_7\}$, where X_0, Y_0, Z_0 are equal to zero, and X_7, Y_7, Z_7 are respectively equal to *W*, *D*, *N*. The arrangement of blocks inside the container is also we describe eight points $\{x_0, y_0, z_0, \ldots, x_7, y_7, z_7\}$ according to three conditions:

Blocks cannot go beyond: $\left\{\begin{array}{l} x_{ij} \le W \\ y_{ij} \le D \\ z_{ij} \le H \end{array}\right\} \quad \forall j = \overline{0,7}$, $\forall i = \overline{1,N}$.

The total volume of blocks may not exceed the volume of the container: $\sum_{i=1}^{N} V_i \le W \cdot D \cdot H$, where V_i is the volume of the *i*-th block.

Blocks cannot overlap each other: $\left\{\begin{array}{l} x_{ij} + x_{i(j+k)} \le x_{(i+1)j} \\ y_{ij} + y_{i(j+k)} \le y_{(i+1)j} \\ z_{ij} + z_{i(j+k)} \le z_{(i+1)j} \end{array}\right\} \quad \forall j,k = \overline{0,7}, \ \forall i = \overline{1,N}, \ \forall k \ne j$.

The collisions of blocks will be determined in accordance with the model's limitations by checking the condition of overlapping blocks on each other by comparing the coordinates of the farthest corner of an already placed block and the nearest corner of the placed (new block). As an optimization target (objective function), we will use the ratio of the usable volume to the volume of the container, which we denote by $F = \frac{V_{useful}}{V_{box}} \to 1$.

Its significance will seek unity with the disappearance of voids. The modified genetic algorithm (MGA) is based on the principles of the genetic algorithm, however, it has modified behavior patterns such as adaptive mutation and the LBFL model for generating the initial chromosome population.

The LBFL model (Late-Botton-Front-Left, Late-Lower-Front-Left) is an imperfect algorithm for arranging (sorting) blocks according to 4 priority levels: unloading time (late-early), height (lower-upper), depth (front-back), width (left-right) [11].

This algorithm generates the order of packing blocks into a container according to the priority list. The latest block at the time of unloading is always placed at the origin. All subsequent blocks, sorted by the time of unloading, are sequentially moved to the upper rear right corner, and then with the help of movement and rotation in 6 variants fill the entire lower level, then the front, then the left [13]. Filling these levels as evenly as possible, the blocks fill the remaining container volume in the same way as a separate empty container, limited by the maximum width, depth and height available for it, continuing the algorithm of the LBFL model. This will continue until all the blocks are laid in a container and the objective function is maximized. Fig. 1 shows the result of the generation of the initial chromosome population using the LBFL model of the arrangement of blocks according to the priority list in three-dimensional space of width W, depth D and height H.

As an algorithm for mutating chromosomes, the adaptive probability of mutation was used. This allowed us to overcome the high probability of population convergence at a local optimum.

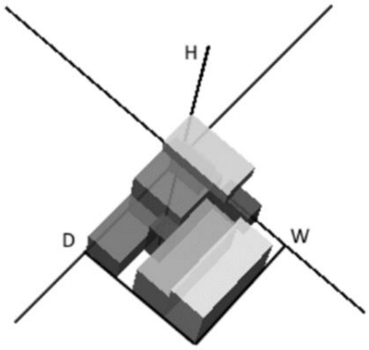

Fig. 1. The result of the algorithm of the LBFL model.

When using it, the degree of mutation for each individual varied depending on the average indicator of the objective function $F = \frac{V_{useful}}{V_{box}}$ obtained from each of the crossing chromosomes.

The closer this indicator was to the global optimum (the best solution to the three-dimensional packing problem, at which $F = 1$), the less the mutation probability. And, on the contrary, the farther the indicator was, the more likely the chromosome mutation was.

In mathematical form, the adaptive probability of mutations of the chromosomes x_1^k and x_2^k of the *k*-th population can be expressed through the eq. (1):

$$\rho = 1 - \frac{F(x_1^{k-1}) + \mathrm{F}(x_2^{k-1})}{2} \qquad (1)$$

To test the effectiveness of this modification of the MGA, a test of the operation of the MGA algorithm was carried out on ten canonical independent sets of source data consisting of 50 boxes of 5 different types. During testing, changes in the adaptive probability of a mutation were evaluated. In Fig. 2. The test results are presented in the form of a graph of the dependence of the mutation probability on the time of the algorithm operation.

As applied to the packaging problem, MGA is a collection of H chromosomes (individuals) consisting of p genes. Where each chromosome Hi is an imperfect algorithm of the i-th arrangement of blocks (boxes), and the pj gene are parameters (a tuple of coordinates of all eight points) of the j-th block [11]. The course of the algorithm is a sequence of operations (mutations) over a set of chromosomes, oriented towards achieving the maximum indicator of the optimization function (in relation to the task of three-dimensional packaging - achieving the maximum indicator of the objective function of the ratio of usable volume to the volume of the container $F = \frac{V_{useful}}{V_{box}} \to 1$ at least one of the chromosomes.

For a more detailed consideration of the algorithm, it can be structurally divided into six stages: generation of individuals, formation of chromosomes, calculation of the objective function, selection of chromosomes, crossing and detection of mutations.

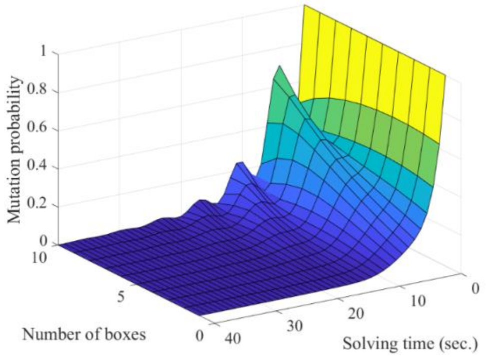

Fig. 2. The dependence of the adaptive probability of a mutation on the time of the algorithm.

At the initial stage, the generation of individuals of the population occurs (the creation of many blocks, the sizes of which are randomly selected from ranges that satisfy the given conditions).

The next step is the formation of chromosomes - various sequences of arrangement of blocks. The number of chromosomes is selected in the range from 2 to $2 \cdot N$, where *N* is the number of blocks.

Next, the main process of the modified genetic algorithm begins - the calculation of the objective function $F = \frac{V_{useful}}{V_{box}}$ for each chromosome, where V_{useful} is the net volume occupied by the blocks, i.e. the volume of the figure, consisting of the closest points of the blocks to coordinate zero and the farthest points of the blocks from coordinate zero, and V_{box} is the volume of the container.

In addition to the objective function, the density function is calculated as $P = \frac{\sum_{i=1}^{N} V_{box}^{i}}{V_{usefull}}$, where V_{box}^{i} is a volume of the *i*-th block, V_{useful} is an occupied space after decoding (usable volume). The density function describes the ratio of the total volume of all blocks to the usable volume and tends to 1 with a decrease in the gaps between the blocks. The next step after completing the first cycle of the main GA process is the selection of chromosomes (selection of chromosomes with the best values of the objective function from the total number of chromosomes in the population). In this work, we used the selection method based on the tournament table: the total number of chromosomes is divided into subgroups with the number of individuals from 2 to 4, and then a chromosome with the best objective function is selected from each subgroup. This method allows you to create a new and objectively better population for the next cycle of the main GA process.

After chromosome selection, their crossing follows - the process of creating two new descendant chromosomes by combining parts of the parent chromosome genes. For this, a random gene is selected in the gene chain of each of their parent chromosomes p_r, first descendant chromosome H_{i1} is made up of genes $p_k\ \forall k = \overline{1,r}$ of the first parent and genes $p_m\ \forall m = \overline{r,j}$ of the second parent, second chromosome descendant H_{i2} made up of genes $p_k\ \forall k = \overline{r,j}$ of first parent genes and genes $p_m\ \forall m = \overline{1,r}$ of the second parent, where j is a total number of genes in parent chromosomes (number of blocks).

The next step after crossing is to add mutations - rare changes in the gene values at random with an adaptive probability value ρ. The probability ρ_{ij} of the mutation of the gene $\rho_j$of the chromosome H_i depends on the fitness of the individual (chromosome H_i), expressed by the value of the objective function $F = \frac{V_{useful}}{V_{box}}$ for this chromosome. The worse the individual is adapted, i.e. the further the value of the objective function is from unity, the higher the probability of mutation of its genes becomes, so that it can optimize the current value of the objective function. In the packaging problem under consideration, the adaptive mutation of the p_j gene is to move the j-th block to the beginning of the priority list, regardless of its size and initial priority of unloading.

IV. Comparison of methods

A comparison of methods for solving the three-dimensional packing problem showed the following results. The BFD algorithm proved to be the most resource-intensive algorithm, having slightly lost in performance to the FFD algorithm. These algorithms are the simplest heuristic approaches to solving the problem of three-dimensional packaging, therefore, are at the end of the performance list. The annealing simulation algorithm (SA) showed the most expression regarding the FFD algorithm, however, it lost in performance to the following search algorithms: VNS, TS, GLS, FLS, whose indicators are also sorted in order of improving performance, however, they differ slightly from each other. Search algorithms can quickly converge to a local minimum in the neighborhood of solutions, choosing both from the full version (TS, GLS), and segmented (VNS, FLS). The performance of FLS bypassed the ant colony optimization (ACO), which works on the principle of an ant colony, with a large gap, followed by the PSO, which describes the simulation of the life of a "swarm of particles", in its behavior similar to a bee swarm. GA, GRASP (a greedy algorithm with random adaptive search, which allows to find a multitude of local minima of the objective function and based on them to suggest the optimal solution) and MGA (a modified genetic algorithm that allowed to increase the speed of the standard genetic algorithm and get around in GRASP performance).

To numerically evaluate the effectiveness of the packaging methods, special test data sets have been developed depending on the type of problem being solved. In this work, ten canonical sets were used, consisting of 1000 boxes of 50 types and a single container of constant sizes, but different among the examples. Two performance indicators were evaluated: the running time of the algorithm and the density of the resulting packaging in decimal units.

In Fig. 3-4 are comparative graphs of the results of the assessment of both performance indicators. The indicators were evaluated as follows: each of the 12 packaging methods was tested on ten canonical sets independent of each other, during testing, two performance indicators were evaluated: the time the algorithm worked in minutes and the density of the resulting package in percent. Then, for each algorithm, the total packing time in minutes was calculated from the total packing time of all ten sets and the average packing density by the average packing density among all ten sets.

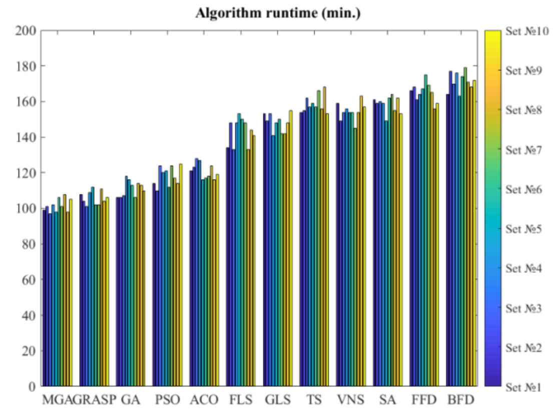

Fig. 3. Algorithm runtime evaluation results.

For an objective assessment of three-dimensional packaging methods, it was proposed to introduce a new variable (2):

$$\varphi = \sum_{i=0}^{n} \frac{\tau_i}{\rho_i} \cdot 0.01 \qquad (2)$$

where τ_i packing time of the i-th set, ρ_i is the density of the obtained packaging of the *i*-th set, and *n* is the number of sets in the experiment, which mathematically illustrates the degree of effectiveness of the three-dimensional packing method (the lower its value, the higher the efficiency of the algorithm).

Comparison of three-dimensional packaging methods was carried out according to three objective performance indicators: the total preparation time, average packing density and the value of the variable φ . The results of the comparison of three-dimensional packaging methods are presented in table 1.

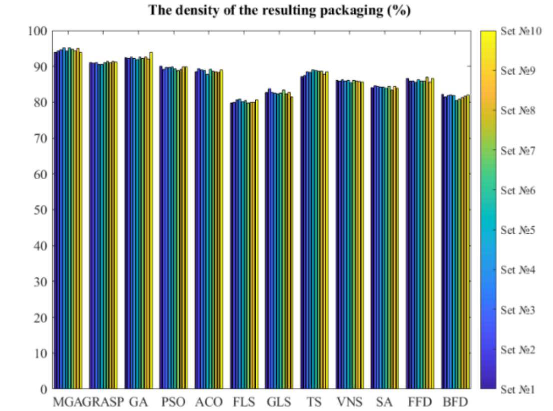

Fig. 4. The results of the assessment of the density of the resulting packaging.

TABLE I. 3D Comparison Results

Method	Total packing time (minute)	Average packing density (%)	Value φ
MGA	1015	94.56	1073.3926
GRASP	1059	91.03	1163.3527
GA	1109	92.48	1199.1782
PSO	1181	89.54	1318.9636
ACO	1209	88.73	1362.5606
FLS	1432	80.25	1784.4237
GLS	1481	82.68	1791.2433
TS	1587	88.31	1797.0785
VNS	1545	85.93	1797.9751
SA	1584	84.19	1881.4586
FFD	1650	86.17	1914.8195
BFD	1714	81.56	2101.5204

V. Software implementation

For the software implementation of the developed MGA, a C # language application was written in Microsoft Visual Studio 2017 for a user of 32-bit and 64-bit versions of Windows XP and higher, which solves the three-dimensional packaging problem and graphically illustrates the final result as a three-dimensional container model with rectangular boxes located in it optimally.

In the simulated example, the task of packing 50 different types of boxes in a 40-pound cargo container of ISO standard was considered.

The algorithm of modified genetic algorithm:

```
Variables Initialization:
$chr$ is a number of chromosomes (population size) $maxiter$ is a maximum number of iterations (parameter for while loop)
Begining:
$k = 1$
Creating an initial population $X^k = \{x_1^k, x_1^k, \dots, x_{chr}^k\}$ using LBFL model
The calculation of the objective function $F(x_i^k)$ for $i = 1, \dots, chr$
Loop while($x^* =$ true):
The cycle beginning:
    $k = k + 1$
    $Loop$ for(iter $= 1$; iter $\le \frac{\text{chr}}{2}$; $iter = iter + 1$):
    The cycle beginning
        Random selection of two chromosomes $x_p$ and $x_q$ of $X^{k-1}$
        Two new chromosomes creation
        $x_p^{new}$ и $x_q^{new}$ as crossingover $x_p$ и $x_q$ with probability $p_c$
        The calculation of the adaptive probability of chromosome mutations $x_p^{new}$ и $x_q^{new}$
        $p_m = 1 - \frac{F(x_p)+F(x_q)}{2}$
        Mutation operation $x_p^{new}$ and $x_q^{new}$ with probability $p_m$
         Objective functions calculation $F(x_p^{new})$ and $F(x_q^{new})$
        Adding $x_p^{new}$ and $x_q^{new}$ into $X^{new}$
    The cycle end.
    Selecting of the best chromosomes from $X^{k-1}$ and $X^{new}$ for $X^k$
    $x^* =$ the best chromosome in $X^k$
    The cycle $while(k \le maxiter$ and $F(x^*) < 1)$:
    The cycle beginning:
    Return $x^*$
    End of cycle.
End of cycle.
```

The result of the MGA: a three-dimensional model of the best individual (the best option for packing boxes in a container) is shown in Fig. 5.

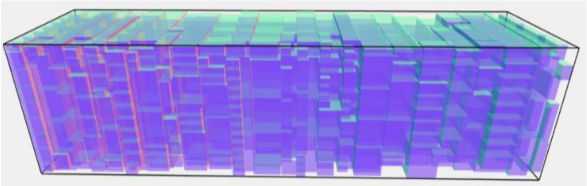

Fig. 5. The result of MGA work of with input data from the 1st set.

As a result of the MGA’s work on the first set, 1000 boxes of 50 types were packed into a 40-foot container, the packing density was 93.96%, and the total algorithm running time was 99 minutes. The best result for the packing density criterion was obtained by the MGA with input data of the 6th set, which amounted to 95.16%. The best result for the total operating time criterion was obtained by the MGA with input data of sets No. 5 and No. 9, which amounted to 98 minutes. The worst operating time of the MGA was 108 minutes with input data of set No. 8. The worst packing density was 93.87% with set 9 input.

To objectively prove the effectiveness of the modified genetic algorithm with respect to 11 standard evolutionary approaches to solving the three-dimensional packaging problem, we will carry out a correlation analysis of the best and worst values of the two criteria for the algorithm's efficiency: total operating time and packing density among 11 standard methods and MGA.

Let τ_{ij} is a packing time i-th set j-th method, ρ_{ij} is a density of the resulting package of the i-th set by the j-th method, $\varphi_{ij} = \frac{\tau_{ij}}{\rho_{ij}}$ is the resultant performance indicator of the j-th method on the i-th set.

Correlation analysis showed a high level of data correlation between sets, i.e. with a change in the method for solving the three-dimensional packing problem in the i-th set, the resulting performance indicators in the remaining sets with a change in the method to the same will change proportionally. This dependence can be clearly seen in Fig. 6.

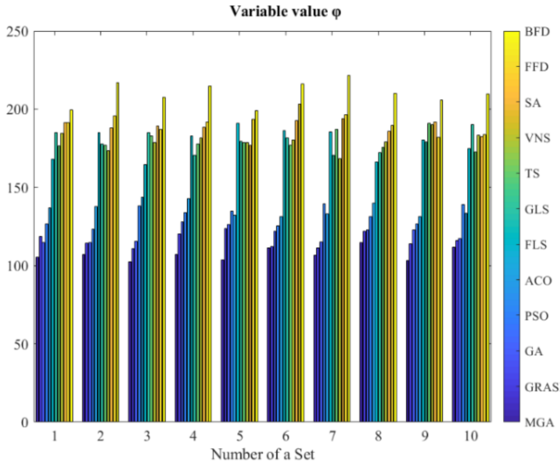

Fig. 6. Dependence of the distribution of φ_{ij} values on the packaging method and the set of boxes.

Thus, a change in the set practically does not affect the efficiency of the method for solving the three-dimensional packing problem; therefore, the effectiveness of the method can be taken as an objectively independent value.

Thus, the effectiveness of the developed modified genetic algorithm for solving the problem of three-dimensional packing of goods in containers based on a mathematical model constructed according to optimal conditions is proved by a series of simulation experiments using a software application in C #, as well as a correlation analysis of simulation results obtained as a result of experiments data.

VI. Conclusions

In this paper, we proposed one of the options for developing a new evolutionary heuristic approach for solving the three-dimensional packing problem called BPP (Bin packing problem), as applied to the variation of this problem with a single container and a set of boxes of various dimensions, called the SKP (Single knapsack problem), and The comparison of 11 basic evolutionary heuristic approaches to solving the problem of three-dimensional packing of BPP (Bin packing problem) variations SKP (Single knapsack problem) with the developed new evolutionary heuristic approach to solving BPP using modi cited genetic algorithm (MGA). By performing correlation and statistical analysis using 10 randomly created sets of input data for solving BPP, the effectiveness of MGAs was proved in comparison with 11 basic evolutionary algorithms for solving BPP. Thus, it was confirmed that MGA and similar algorithms can be effectively used to solve such logistic NP-difficult problems. The produced methodology of model analysis also has interest for using in different areas [14-24].

REFERENCES

[1] M. E. Silveira, S. M. Vieira and J.M. Da Costa Sousa, “An ACO algorithm for the 3D bin packing problem in the steel industry,” IEA/ AIE, LNCS, vol. 7906, pp. 535-544, 2013.

[2] D. Mack, A. Bortfeldt, H. Gehring, “A paralled hybrid local search algorithm for the container loading problem,” Int Trans Oper Res, vol. 11, no. 5, pp. 511-533, 2004.

[3] K.K. Lai and J.W.N.Chan, “Developing a simulated annealing algorithm for the cutting stock problem,” Computers & Industrial Engineering, vol. 32, no. 1, pp. 115-127, 1997.

[4] A. Bortfeldt and H. Gehring, “Applying tabu search to container loading problems,” Symposium on Operations Research, Jena. Operations Research Proceedings, vol. 1997, pp. 533-538, 1998. DOI: 10.1007/978-3-642-58891-4_84.

[5] O. Faroe, D. Pisinger and M. Zachariasen, “Guided local search for the three-dimensional bin packing problem,” Tech. Rep. 99-13, Department of Computer Science, University of Copenhagen, 1999.

[6] Y. Kochetov and A. Kondakov, “VNS matheuristic for a bin packing problem with color constraint,” Electron. Notes Discrete Math., vol. 58, pp. 39-46, 2017.

[7] P. Hansen and N. Mladenovic’, “Variable neighborhood search: Principles and applications,” Eur. J. Oper. Res., vol. 130, no. 3, pp. 449-467, 2001.

[8] F. Parreo, R. Alvarez-Valds, J.F. Oliveira, J.M. Tamarit, “A hybrid GRAPS/VND algorithm for two-and three-dimensional bin packing,” Annal. Oper. Res., vol. 179, no. 1, pp. 203-220, 2010.

[9] T.G. Crainic, G. Perboli and R. Tadei, “Extreme point-based heuristics for three-dimensional bin packing,” Journal on Computing, vol. 20, no. 3, pp. 368-384, 2008.

[10] D. Liu, K. Tan, S. Huang, C. Goh and W. Ho, “On solving multiobjective bin packing problems using evolutionary particle swarm optimization,”European Journal of Operational Research, vol. 190, pp. 357-382, 2008.

[11] A.V. Mokshin, V.V. Mokshin and L.M. Sharnin, “Adaptive genetic algorithms used to analyze behavior of complex system,” Communications in Nonlinear Science and Numerical Simulation, vol. 71, pp. 174-186, 2019. DOI: 10.1016/j.cnsns.2018.11.014.

[12] V.V. Mokshin, “Parallel genetic algorithm for selecting significant factors affecting the evolution of a complex system,” Bulletin of Kazan State Technical University. A.N. Tupolev, vol. 3, pp. 89-93, 2009.

[13] V.V. Mokshin, I.R. Saifudinov, L.M. Sharnin, M.V. Trusfus and P.I. Tutubalin, “Parallel genetic algorithm of feature selection for complex system analysis,” Journal of Physics: Conf. Series, vol. 1096, 012089, 2018. DOI:10.1088/1742-6596/1096/1/012089.

[14] P. Tutubalin, S. Novikova, A. Semenova, E. Komissarova, N. Arutyunova, S. Sotnikov and A. Alexandrov, “Status of creation of hardware-software complex of automatic control of the insulin delivery”, J. Phys.: Conf. Ser., vol. 1368, no. 4, 042006, 2019.

[15] P.I. Tutubalin, V.V. Mokshin, “The Evaluation of the cryptographic strength of asymmetric encryption algorithms,” Second Russia and Pacific Conference on Computer Technology and Applications (RPC), pp. 180-183, 2017. DOI: 10.1109/RPC.2017.8168094.

[16] I. Yakimov, A. Kirpichnikov, V. Mokshin, Z. Yakhina, R. Gainullin, “The comparison of structured modeling and simulation modeling of queueing systems,” Communications in Computer and Information Science (CCIS), vol. 800, 2017. DOI: 10.1007/978–3–319–68069–9 21.

[17] I.M. Yakimov, M.V. Trusfus, V.V. Mokshin and A.P. Kirpichnikov, “AnyLogic, ExtendSim and Simulink Overview Comparison of Structural and Simulation Modelling Systems,” 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC), IEEE, 2018. DOI: 10.1109/RPC.2018.8482152.

[18] P. Tutubalin, S. Novikova, A. Semenova, E. Komissarova, N. Arutyunova, S. Sotnikov and A. Alexandrov, “Status of creation of hardware-software complex of automatic control of the insulin delivery,” Journal of Physics: Conference Series, vol. 1368, no. 4, 2019.

[19] P. Tutubalin and V. Mokshin, “Structural and Functional Model of the On-board Expert Control System for a Prospective Unmanned Aerial Vehicle,” Lecture Notes in Electrical Engineering, vol. 641, pp. 262-272, 2020.

[20] P.I. Tutubalin, “Application of models and methods of stochastic matrix games for ensuring information security in mobile distributed automated control systems,” Nonlinear World, vol. 9, no. 8, pp. 535-538, 2011.

[21] V. Mokshin, A. Kirpichnikov, A. Soiko, “Simulation and optimization of the cargo terminal in the AnyLogic environment,” Journal of Physics: Conference Series, vol. 1368, no. 4, 2019.

[22] I.M. Yakimov, A.P. Kirpichnikov, V.V. Mokshin, Z.T. Yahina, “The Modelling and Optimization of Machine Management System with Computer Numerical Control,” Advances in Automation, pp. 724-731, 2019. DOI: 10.1007/978-3-030-39225-3_79.

[23] I.M. Kulikovsky, “Lower computational cost in deep learning with nearly perfect linear separability of the training set,” Computer Optics, vol. 44, no. 2, pp. 282-289, 2020. DOI: 10.18287/2412-6179-CO-645

[24] Ye.V. Goshin and A.P. Kotov, “Parallel implementation of the multi-view image segmentation algorithm using the Hough transform,” Computer Optics, vol. 41, no. 4, pp. 588-591, 2017. DOI: 10.18287/ 2412-6179-2017-41-4-588-591.