

## БИОИНСПИРИРОВАННЫЙ АЛГОРИТМ ОДНОМЕРНОЙ УПАКОВКИ

*Б.К. Лебедев, О.Б. Лебедев, О.В. Вепринцева*  
(г. Таганрог, Южный федеральный университет)  
*lebedev.b.k@gmail.ru, lebedev.ob@mail.ru, veprinceva.ov@mail.ru*

## BIOINSPIRATED ALGORITHM OF ONE-DIMENSIONAL PACKING

*B.K. Lebedev, O.B. Lebedev, O.V. Veprinceva*  
(Taganrog, Southern Federal University)

**Abstract.** New mechanisms for solving the problem of one-dimensional packing are proposed, using mathematical methods in which the principles of natural decision-making mechanisms are laid. Unlike the canonical paradigm, an ant on the solution search graph forms subgraphs on the edges of which pheromone is deposited. To enhance convergence, a two-level deposition of pheromone was used.

**Keywords:** One-dimensional packing; swarm intelligence; optimization; adaptive behavior, ant colony.

**Введение.** Задача упаковки одномерных блоков является распространенной производственной задачей. Она решается при производстве стали, стекла, бумаги, дизайне СБИС, составлении бюджета и т.д [1]. Рассматриваемая задача упаковки одномерных элементов в блоки является *NP*-полной. Поэтому основные усилия исследователей направлены на построение эффективных приближенных алгоритмов.

Несмотря на высокую изученность задачи, существование огромного количества различных методов ее решения для ряда контрольных (тестовых) задач оптимальное решение не получено. Более того, на данный момент не существует представленного в литературе универсального алгоритма, способного одинаково эффективно решать все тестовые задачи. Недостатки известных алгоритмов, применяемых для решения задачи упаковки блоков, заключаются в низкой степени учета особенностей задачи, что приводит к излишним требованиям по объему памяти, времени работы и к ухудшению качества получаемых решений. Ввиду вышеизложенного, задача упаковки блоков является актуальной проблемой комбинаторной оптимизации, стоящей перед специалистами в различных областях производства.

Результатом непрекращающегося поиска наиболее эффективных методов упаковки стало использование бионических методов и алгоритмов [2-6]. Одним из новых направлений таких методов являются мультиагентные методы интеллектуальной оптимизации, базирующиеся на моделировании коллективного интеллекта [7-10]. К таким методам можно отнести, и муравьиные алгоритмы (Ant Colony Optimization - ACO) [11,12]. Основу поведения муравьиной колонии составляет самоорганизация, обеспечивающая достижения общих целей колонии на основе низкоуровневого взаимодействия благодаря которому, в целом, колония представляет собой разумную многоагентную систему.

В работе излагается метод решения задачи упаковки, основанный на моделировании адаптивного поведения муравьиной колонии.

**Постановка задачи.** Проблема одномерной упаковки может быть сформулирована следующим образом. Дано множество элементов  $A = \{a_i | i = 1, 2, \dots, n\}$ . Вес элементов задается множеством  $W = \{w_i | i = 1, 2, \dots, n\}$ . Необходимо сформировать  $F$  узлов, т.е. множество  $A$  разбить на  $F$  непустых и непересекающихся подмножеств  $A_j$ .  $A = \cup A_j$ ,  $A_j \neq \emptyset$ ,  $(\forall i, j) [A_i \cap A_j = \emptyset]$ .

Обозначим через  $W_j$  вес элементов, назначенных в узел  $A_j$ .  $W_j = \sum w_i$ , для  $\{i | a_i \in A_j\}$ .

Задается максимально допустимый суммарный вес –  $C$  элементов, назначенных в каждый узел:  $W_j \leq C$ .

Критерий оптимизации – число узлов  $F$ . Цель оптимизации – минимизация  $F$ .

Рассмотрим стандартную процедуру упаковки [2]. Дан список  $S$  элементов для упаковки. Предположим, что элементы для упаковки отсортированы некоторым образом. Первый элемент списка  $S$  упаковывается в блок  $A_1$ . Элементы 2, ...,  $n$  рассматриваются в порядке

возрастания их индексов: рассматриваемый элемент упаковывается в текущий блок, если не происходит переполнения блока; в противном случае он упаковывается в новый блок, который становится текущим. Временная сложность алгоритма  $O(n)$ . Очевидно, что существует такая последовательность элементов в списке, при которой решение задачи упаковки будет оптимальным. Общее число списков равно  $n!$ . Таким образом, решение задачи упаковки сводится к нахождению списка, используемого стандартной процедурой упаковки. Для нахождения оптимального списка используются различные алгоритмы, особую эффективность показали алгоритмы, построенные на основе метода муравьиной колонии [5,11,12]. Идея муравьиного алгоритма моделирование поведения муравьёв, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи. Поиск решений осуществляется на полном графе поиска решений  $G=(X,U)$ . Вершины множества  $X=\{x_i/i=1,2,\dots,n\}$  соответствуют элементами списка  $S$ . Решением является маршрут в графе  $G$ , включающий все вершины.

В работе предлагается модифицированный муравьиный алгоритм решения задачи упаковки с использованием последовательной процедуры упаковки.

**Поиск решений на основе моделирования адаптивного поведения муравьиной колонии.** Пусть имеется некоторое решение задачи упаковки  $n$  элементов в  $F$  узлов. Пусть  $n_i$  – число элементов, уложенных в  $i$ -ый блок. Поскольку порядок, в котором элементы множества  $A_i$  вошли в  $i$ -ый блок не имеет значения, то решению будет соответствовать число списков равно  $n_1! \cdot n_2! \cdot \dots \cdot n_i! \cdot \dots \cdot n_F!$ . В связи с этим при поиске решения актуальна проблема синтеза и просмотра только одного списка, соответствующего конкретному решению.

Для уменьшения общей трудоемкости поисковой процедуры предлагается подход, основанный на декомпозиции структуры данных, используемой в процессах построения маршрута и отложения феромона. Отложение феромона осуществляется на полном графе  $G=(X,U)$ , а построение маршрута будем осуществлять на ориентированном графе  $B(X,V)$ , где  $X$  – множество вершин, соответствующих элементам множества  $A$ ,  $|X| = n$ ,  $V$  – множество ориентированных ребер. Из каждой вершины  $x_i \in X$  выходит  $n-i$  ребер, которые входят в вершины с номерами от  $i+1$  до  $n$ .

Процесс формирования маршрута связан с процессом заполнения блоков элементами. В результате муравей строит маршрут, разбитый на части. Каждой части соответствует множество элементов размещенных с соблюдением ограничений  $W_j \leq C$  в одном блоке.

Формирование маршрута производится путем циклического просмотра вершин графа  $B(X,V)$ . На каждом цикле формируется очередной блок (очередная часть маршрута).

При этом в любом маршруте построенным любым муравьем, содержащем некоторую фиксированную часть  $X_j \neq \emptyset$ , элементы следуют в одном порядке – в порядке возрастания индексов. Таким образом, конкретному допустимому разбиению множества элементов по блокам соответствовать один список.

Расположим вершины графа  $B(X,V)$  в линейку, так чтобы все ребра были направлены слева направо (рис .1).

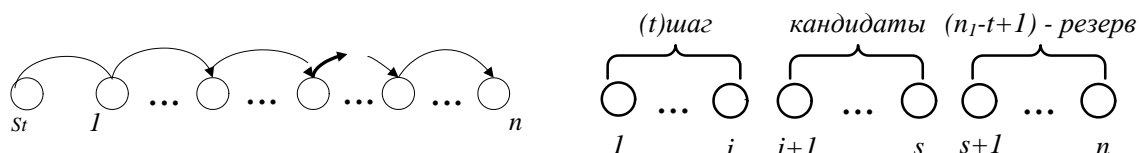


Рис. 1 – Представление графа  $B(X,V)$  в линейку

Введем стартовую вершину  $St$ , связанную со всеми вершинами графа  $B(X,V)$ . Вершина  $St$  используется для начального размещения агентов и не входит в состав формируемого

подмножества. Назовем граф  $B(X, V)$  с расположенными в линейку (упорядоченными) вершинами шаблоном графа  $B(X, V)$ .

Пусть на некотором цикле формируется очередной блок  $A_j$ . Необходимо разбить множество еще нераспределенных вершин  $X$  на два непустых и непересекающихся подмножества  $X_j$  и  $X_o$ ,  $X_j \cup X_o = X$ ,  $X_j \cap X_o = \emptyset$ ,  $X_j \neq \emptyset$ . На формируемый блок накладываются ограничения:  $W_j \leq C$ . Пусть  $|X_j| = n_j$ ,  $|X_o| = n_o$ ,  $n_j + n_o = n$ .

Сформировать узел  $X_j$ , содержащий  $n_j$  вершин, это значит построить в графе  $B(X, V)$  ориентированный маршрут, содержащий  $n_j$  вершин, с соблюдением ограничения:  $W_j \leq C$ . Процесс построения ориентированного маршрута пошаговый. На первом шаге в маршрут могут быть включены вершины с номерами от 1 до  $n - (n_j - 1)$ . При этом суммарный вес  $(n_j - 1)$  последних в линейке вершин должен быть больше остаточной вместимости формируемого блока.  $(n_j - 1)$  последних в линейке вершин резервируются для того, чтобы можно было «до упора» заполнять формируемый блок. Пусть выполнено  $(l - 1)$  шагов, при этом последней вершиной маршрута, является вершина с номером  $i$ . Кандидатами на включение в маршрут являются вершины с номерами от  $(i + 1)$  до  $s$ . При этом суммарный вес последних вершин с номерами от  $s + 1$  до  $n$  должен быть больше остаточной вместимости формируемого блока. Эти вершины резервируются для того, чтобы можно было построить ориентированный маршрут, «до упора» заполняющий формируемый блок.

После построения маршрута, все вершины маршрута образуют множество  $X_j$ , а все вершины не входящие в маршрут образуют множество  $X_o$ .

Задача распределения элементов по блокам заключается в циклическом построении  $F$  ориентированных подмаршрутов на графе  $B(X, V)$ . Каждый  $j$ -й подмаршрут маршрута образует множество  $X_j$ . Каждый муравей  $z_k$  использует свой граф  $B_k(X, V)$ , изменяющийся после построения очередного подмаршрута, но все муравьи используют для отложения феромона и анализа альтернатив один общий граф  $G = (X, U)$ .

На первом шаге каждым муравьем  $z_k$  на своем графе  $B^1_k(X, V)$  строится маршрут  $M^1_k$  и формируется множество  $X^1_k$ . На втором шаге предварительно модифицируется граф  $B^1_k$  путем удаления из него всех вершин, входящих в множество  $X^1_k$ , а также всех ребер, смежных этим вершинам. Образуется граф  $B^2_k$ , на котором муравей строит маршрут  $M^2_k$ .

На шаге  $j + 1$  формируется граф  $B^{j+1}_k$  путем удаления из  $B^j_k$  всех вершин, входящих в сформированные множества  $X^j_k - X^i_k$ , а также всех ребер, смежных этим вершинам, на котором муравей строит маршрут  $M^{j+1}_k$ .

После формирования каждым муравьем  $F$  подмножеств  $X^j_k - X^F_k$  рассчитываются оценки решений, в соответствии с которыми на графе  $G(X, U)$  откладывается феромон.

Моделирование поведения муравьев в задаче упаковки связано с распределением феромона на ребрах графа  $G$ . На начальном этапе на всех ребрах графа  $G$  откладывается одинаковое (небольшое) количество феромона  $\Phi/v$ , где  $v = |U|$ . Параметр  $\Phi$  задается априори. Процесс поиска решений итерационный. Каждая итерация  $l$  включает три этапа. На первом этапе муравей находит решение, на втором этапе откладывает феромон, на третьем этапе осуществляется испарение феромона. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромон откладывается агентом на ребрах после полного формирования решения. На первом этапе каждой итерации каждый  $k$ -й муравей формирует свой собственный маршрут  $M_k$ . Процесс поиска решений итерационный. На первом этапе каждой итерации каждый  $k$ -й муравей формирует свой собственный маршрут  $M_k$  с разбиением его на части. Каждая часть  $M^i_k \in M_k$  включает вершины, соответствующие элементам, помещаемым в блок  $A_j$ . Процесс построения маршрута  $M_k$  пошаговый. На каждом шаге  $t$  агент  $z_k$  применяет вероятностное правило выбора на графе  $B_k$  следующей вершины для включения ее формируемый маршрут  $M_k(t)$ . Пусть на шаге  $t$  формируется часть  $M^j_k(t)$ , являющаяся хвостом  $M_k(t)$ . Для этого в графе  $B^j_k$  выделяется множество вершин  $X_k(t) \in X$ , таких, что если  $x_i \in X_k(t)$ , то с одной стороны соответствующий этой вершине элемент может быть зафиксиро-

ван в блоке  $A_j$ , без переполнения, т.е. без превышения ограничения на суммарный вес размещенных в нем элементов. А с другой стороны суммарный вес вершин в графе  $B_k^j$  с номерами от  $i+1$  до  $n$  больше остаточной вместимости блока  $A_j$ .

Пусть  $e_k(t)$  - множество вершин части  $M_k^j(t)$  маршрута  $M_k(t)$ . Агент просматривает все вершины  $x_i \in X_k(t)$  и для каждой вершины рассчитывается параметр  $f_{ik}$  - суммарный уровень феромона на ребрах графа  $G$ , связывающих  $x_i$  с вершинами множества  $e_k(t)$ . Вероятность  $P_{ik}$  включения вершины  $x_i \in X_k(t)$  в часть  $M_k^j(t)$  маршрута  $M_k(t)$  определяется следующим соотношением:

$$P_{ik} = f_{ik} / \sum_i f_{ik} . \quad (1)$$

Агент  $z_k$  с вероятностью  $P_{ik}$  выбирает одну из вершин  $x_i \in X_k(t)$ , которая включается в конец маршрута  $M_k(t)$ . После этого фиксируется размещение элемента  $x_i \in X_k(t)$  в блоке  $A_j$ . Если  $X_k(t)$  оказывается пустым, то осуществляется переход к формированию блока  $A_{j+1}$ .

На втором этапе итерации, каждый муравей  $z_k$  откладывает феромон на ребрах полных подграфов  $G_k^j$  графа  $G$ . Каждый  $G_k^j$  построен на множестве вершин, входящих в состав одной из частей  $M_k^j(t)$  маршрута  $M_k(t)$ . Количество феромона  $\Delta\tau_k(l)$ , откладываемое муравьем  $z_k$  на каждом ребре подграфов графа  $G$ , построенных на  $l$ -ой итерации, определяется следующим образом:

$$\Delta\tau_k(l) = \Phi / F_k(l), \quad (2)$$

где  $l$ -номер итерации,  $\Phi$  - общее количество феромона, откладываемое муравьем на ребрах полных подграфов  $G_k^j$ ,  $F_k(l)$  - целевая функция для решения, полученного муравьем  $z_k$  на  $l$ -ой итерации. Чем меньше  $F_k(l)$ , тем больше феромона откладывается на ребрах подграфов, и, следовательно, тем больше вероятность выбора этих ребер при построении маршрутов на следующей итерации.

При втором подходе отложение феромона на ребрах графа осуществляется дважды. Вначале, также как и при первом подходе, каждый муравей откладывает феромон на ребрах полного подграфа  $G_k^j$ , построенного на вершинах множества  $X_k^j$ , а затем откладывается дополнительное количество феромона на всех ребрах построенного муравьем ориентированного маршрута, включающего вершины множества  $X_k^j$ . Таким образом, используется двухуровневое отложение феромона.

После того, как каждый агент сформировал решение и отложил феромон, на третьем этапе происходит общее испарение феромона на ребрах полного графа  $G$  в соответствии с формулой (3).

$$f_{ik} = f_{ik}(1 - \rho), \quad (3)$$

где  $f_{ik}$  - общее количество феромона, отложенного на ребре  $u^i$  графа  $G$ ,  $\rho$  - коэффициент обновления.

После выполнения всех действий на итерации находится агент с лучшим решением, которое запоминается. Далее осуществляется переход на следующую итерацию.

Временная сложность этого алгоритма зависит от времени жизни колонии  $l$  (число итераций), количества вершин графа  $n$  и числа муравьев  $m$ .

Алгоритм одномерной упаковки на основе метода муравьиной колонии формулируется следующим образом.

1. В соответствии с исходными данными формируется полный граф  $G=(X,U)$  для отложения феромона и ориентированный граф  $B(X,V)$  для построения маршрута.
2. Определяются число агентов и вершины, в которые они помещаются.
3. Задается значение параметра  $\Phi$  и число итераций -  $N_l$ .
4. На всех ребрах графа  $G$  откладывается начальное количество феромона.  $l=1$ .
5. На первом этапе  $l$ -й итерации на ГПР  $B_k$  базирываясь на данных графа  $G$  каждым агентом  $z_k$ , находится маршрут  $M_k(l)$  и соответствующее ему решение задачи упаковки.
6. Для каждого решения задачи упаковки находится значение целевой функции  $F_k(l)$ .

7. В графе  $G$  на рёбрах полных подграфов  $G_k^j$ , каждый из которых построен на множестве вершин, входящих в состав одной из частей  $M_k^j(t)$  маршрута  $M_k(l)$ , откладывается феромон. Количество откладываемого каждым агентом феромона пропорционально  $F_k(l)$ .

8. Выполняется процедура испарения феромона на ребрах графа  $G$ .

9. Выбирается лучшее решение, полученное на протяжении всех выполненных итераций.

10. Если все итерации выполнены, то конец работы алгоритма, в противном случае, переход к пункту 5 для выполнения очередной итерации.

**Экспериментальные исследования.** Целью исследования программы было определение качества решения, которое определяется величиной разницы (отклонением) между количеством блоков в известном лучшем решении (либо глобальном оптимуме) и количеством блоков в решении, достигнутом МА. Для проведения объективных экспериментов были использованы доступные в библиотеке OR-объектов (<http://www.ms.ic.ac.uk/info.html>) контрольные задачи класса U, на которых был протестирована разработанная программа. Класс задач состоит из элементов, веса которых равномерно распределены на интервале (20, 100). Для всех тестовых задач класса U оптимальные решения были получены при помощи точного алгоритма, разработанного Valerio de Carvalho [13]. Данный алгоритм построен по методу ветвей и границ и обладает значительной трудоемкостью.

Для решения каждой из тестовых задач разработанная программа запускалась 10 раз. Каждый запуск являлся независимым. В таблице 1 в столбце «Задача» приведено кодовое название контрольной (тестовой) задачи, «GO» - количество блоков в глобально оптимальном решении, «МА» - количество блоков в лучшем решении, полученном при помощи муравьиного алгоритма упаковки, «О» - отклонение полученного решения от глобально оптимального решения.

Таблица 1 - Результаты, полученные на контрольных задачах наборов U120 и U250

Задача	GO	МА	О	Задача	GO	МА	О	Задача	GO	МА	О	Задача	GO	МА	О
U120-1	48	48	0	U120-11	52	52	0	U250-1	99	99	0	U250-11	105	105	0
U120-2	49	49	0	U120-12	49	49	0	U250-2	100	100	0	U250-12	101	101	0
U120-3	46	46	0	U120-13	48	48	0	U250-3	102	102	0	U250-13	105	105	0
U120-4	49	49	0	U120-14	49	49	0	U250-4	100	100	0	U250-14	102	102	0
U120-5	50	50	0	U120-15	50	50	0	U250-5	101	101	0	U250-15	100	100	0
U120-6	48	48	0	U120-16	48	48	0	U250-6	101	101	0	U250-16	105	105	0
U120-7	48	48	0	U120-17	52	52	0	U250-7	102	102	0	U250-17	97	97	0
U120-8	49	49	0	U120-18	52	52	0	U250-8	103	103	0	U250-18	100	100	0
U120-9	50	50	0	U120-19	49	49	0	U250-9	105	105	0	U250-19	100	100	0
U120-10	46	46	0	U120-20	49	49	0	U250-10	101	101	0	U250-20	102	102	0

Как видно из таблицы 1, использование разработанной программы позволило получить оптимальные решения для всех задач набора. Следует отметить, что время, затраченное на решение каждой из тестовых задач, указанных в таблицах выше, не превышало 20 секунд.

Исследования показали, что временная сложность рассмотренного алгоритма имеет оценку  $O(n^2)$ , где  $n$  – число элементов, а максимальная эффективность адаптивного поиска обеспечивается при значениях управляющих параметров:  $\zeta$ ,  $\delta$ ,  $\rho$ , приведенные в таблице:

$\zeta$  – априорно задаваемое количество феромона;

$\delta$  – общее количество феромона, откладываемое  $k$ -м муравьем;

$\rho$  – коэффициент обновления.

**Заключение.** Предложены новые механизмы решения задачи упаковки, использующие математические методы, в которых заложены принципы природных механизмов принятия решений. В отличие от канонической парадигмы муравьиного алгоритма произведена декомпозиция структуры данных, используемой в процессах построения маршрута (формирование подграфа) и отложения феромона. Муравьем строится маршрут с разбиением на ча-

сти на графе поиска решений  $B(X, V)$  с последующим формированием на вершинах графа  $G=(X, U)$ , входящих в каждую часть, подграфов, на ребрах которых откладывается феромон. Это позволило снизить комбинаторную сложность задачи. Для повышения сходимости использовано двухуровневое отложение феромона.

Экспериментальные исследования проводились на IBM PC. Временная сложность алгоритма (BCA), полученная экспериментальным путем, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет ( $BCA \approx O(n^2)$ ).

Для проведения объективных экспериментов были использованы известные тестовые задачи, представленные в литературе и Интернет. Задачи, на которых был протестирован разработанный алгоритм, доступны в библиотеке OR-объектов (<http://www.ms.ic.ac.uk/info.html>). Для составления достоверных выводов был проведен не один, а серия опытов-экспериментов.

***Работа выполнена при финансовой поддержке гранта РФФИ № 18-07-00737.***

## ЛИТЕРАТУРА

1. Bischoff E. E. and Wäscher G. Cutting and packing. European Journal of Operational Research, 84, 1995. -pp. 503–505.
2. Потарусов Р.В., Курейчик В.М. Проблема одномерной упаковки элементов // Известия ТРТУ. Изд-во ТРТУ, 2006. №8. - С. 88 – 93.
3. Ross P., Marin-Blazquez J.G., Schulenburg, S., and Hart E. 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 2003, Chicargo, Illinois, USA, 2003, 1295-1306.
4. Gupta J. N. and Ho J. C. A New Heuristic Algorithm for the One-dimensional Bin-packing Problem, Production Planning & Control. 10, 1999, -pp. 598-603.
5. Levine J. and F. Ducatelle. Ant Colony Optimization and Local Search for Bin Packing and Cutting Stock Problems. Centre for Intelligent Systems and their Applications, School of Informatics, University of Edinburgh, 2003. –pp. 834-851.
6. Fleszar, K. and Hindi, K. S. New Heuristics for One-dimensional Bin-packing, Computers & Operations Research. 29, 2002, -pp. 821-839.
7. Лебедев О.Б. Модели адаптивного поведения муравьиной колонии в задачах проектирования. - Таганрог: Изд-во ЮФУ, 2013. – 199 с.
8. M. Dorigo and T. Stützle. Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.
9. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Поисковая адаптация: теория и практика. М.: Физматлит, 2006. – 269 с.
10. Engelbrecht A. P. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, Chichester, UK, 2005. –pp. 368-383.
11. Лебедев Б.К., Лебедев О.Б. Муравьиные алгоритмы разбиения, использующие представления задачи, отличные от канонического // Вестник Ростовского государственного университета путей сообщения – Ростов-на-Дону: Изд-во РГУПС, 2016, №3 (63). – С. 42-47.
12. Alvim Adriana C. F., Glover Fred S., Ribeiro Celso C., and Aloise Dario J. Local search for the bin packing problem // Journal of Heuristics, 10, 2004, -pp.205–229.
13. Valerio de Carvalho, J. M. Exact Solution of Bin-packing Problems Using Column Generation and Branch-and-bound //Annals of Operations Research. 86, 1999, -pp. 629-659.