One-dimensional swarm algorithm packaging

Boris K Lebedev, Oleg B Lebedev, Ekaterina O Lebedeva

Southern Federal University, 105/42 Bolshaya Sadovaya Str., Rostov-on-Don, 344006, Russia

E-mail: lebedev.b.k@gmail.com, lebedev.ob@mail.ru, lbedevakate@mail.ru

Abstract. The paper considers an algorithm for solving the problem of one-dimensional packaging based on the adaptive behavior model of an ant colony. The key role in the development of the ant algorithm is the choice of representation (interpretation) of the solution. The structure of the solution search graph, the procedure for finding solutions on the graph, the methods of deposition and evaporation of pheromone are described. Unlike the canonical paradigm of an ant algorithm, an ant on the solution search graph generates sets of elements distributed across blocks. Experimental studies were conducted on IBM PC. Compared with the existing algorithms, the results are improved.

1. Introduction

The task of packing one-dimensional blocks is a common production task. It is solved in the production of steel, glass, paper, VLSI design, budgeting, etc. [1]. The problem of packing one-dimensional elements into blocks is NP-complete. Therefore, the main efforts of researchers are aimed at constructing effective approximate algorithms.

Despite the high degree of study of the problem, the existence of a huge number of different methods for solving it for a number of control (test) problems has not been optimized. Moreover, at the moment, there is no universal algorithm presented in the literature, capable of equally effectively solving all test problems. The disadvantages of the known algorithms used to solve the problem of packing blocks are a low degree of consideration of the features of the problem, which leads to unnecessary demands on the amount of memory, the operating time and the deterioration of the quality of the solutions obtained. In this connection, the task of packing blocks is an actual problem of combinatorial optimization, which is facing specialists in various fields of production.

The result of an ongoing search for the most effective packaging methods has been the use of bionic methods and algorithms [2-6]. One of the new directions is multi-agent methods of intellectual optimization, based on the modeling of collective intelligence [7-10]. To such methods, it is possible to carry use ant algorithms (Ant Colony Optimization - ACO) [11, 12]. The idea of an ant algorithm is the modeling of the behavior of ants, related to their ability to quickly find the shortest path from an ant hill to a food source. The basis of the behavior of the ant colony is self-organization, which ensures achievement of the common goals of the colony on the basis of low-level interaction due to which, in general, the colony is a reasonable multi-agent system. The paper describes a method for solving the one-dimensional packing problem, based on modeling the adaptive behavior of the ant colony.
The search for solutions is carried out on a full scale; the search of solutions graph is (SSG) 3.  Search for solutions on the basis of modeling the adaptive behavior of ants colony procedure.

The vertices of set $\mathcal{B} = \{a_i | i = 1, 2, ..., n\}$ are considered in an increasing order of their indices: the element in question is packed into the current block if the block does not overflow; otherwise it is packed into a new block that becomes current. The time complexity of the algorithm is $O(n)$. Obviously, there is such a sequence of elements in the list at which the solution of the packing problem will be optimal. Thus, the solution of the packaging problem is to find the list used by the standard packing procedure.

The search for solutions is carried out on a full graph of the search for solutions $G = (\mathcal{X}, \mathcal{U})$. The vertices of set $\mathcal{X} = \{x_i | i = 1, 2, ..., n\}$ correspond to elements of list $\mathcal{S}$. The solution is a route in graph $G$ that includes all vertices. To find the route in the graph, various algorithms are used, algorithms constructed on the basis of the method of the ant colony [5, 9] showed particular efficiency.

The paper proposes an ant algorithm for solving the packing problem using the standard packing procedure.

2. Formulation of the problem

The problem of one-dimensional packing can be formulated as follows. There is the given set of elements $\mathcal{A} = \{a_i | i = 1, 2, ..., n\}$. The weight of the elements is given by set $\mathcal{W} = \{w_i | i = 1, 2, ..., n\}$. It is necessary to form $\mathcal{F}$ nodes, i.e. set $\mathcal{A}$ is decomposed into $\mathcal{F}$, into non-empty and disjoint subsets of

$$\mathcal{A} = \bigcup \mathcal{A}_j, \mathcal{A}_j \neq \emptyset, (\forall i, j) [\mathcal{A}_i \cap \mathcal{A}_j = \emptyset]$$

Let us denote by $W_j$ the weight of the elements assigned to node $\mathcal{A}_j$.

The optimization criterion is the number of nodes $\mathcal{F}$. The goal of optimization is minimization of $\mathcal{F}$.

Let us consider the standard packing procedure [2]. A list $\mathcal{S}$ items for packing is given. Let us suppose that the elements for packing are sorted in some way. The first element of list $\mathcal{S}$ is packed in block $\mathcal{A}_1$. Elements 2, ..., $n$ are considered in an increasing order of their indices: the element in question is packed into the current block if the block does not overflow; otherwise it is packed into a new block that becomes current. The time complexity of the algorithm is $O(n)$. Obviously, there is such a sequence of elements in the list at which the solution of the packing problem will be optimal. Thus, the solution of the packaging problem is to find the list used by the standard packing procedure.

The search for solutions is carried out on a full graph of the search for solutions $G = (\mathcal{X}, \mathcal{U})$. The vertices of set $\mathcal{X} = \{x_i | i = 1, 2, ..., n\}$ correspond to elements of list $\mathcal{S}$. The solution is a route in graph $G$ that includes all vertices. To find the route in the graph, various algorithms are used, algorithms constructed on the basis of the method of the ant colony [5, 9] showed particular efficiency.

The paper proposes an ant algorithm for solving the packing problem using the standard packing procedure.

3. Search for solutions on the basis of modeling the adaptive behavior of ants colony

The search for solutions is carried out on a full scale; the search of solutions graph is (SSG) $G = (\mathcal{X}, \mathcal{U})$. The vertices of set $\mathcal{X} = \{x_i | i = 1, 2, ..., n\}$ correspond to elements $a_i$. In the general case, the search for the solution of the packing problem is carried out by the collective of ants $\mathcal{Z} = \{z_k | k = 1, 2, ..., n\}$. At each iteration of the ant algorithm, each ant $z_k$ builds its specific solution - the route in graph $G$ broken into parts. The vertices of each part of the route correspond to elements assigned to one block. For uniform distribution of ants and creation of equal starting conditions, the vertices of set $\mathcal{X}$ with total number $n$ are used as the initial vertices for the routes formed by the ants.

Simulation of the behavior of ants in the packing problem is associated with the distribution of the pheromone on the edges of graph $G$. At the initial stage, the same (small) amount of pheromone $\Phi / \nu$ is deposited on all edges of graph $G$, where $\nu = |\mathcal{U}|$. Parameter $\Phi$ is given a priori. The process of finding solutions is iterative. Each iteration $l$ consists of three steps. At the first stage, the ant finds a solution; at the second stage there is the pheromone; in the third stage the pheromone is evaporated. The work uses the ant-cycle method of ant systems. In this case, the pheromone is deposited by the agent on the edges after the complete formation of the solution. The process of finding solutions is iterative. At the first stage of each iteration, each ant $z_k$ forms its own route $M_k$, splitting it into parts. Each part of route $M_k$, $M_{kj} \in M_k$, includes vertices corresponding to the elements placed in block $A_j$. The process of constructing route $M_k$ is incremental. At each step $t$, agent $z_k$ applies the probability rule for selecting the next vertex to include its generated route $M_k(t)$. Let step $t$ form part $M_{kj} \in M_k$, which is the tail of $M_k(t)$. To do this, let us select a set of vertices $X_{kj}(t) \in X$ such that if $x_i \in X_{kj}(t)$, then element $s(t) \in S_k$, considered at step $t$, corresponding to vertex $x_i$, can be fixed in block $A_j$, without overflow, those without exceeding the weight of the elements fixed in it.

Let $e_k(t)$ be the set of vertices of part $M_{kj}(t)$ of route $M_k(t)$. The agent looks through all vertices $x_i \in X_{kj}(t)$ and calculates the $f_{ikj}$ parameter for each vertex - the total level of pheromone on the edges of graph $G$ that connect $x_i$ to the vertices of set $e_k(t)$. Probability $P_{ikj}$ of the inclusion of vertex $x_i \in X_{kj}(t)$ in part $M_{kj}(t)$ of route $M_k(t)$ is determined by the following relation:
Agent $z_k$ with probability $P_{ik}$ chooses one of vertices $x_i \in X_k(t)$, which is included at the end of route $M_k(t)$. After this, the placement of element $x_i \in X_k(t)$ in block $A_j$ is fixed. If $X_k(t)$ is empty, then a vertex that is not included in the route is selected and the corresponding element is placed in the next block $A_{j+1}$.

In the second iteration step, each ant $z_k$ lays the pheromone on the edges of the complete subgraphs, each of which is constructed on the set of vertices that make up one of parts $M_{kj}(t)$ of route $M_k(t)$. The amount of pheromone $\Delta_k(l)$, deposited by ant $z_k$ on each edge of the subgraphs built on the $l$-th iteration is determined as follows:

$$\Delta_k(l) = \frac{\Phi}{F_k(l)}$$  \hspace{1cm} (2)

where $l$ is the iteration number; $\Phi$ is the total amount of the pheromone deposited by the ant on the edges of full subgraphs $G_{kj}$; $F_k(l)$ is the target function for the solution obtained by ant $z_k$ at the $l$-th iteration. The smaller $F_k(l)$, the more pheromone is deposited on the edges of the constructed route and, consequently, the greater the probability of selecting these edges when constructing routes at the next iteration.

After each agent has formed a solution and postponed the pheromone, at the third stage, the total evaporation of the pheromone occurs on the edges of complete graph $G$ in accordance with formula (3):

$$f_{ik} = f_{ik}(1 - \rho).$$  \hspace{1cm} (3)

where $\rho$ is the refresh rate.

After performing all the actions on the iteration, there is an agent with the best solution that is remembered. Then the next iteration is performed.

The time complexity of this algorithm depends on the lifetime of colony $l$ (the number of iterations), the number of vertices of graph $n$, and the number of ants $m$, and is defined as $O(ln^2m)$.

The algorithm of one-dimensional packaging based on the method of the ant colony is formulated as follows.

1. In accordance with the initial data, there is a complete the search for solutions graph $G = (X, U)$.
2. The number of agents and the vertices into which they are placed is determined.
3. The value of parameter $\Phi$ is given and the number of iterations is $N_l$.
4. On all edges of graph $G$, the initial amount of the pheromone is deposited. $l = 1$.
5. At the first stage of the $l$-th iteration on the SSG, every agent $z_k$ is located on route $M_k(l)$ and there is the corresponding solution of the packing problem.
6. For each solution of the packing problem, the value of criterion $F_k(l)$is found.
7. In graph $G$ on the edges of complete subgraphs $G_{kj}$, each of which is constructed on the set of vertices included in one of parts $M_{kl}(t)$of route $M_k(l)$, the pheromone is deposited. The amount of the pheromone postponed by each agent is proportional to $F_k(l)$.
8. The procedure for evaporation of pheromone on the edges of graph $G$.
9. The best solution obtained during all performed iterations is selected.
10. If all iterations are fulfilled, then there is the end of the algorithm's work; otherwise, go to step 5 to perform the next iteration.

4. Experimental research

The purpose of the program research was to determine the quality of the solution, which is determined by the difference (deviation) between the number of blocks in the known best solution (or global optimum) and the number of blocks in the solution achieved by ant algorithm. To carry out objective experiments, the authors used the control tasks of class U available in the library of OR objects.
(http://www.ms.ic.ac.uk/info.html), on which the developed program was tested. The class of problems consists of elements whose weights are uniformly distributed on the interval of (20, 100). For all test problems of class U, optimal solutions were obtained using an exact algorithm developed by Valerio de Carvalho [13]. This algorithm is built on the method of branches and boundaries and has considerable laboriousness.

| No | Task  | GO | APA | Δ | Task  | GO | APA | Δ |
|----|-------|----|-----|---|-------|----|-----|---|
| 1  | U120-1| 48 | 48  | 0 | U250-1| 99 | 99  | 0 |
| 2  | U120-2| 49 | 49  | 0 | U250-2| 100| 100 | 0 |
| 3  | U120-3| 46 | 46  | 0 | U250-3| 102| 102 | 0 |
| 4  | U120-4| 49 | 49  | 0 | U250-4| 100| 100 | 0 |
| 5  | U120-5| 50 | 50  | 0 | U250-5| 101| 101 | 0 |
| 6  | U120-6| 48 | 48  | 0 | U250-6| 101| 101 | 0 |
| 7  | U120-7| 48 | 48  | 0 | U250-7| 102| 102 | 0 |
| 8  | U120-8| 49 | 49  | 0 | U250-8| 103| 103 | 0 |
| 9  | U120-9| 50 | 50  | 0 | U250-9| 105| 105 | 0 |
| 10 | U120-10| 46| 46  | 0 | U250-10| 101| 101 | 0 |
| 11 | U120-11| 52| 52  | 0 | U250-11| 105| 105 | 0 |
| 12 | U120-12| 49| 49  | 0 | U250-12| 101| 101 | 0 |
| 13 | U120-13| 48| 48  | 0 | U250-13| 105| 105 | 0 |
| 14 | U120-14| 49| 49  | 0 | U250-14| 102| 102 | 0 |
| 15 | U120-15| 50| 50  | 0 | U250-15| 100| 100 | 0 |
| 16 | U120-16| 48| 48  | 0 | U250-16| 105| 105 | 0 |
| 17 | U120-17| 52| 52  | 0 | U250-17| 97 | 97  | 0 |
| 18 | U120-18| 52| 52  | 0 | U250-18| 100| 100 | 0 |
| 19 | U120-19| 49| 49  | 0 | U250-19| 100| 100 | 0 |
| 20 | U120-20| 49| 49  | 0 | U250-20| 102| 102 | 0 |

To solve each of the test problems, the developed program was launched 10 times. Each launch was independent. Table 1 in the “Task” column shows the code name of the task, "GO" - the number of blocks in the globally optimal solution, "APA" - the number of blocks in the best solution obtained with the ant packaging algorithm, "Δ" - deviation obtained solution from a globally optimal solution.

As can be seen from Table 1, the use of the developed program allowed one to obtain optimal solutions for all set tasks. It should be noted that the time taken to solve each of the test tasks listed in the tables above did not exceed 20 seconds.

Studies have shown that the time complexity of the algorithm considered has an estimate of \(O(n^2)\), where \(n\) is the number of elements

5. Conclusion

New mechanisms for solving the one-dimensional packaging problem using mathematical methods, in which the principles of natural decision-making mechanisms are laid, are proposed. In contrast to the canonical paradigm ant on finding solutions, graph \(G = (X, U)\) is built on the route to the partition of the formation and on the peaks included in each part, subgraphs, on the edges of which the pheromone is delayed. Experimental studies were conducted on IBM PC. The temporal complexity of the algorithm (TCA), obtained experimentally, practically coincides with the theoretical studies and, for the considered test problems, is \((TCA \approx O(n^2))\).
To conduct objective experiments, the authors used well-known test problems presented in the literature and the Internet. The tasks, on which the developed algorithm was tested, are available in the OR-objects library (http://www.ms.ic.ac.uk/info.html). To draw up reliable conclusions, not one, but a series of experiments-experiments was carried out. The developed algorithm allowed one to obtain optimal solutions for all the set tasks.

6. Acknowledgments
This research is supported by grants of the Russian Foundation for Basic Research of the Russian Federation, project № 18-07-00737.

References
[1] Bischoff E E and Wäscher G 1995 Cutting and packing (European Journal of Operational Research) pp 503–505
[2] Potarousov R V and Kurciechik VM 2006 The problem of one-dimensional packing of elements № 8 (Publishing House of TRTU) pp 88–93
[3] Ross P and Marin-Blazquez and Schulenburg S and Hart E 2003 Learning a Procedure That Can Solve Hard Bin-Packing Problems: A New GA-Based Approach to Hyper-heuristics (Proceeding of the Genetic and Evolutionary Computation Conference, GECCO: Chicago Illinois USA) pp 1295–1306
[4] Gupta J N and Ho J C 1999 New Heuristic Algorithm for the One-dimensional Bin-packing Problem vol 10 (Production Planning & Control) pp 598–603
[5] Levine J and Ducatelle F 2003 Ant Colony Optimization and Local Search for Bin Packing and Cutting Stock Problems (Center for Intelligent Systems and their Applications, University of Edinburgh) pp 589–601
[6] Fleszar K and Hindi K S 2002 New Heuristics for One-dimensional Bin-packing (Computers & Operations Research) pp 821–839
[7] Lebedev O B 2013 Models of adaptive behavior of ants colony in design problems (Taganrog: Publishing house of SFEDU) p 199
[8] Dorigo M and Stützle T 2004 Ant Colony Optimization (MIT Press Cambridge MA) p 345
[9] Kureychik V M and Lebedev B K and Lebedev O B 2006 Search engine adaptation: theory and practice (Moscow: Fizmatlit) p 245
[10] Engelbrecht A P 2005 Fundamentals of Computational Swarm Intelligence (John Wiley & Sons Chichester UK) p 478
[11] Lebedev B K and Lebedev O B 2016 Ant partition algorithms using representations of a problem other than the canonical one (Rostov-on-Don: Publishing House Rostov State Transport University) 3(63) 42–47
[12] Alvim Adriana C F and Glover Fred S and Ribeiro Celso C and Aloise Dario J 2004 Local search for the bin-packing problem (Journal of Heuristics) pp 205–229
[13] Valerio de Carvalho JM 1999 Exact Solution of Bin-packing Problems Using Column Generation and Branch-and-bound (Annals of Operations Research) pp 629–659