

Курейчик Владимир Викторович – Южный федеральный университет; e-mail: vkur@sfedu.ru; г. Таганрог, Россия; тел.: 88634371651; кафедра систем автоматизированного проектирования им. В.М. Курейчика; зав. кафедрой; д.т.н.; профессор.

Герасименко Петр Сергеевич – e-mail: pege@sfedu.ru; тел.: 88634371651; кафедра систем автоматизированного проектирования им. В.М. Курейчика; аспирант.

Kureichik Vladimir Victorovich – Southern Federal University; e-mail: vkur@sfedu.ru; Taganrog, Russia; phone: +78634371651; the department of Computer Aided Design; head of department; dr. of eng. sc.; professor.

Gerasimenko Petr Sergeevich – e-mail: pege@sfedu.ru; phone: +78634371651; the department of Computer Aided Design; post graduate student.

УДК 004.896

DOI 10.18522/2311-3103-2024-4-14-30

Б.К. Лебедев, О.Б. Лебедев, М.А. Ганжур

БИОИНСПИРИРОВАННЫЙ АЛГОРИТМ ПЛОТНОЙ УПАКОВКИ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ РАСКРОЯ ПОЛУОГРАНИЧЕННОЙ ПОЛОСЫ

Предлагается архитектура и методология раскроя-упаковки полуограниченной полосы на основе методов биоинспирированного поиска. В основе подхода к декомпозиции общей задачи упаковки и методологии формированию карт раскроя лежат эвристики уровневого подхода к упаковке полосы. Архитектура сформирована на основе декомпозиции общей задачи и включает 5 основных секций: управление процессом поиска; формирования блоков; формирование контейнеров; компакция контейнеров; заполнения полосы контейнерами. Упаковка ориентирована на двухуровневый раскрой полосы. На первом уровне путем гильотинного разреза выполняется раскрой на контейнеры. На втором уровне два варианта раскроя: путем гильотинного или путем не гильотинного разреза выполняется раскрой контейнеров на детали (элементы прямоугольной формы). Упаковка выполняется путем последовательного заполнения уровней полосы контейнерами. В основу методологии раскроя-упаковки полуограниченной полосы положен иерархический подход снизу вверх. Задача, решаемая на первом уровне иерархии, заключается в формировании множества блоков В одинаковой ширины на базе исходного набора А прямоугольников, включаемых в блоки. Для решения поставленной задачи авторами разработан биоинспирированный алгоритм одномерной упаковки элементов в одинаковые блоки. На втором уровне иерархии решается задача распределения блоков по контейнерам. Все контейнеры и блоки имеют одинаковую ширину D, равную ширине полосы. В каждом контейнере помещаются два блока. Задача распределения блоков по контейнерам сведена к задаче нахождения максимального паросочетания минимальной стоимости. В отличие от канонической метаэвристики муравьиного алгоритма в работе агента на графе поиска решений строится максимальная клика, которая является интерпретацией решения. На третьем уровне иерархии решается задача компактизации контейнеров. Процесс распределения блоков по контейнерам сопровождается процедурой сжатия каждой пары блоков, назначаемых в один контейнер. Целью компактизации является минимизация общей площади контейнера путем плотного размещения блоков. Компактизацию последовательно проводят во всех контейнерах. На четвертом уровне иерархии решается задача заполнения полосы контейнерами. В качестве модели для представления решения на графе поиска решений служит клика. Разработана база данных коллективной эволюционной памяти. Разработана методика формирования феромоновых точек и структур данных коллективной эволюционной памяти. Для проведения объективных экспериментов были использованы известные тестовые задачи, представленные в литературе и сети Интернет. По сравнению с существующими алгоритмами достигнуто улучшение результатов на 3-5%. Временная сложность алгоритма, полученная экспериментальным путем, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет ($BCA \approx O(n^2)$).

Раскрой-упаковка; полуограниченная полоса; декомпозиция; методология; задача об упаковке в контейнеры; компакция; поисковая оптимизация; роевой интеллект; муравьиная колония; адаптивное поведение.

B.K. Lebedev, O.B. Lebedev, M.A. Ganzhur

BIO-INSPIRED DENSE PACKING ALGORITHM TO INCREASE THE EFFICIENCY OF SEMI-LIMITED STRIP CUTTING

A methodology has been developed for finding solutions to the semi-infinite strip packing problem based on models of adaptive behavior of biological systems. To reduce the overall labor intensity of the search procedure, an approach based on decomposition of the problem being solved is proposed. The packaging is designed for cutting by guillotine cutting of the tape into containers and non-guillotine cutting of containers into elements. Packaging is carried out by sequentially filling the strip with containers. The problem of packing rectangles into strips is solved in three stages. At the first stage, the agent solves the problem of distributing a set A of rectangular-shaped elements in a set of blocks B . The problem of forming a set of blocks B , including sets of rectangular-shaped elements A , is solved by an algorithm for one-dimensional packing of elements into identical blocks. At the second stage, the problem of distributing blocks among containers is solved. All containers have the same width D , equal to the width of the strip. Each container holds two blocks. The process of distributing blocks into containers is accompanied by a compaction procedure for each pair of blocks assigned to one container. The purpose of compaction is to minimize the total area of the container by densely placing the blocks. Compaction is carried out sequentially in all containers. The problem of distributing blocks into containers is reduced to the problem of finding the maximum matching of the minimum cost. In contrast to the canonical paradigm of the ant algorithm, when working as an agent, a clique is built on the solution search graph, on the edges of which a pheromone is deposited. A technique has been developed for the formation of pheromone points and data structures of collective evolutionary memory. To conduct objective experiments, well-known test problems presented in the literature and on the Internet were used. Compared to existing algorithms, a 3-5% improvement in results was achieved. The time complexity of the algorithm, obtained experimentally, practically coincides with theoretical studies and for the considered test problems is $\approx O(n^2)$.

Packing in an endless strip; decomposition; distribution of blocks into containers; compaction; swarm intelligence; ant colony; adaptive behavior; search engine optimization.

Введение. В работе рассматривается задача упаковки полуограниченной полосы (*1.5 Dimensional Bin Packing, 1.5 DBP*). Задачи раскроя-упаковки (*Bin Packing*) занимают важное место в современной комбинаторной оптимизации и привлекают внимание многих ученых, как в России, так и за рубежом. Интерес к задачам раскроя-упаковки объясняется, в частности, их большой практической значимостью [1–3]. Как правило, приложения задач раскроя-упаковки относятся к материалоемким производствам, где одним из основных факторов снижения себестоимости выпускаемой продукции является рациональное использование ресурсов [4, 5].

Задачи раскроя-упаковки (ЗРУ) относятся к классу NP -трудных задач комбинаторной оптимизации. Для решения задачи раскроя-упаковки российскими и зарубежными учеными разработано большое количество методов и алгоритмов, различающихся как областью применения, так и эксплуатационными свойствами. Оптимальное решение задач раскроя, как правило, дает высокий экономический эффект. Методы решения ЗРУ принято разделять на две большие группы: точные и приближенные [6–8]. Поиск эффективных методов решения NP -полных задач является предметом исследования многих специалистов различных областей науки и практики [9, 10].

На сегодняшний день для решения задач раскроя-упаковки используются различные оптимизационные алгоритмы [11, 12]. Во многих работах уделяется большое внимание разработке приближенных и эвристических методов. В связи с тем, что точные методы, основанные на полном переборе решений, нереализуемы при большой размерности данных, используют более эффективные методы сокращенного перебора.

Одним из перспективных и интенсивно развивающихся направлений в этой области является использование бионических методов и алгоритмов [13, 14]. Результатом непрекращающегося поиска наиболее эффективных методов упаковки стало использование методов биоинспирированного поиска [15–17].

В работе для уменьшения общей трудоемкости поисковой процедуры предлагается подход, основанный на декомпозиции решаемой задачи. Постулат декомпозиции применяется для снижения уровня сложности задачи упаковки путем расчленения общей задачи на подзадачи [18].

Для решения задачи прямоугольной упаковки в полуограниченную полосу предложен новый гибридный подход на основе декомпозиции общей задачи упаковки и построения поисковой процедуры на основе модифицированного метода муравьиной колонии [19]. Для оптимизации раскроя путем минимизации общей площади полосы в общий маршрут упаковки включаются процедуры компактизации (сжатия). Предложена новая методология и метод распределения блоков по контейнерам, являющийся ядром решения задачи упаковки. Разработаны поисковые методы для решения задач гильотинного и не гильотинного прямоугольного раскроя.

Постановка задачи упаковки. Основная задача методологии проектирования состоит в декомпозиции задачи с целью получения последовательности действий, наилучшим образом организующей процесс проектирования. При выборе маршрута решения задачи упаковки, решающее значение имеет связь этапов проектирования в единый процесс. В соответствии с принципом модульности и иерархичности (многоуровневости), работы по упаковке полосы делятся по этапам таким образом, что решения задач, получаемых на одних этапах, служат исходными данными для задач следующих этапов.

Необходимо: произвести упаковку прямоугольников в полуограниченную полосу шириной D не выходя за границы полосы, без перекрытия друг друга с минимизацией занимаемой площади.

Дано: полуограниченная полоса P ; множество прямоугольников $A=\{a_i|i=1,2,...,n\}$, размеры которых задаются: размеры прямоугольников вектором $W=\{w_i|i=1,2,...,n\}$ ширины и вектором $H=\{h_i|i=1,2,...,n\}$ высот прямоугольников.

Требуется найти такую упаковку полосы, для которой высота H^P ее занятой части имеет минимальное значение.

Упаковка полосы осуществляется путем последовательного заполнения контейнерами (рис. 1).

В основе метода декомпозиции общей задачи упаковки и методологии формирования карт раскроя лежат эвристики уровня подхода к упаковке полосы [3–6]. Уровеньные алгоритмы обладают лучшими возможностями улучшения и коррекции решения. Упаковка выполняется путем последовательного заполнения уровней полосы контейнерами.

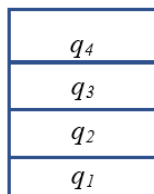


Рис. 1. Упаковка полосы осуществляется путем последовательного заполнения

Архитектура, сформированная на основе декомпозиции общей задачи, включает 5 основных секций: управление процессом поиска; формирования блоков; формирование контейнеров; компактизация контейнеров; заполнения полосы контейнерами. В состав стандартного маршрута поиска решения дополнительно включена процедура распределения блоков между контейнерами и процедура компактизации контейнера. Целью добавленных процедур является минимизация общей площади контейнера путем плотного размещения блоков. Это позволило улучшить характеристики решений, что подтверждается результатами экспериментов. У всех блоков и контейнеров одинаковая ширина, равная ширине полосы – D .

В соответствии с иерархическим подходом к упаковке первой решается задача формирования множества $B=\{b_j|j=1,2,...,n_b\}$ одинаковых по ширине блоков. Элементы в блоках располагаются на горизонтальном основании внутри контейнера вплотную друг к другу.

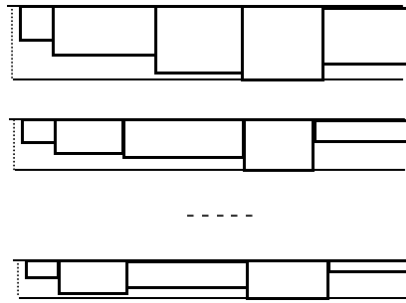


Рис. 2. Множество блоков

Задача распределения множества прямоугольников по блокам сводится к задаче разбиения множества $A = \{A_j | j = 1, 2, \dots, m\}$ на n_b непустых и непересекающихся подмножеств A_j . $A = \cup A_j$, $A_j \neq \emptyset$, $(\forall i, j)[A_i \cap A_j = \emptyset]$. Суммарная ширина W_j прямоугольников множества A_j , входящих в состав блока b_j , определяется как $W_j = \sum w_i$ для $\{i | a_i \in A_j\}$. Значение W_j ограничено размером ширины контейнера, $W_j \leq D$.

В качестве высоты h_j^b блока b_j служит высота самого высокого элемента в блоке.

Критерий оптимизации – суммарная высота сформированного множества блоков: $F = \sum h_j^b$ для $\{j = 1, \dots, m\}$. Цель оптимизации – минимизация F .

Для решения задачи формирования множества блоков $B = \{b_j | j = 1, 2, \dots, n_b\}$, включающих наборы элементов $A = \{a_i | i = 1, 2, \dots, n_a\}$, используется разработанный авторами биоинспирированный алгоритм одномерной упаковки элементов в одинаковые блоки [20].

Второй по важности является задача распределения блоков по контейнерам. В каждом контейнере помещаются два блока. Один над другим. Процесс распределения блоков по контейнерам сопровождается процедурой компактизации (сжатия). Суть компактизации заключается в максимально возможном сближении по вертикали блоков, назначаемых в один контейнер. В результате исследования подходов к компактизации была установлена зависимость плотности упаковки полосы от порядка расположения элементов в блоке. В связи с этим в работе используется эвристика предварительного упорядочивания расположения элементов в блоках по возрастанию высоты элементов в нижнем блоке контейнера с начала блока, в верхнем блоке с конца.

Целью компактизации является минимизация общей площади контейнера путем плотного размещения блоков. Компактизацию последовательно проводят во всех контейнерах. Высота H_k^q контейнера q_k определяется после компактизации. Критерий оптимизации $F = \sum H_k^q$ для $\{k | q_k \in Q\}$. Цель оптимизации – минимизация F .

Размещение блоков в контейнере. В работе [20] рассматриваются два возможных варианта размещения блоков в контейнере, в зависимости от типа разрезов и компактизации контейнера.

Первый вариант сближения блоков в контейнере реализуется при использовании одного типа разреза – гильотинного при раскрое полосы на контейнеры и раскрое контейнеров на элементы. В этом случае блок представляется в виде прямоугольника, у которого верхняя и нижняя стороны являются прямыми линиями. Компактизация реализуется путем сближения пары блоков b_i и b_j без взаимного перекрытия до соприкосновения границ блоков (рис. 3).

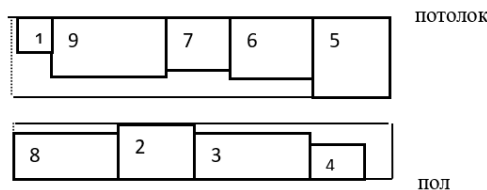


Рис. 3. Первый вариант компактизации

В этом случае высота контейнера q_k определится как:

$$H_k^q = (h_i^b + h_j^b), \quad (1)$$

где h_i^b и h_j^b – высота соответственно верхнего и нижнего блоков, входящих в состав контейнера q_k .

Второй вариант сближения блоков в контейнере реализуется при использовании двух типов разреза:

- ♦ гильотинного при раскрое полосы на контейнеры и не гильотинного разреза при раскрое контейнеров на элементы.

В этом случае блок с одной стороны ограничен прямой линией, а другая стороны блок ограничен ломанной линией. Блоки сближаются до первого касания границ блоков. (рис. 4).

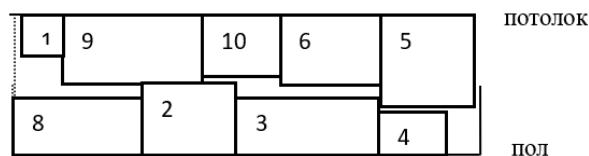


Рис. 4. Сближения блоков с неупорядоченным размещением элементов

Сближение блоков контейнера выполняется до первого касания границ элементов по вертикали. Отметим, что положение элементов друг относительно друга по вертикали в обоих блоках зафиксировано. В результате сближения блоков отдельные прямоугольники верхнего уровня будут соприкасаться с прямоугольниками нижнего уровня. На рис. 4 упорядочивание элементов в блоке не проводилось. Девятый элемент соприкасается с вторым. $H_k^q = (h_9 + h_2)$.

Третий вариант. предусматривает предварительное упорядочивание элементов в блоках: в верхнем блоке элементы расположены по убыванию, а в нижнем блоке по возрастанию высоты прямоугольников.

На рис. 5 соприкасаются второй и десятый элементы. Высота контейнера $H_k^q = (h_{10} + h_2)$. В большинстве случаев такой подход приводит к минимизации высоты контейнера.



Рис. 5. Вариант сближения упорядоченных блоков

Процедура компактизации контейнера. В каждом контейнере помещаются два блока. На рис. 6 представлен контейнер с размещенными в нем блоками. В первый (верхний) блок входят элементы $a_1 - a_5$, во второй (нижний) блок входят элементы $a_6 - a_{10}$.

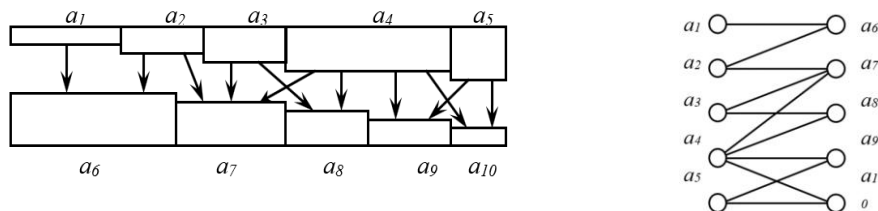


Рис. 6. Формирование графа ограничений

Для формирования графа ограничений разработан алгоритм распространения тени. «Тень» элемента распространяется вдоль направления компакции. От каждого элемента a_i верхнего блока отбрасывается тень на элементы нижнего блока контейнера. Для каждого элемента a_i верхнего блока определяется множество T_i элементов нижнего блока, на которые падает тень элемента a_i верхнего блока. $T_1=\{a_6\}$, $T_2=\{a_6, a_7\}$, $T_3=\{a_7, a_8\}$, $T_4=\{a_7, a_8, a_9, a_{10}\}$, $T_5=\{a_9, a_{10}\}$. На базе a_i и T_i формируется множество пар $P_i=\{p_{ij}\}$ элементов контейнера q_k , находящихся в конфликте, где $p_{ij}=\langle a_i, (a_j \in T_i) \rangle$. Элемент a_i принадлежит верхнему блоку контейнера, а элемент a_j , лежащий в тени элемента a_i , принадлежит нижнему блоку контейнера. На рис. 6:

$$P_1=\{\langle a_1, a_6 \rangle\}, P_2=\{\langle a_2, a_6 \rangle, \langle a_2, a_7 \rangle\}, P_3=\{\langle a_3, a_7 \rangle, \langle a_3, a_8 \rangle\}, \\ P_4=\{\langle a_4, a_7 \rangle, \langle a_4, a_8 \rangle, \langle a_4, a_9 \rangle, \langle a_4, a_{10} \rangle\}, P_5=\{\langle a_5, a_9 \rangle, \langle a_5, a_{10} \rangle\}.$$

На базе множества $P = \cup P_i$ пар p_{ij} формируется взвешенный граф ограничений $G=(A,E)$ каждая вершина $a_i \in A$ соответствует элементу, а ребра отображают ограничения.

Для каждой пары $p_{ij} \in P$ элементов a_i и a_j , находящихся в конфликте друг с другом в контейнере, рассчитывается сумма $h_{ij}=h_i+h_j$ высот элементов, входящих в состав пары p_{ij} . Ребру, связывающему вершины a_i и a_j , присваивается вес, равный h_{ij} . Отыскивается максимальное значение h_k^* , ($\forall i, j$) [$h_k^* \geq h_{ij}$]. Величина h_k^* является минимально возможной высотой h_k^q контейнера q_k . В рассматриваемом примере суммарная высота пары конфликтующих блоков имеет значение: $h_{16}=4$; $h_{26}=4,5$; $h_{27}=4$; $h_{37}=4,5$; $h_{47}=5$; $h_{48}=4,5$; $h_{49}=4,5$; $h_{4,10}=3,5$; $h_{5,10}=4$.

Максимальное значение имеет $h_{47}=5$. Следовательно значение $h_{47}=5$ будет минимальной высотой H_k^q контейнера q_k .

Подход к решению задачи распределения блоков по контейнерам с компакцией. Алгоритм распределения блоков по контейнерам разработан на основе модифицированных метаэвристик адаптивного поведения муравьиной колонии [5, 12].

Так как в каждом контейнере q_k помещаются два блока, то задача распределения множества $B=\{b_j | j=1, 2, \dots, n_b\}$ блоков по контейнерам сводится к задаче разбиения множества блоков B на множество $P=\{p_k | k=1, 2, \dots, n_k\}$ пар, таких, что каждая пара $p_k=(b_i, b_j)$ блоков размещается в одном контейнере q_k . ($\forall i, j$) [$p_i \cap p_j = \emptyset$], $n_k = n_b/2$.

Если число блоков нечетное, то в множество $B=\{b_j | j=1, 2, \dots, n_b\}$ добавляется один пустой блок. Поскольку число блоков становится четным, то формируется $n_b/2$ пар блоков. Распределение блоков по контейнерам осуществляется одновременно с процедурой компакции контейнера.

В качестве пространства поиска решений служит полный граф $G=(X,U)$. Вершинам множества X и соответствует множество блоков $B=\{b_j | j=1, 2, \dots, n_b\}$. $S_1: X \rightarrow B$; $x_j = S_1(b_j)$.

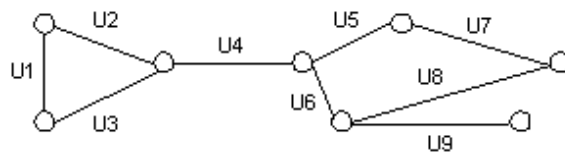
Каждому ребру $u_k \in U$, $u_k=(x_i, x_j)=(S_1(b_i), S_1(b_j))$ графа G , связывающего вершины x_i и x_j , соответствует пара блоков $p_k=(b_i, b_j)$, $b_i=S_1(x_i)$ и $b_j=S_1(x_j)$. Между ребрами u_k и множеством P пар $p_k=(b_i, b_j)$ блоков $B=\{b_j | j=1, 2, \dots, n_b\}$ установлено соответствие $S_2: U \rightarrow P$; $p_k=S_2(u_k)$.

Вес $\varphi(u_k)$ ребра $u_k=(x_i, x_j)$ графа G , связывающего вершины x_i и x_j , равен высоте H_k^q контейнера q_k в состав которого входит пара блоков $b_i=S_1(x_i)$ и $b_j=S_1(x_j)$. $\varphi(u_k)=H_k^q$.

В случае первого варианта сближения пары блоков в контейнере высота контейнера определяется по формуле (1).

Если реализуется второй вариант сближения пары блоков в контейнере, то предварительно в верхнем блоке прямоугольники располагаются по убыванию, а в нижнем блоке по возрастанию высоты прямоугольников. Высота контейнера рассчитывается после процедуры компакции контейнера.

Задача разбиения множества блоков B на множество пар, таких, что каждая пара $p_k=(b_i, b_j)$ блоков размещается в одном контейнере q_k , сводится к задаче построения паросочетания U_1 в графе $G=(X,U)$. $U_1 \subset U$. Основное свойство паросочетания U_1 в графе $G=(X,U)$ заключается в том, что любые два ребра $u_i \in U_1$ и $u_j \in U_1$ не имеют общих вершин, т.е. не смежны. Например: в графе G (рис. 7) паросочетание $U_1=\{u_1, u_4, u_7, u_9\}$.

Рис. 7. Пример графа G

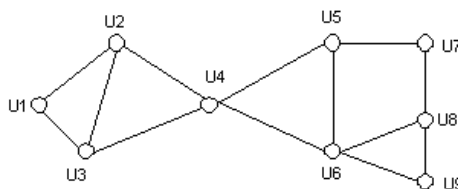
Паросочетание максимальной мощности U_1 полного графа поиска решений $G=(X,U)$ у которого $n_b=|X|$ – четное число, включает $n_b/2$ не смежных ребер.

Обозначим как F суммарный вес ребер, входящих в паросочетание U_1 .

В ряде прикладных задач граф G трансформируется в реберный граф $G_r=(U,V)$. Вершины графа G_r соответствуют рёбрам графа G . Если в графе G рёбра u_i и u_j смежны, т.е. инциденты одной вершине, то вершины u_i и u_j в графе G_r связаны ребром.

На рис. 8. представлен реберный граф G_r графа поиска решений $G=(X,U)$ представленного на рис.7.

Подмножество вершин $P=\{u_1, u_4, u_8\}$ реберного графа G_r (рис. 8) является внутренне устойчивым, т.к. любые две вершины подмножества P не смежны. Паросочетанию $P=\{u_1, u_4, u_7, u_9\}$ графа G соответствует внутренне-устойчивое подмножество $P=\{u_1, u_4, u_7, u_9\}$ реберного графа G_r .

Рис. 8. Реберный граф G_r графа G

Авторами разработан метод и алгоритм последовательного формирования в полном графе $G_r=(U,V)$ максимального внутренне-устойчивого множества вершин $u_k \in U_1$, $U_1 \subset U$, соответствующего паросочетанию V_1 в графе G таких, что любые две вершины $u_i \in U_1$ и $u_j \in U_1$ не являются смежными, $u_i \cap u_j = \emptyset$. Отличительной особенностью является то, что вершина u_k в графе G_r соответствует паре вершин $u_k=(x_i, x_j)$ графа G , которые в свою очередь соответствуют паре блоков $p_k=(b_i, b_j)$ размещаемых в одном контейнере. Пусть $|B|=n_b$, $|U_1|=n_b/2$. Критерий оптимизации – суммарная стоимость вершин в подмножестве U_1 . Цель оптимизации минимизация критерия $F=\sum_k (\varphi(u_k))$ ($k|u_k \in U_1$).

Муравьиный алгоритм распределения блоков по контейнерам. Алгоритм распределения блоков по контейнерам базируется на методе муравьиной колонии. Поиск решений осуществляется в виде двух итерационных процессов – внешнего и внутреннего, вложенного во внешний.

Каждая итерация с внутреннего цикла включает две процедуры, выполняемые агентом z_c , которые выполняются последовательно одна за другой. С помощью первой процедуры на каждой итерации с внутреннего цикла каждый агент z_c формирует конструктивным алгоритмом свое собственное решение – распределение блоков по контейнерам и рассчитывает оценку F_c решения. Лучшее решение внутреннего цикла запоминается.

Вторая процедура заключается в формировании множества феромонных точек, соответствующих решению, и отложении феромона агентом в промежуточные матрицы коллективной эволюционной памяти (КЭП) в количестве, пропорциональном целевой функции.

После выхода из внутреннего цикла выполняются две постпроцедуры.

Первая постпроцедура добавляет в основную матрицу КЭП феромон, накопленный в промежуточной матрице КЭП.

Второй постпроцедурой осуществляется испарение феромона в феромонных точках.

На каждой итерации внешнего цикла последовательно выполняются процедуры внутреннего цикла и две постпроцедуры.

Работа муравьиного алгоритма базируется на использовании коллективной эволюционной памяти (КЭП). КЭП муравьиной колонии представляет собой набор статистических показателей показывающие полезность фрагмента при построении решений на предыдущих итерациях алгоритма. Ядром муравьиного алгоритма является конструктивный алгоритм, с помощью которого каждый член популяции на каждой итерации находит решение задачи.

Для отражения коллективной эволюционной памяти в течение жизни популяции агентов и для формирования решения задачи используется граф учета качества решений $R(U, V^*)$, являющийся дополнением реберного графа $G_r=(U, V)$.

Граф $R(U, V^*)$, имеющий то же множество вершин, что и реберный граф $G_r=(U, V)$, но в котором две несовпадающие вершины u_i и u_j смежны тогда и только тогда, когда они не смежны в графе $G_r=(U, V)$. Формально дополнение графа $G_r=(U, V)$, определяется как граф $R(U, V^*)$, с исходным набором множества вершин U и с набором ребер V^* , полученным из полного графа на множестве вершин U , удалением множества ребер V имевшихся в заданном графе G_r . Дополнение пустого графа (содержащего только вершины, но нет ребер) является полным графом, и наоборот. Независимое множество графа $G_r=(U, V)$ является кликой в дополнении $R(U, V^*)$ графа $G_r=(U, V)$, и наоборот.

Максимальному по мощности паросочетанию в графе G соответствует предельное внутренне-устойчивое подмножество реберного графа G_r , которое является кликой в дополнении $R(U, V^*)$ графа G_r .

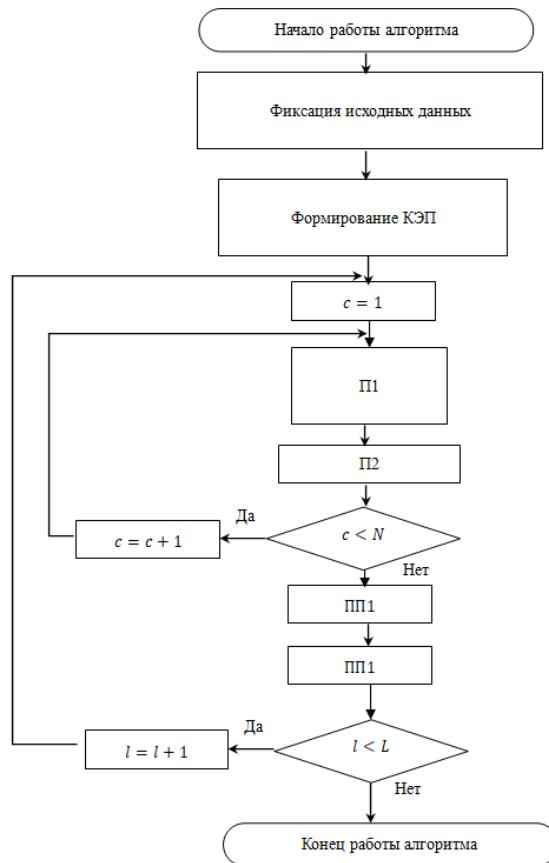


Рис. 9. Структурная схема муравьиного алгоритма

Механизмы формирования в графе независимого подмножества вершин на основе метода муравьиной колонии. Процесс поиска решений в МА итерационный. Каждая итерация l включает три процедуры, которые выполняются последовательно одна за другой. На первом этапе каждой итерации каждый агент z_c формирует конструктивным алгоритмом свое собственное решение – распределение блоков по контейнерам.

На втором этапе агенты откладывают феромон в матрицы коллективной эволюционной памяти (КЭП) в количестве пропорциональном целевой функции. На третьем этапе итерации осуществляется испарение феромона.

В основе конструктивного алгоритма распределения блоков по контейнерам лежит процедура формирования в графе $G_r=(U,V)$ максимального внутри устойчивого множества U_l с минимальным значением суммарного веса элементов (контейнеров), входящих в его состав. Вершине $u_k \in U_l$ соответствует контейнер q_k . Вес сформированного агентом внутренне устойчивого множества U_l определяется после компакции каждого контейнера, «входящего» в состав ВУМ U_l .

Классическая формулировка муравьиного алгоритма включает процедуру отложения феромона в феромонных точках, в качестве которых обычно служат ребра графов. В начальной части МА для откладывания феромона формируется множество $\Phi=\{\varphi_k | k=1,2,\dots,n\}$ феромонных точек, соответствующих ребрам графа $R(U,V^*)$. Между Φ и V^* установлено соответствие $S_2: \Phi \rightarrow V^*$; $\varphi_k = S_2(v_k)$, $v_k \in V^*$ – ребро, связывающее вершины u_i и u_j графа $R(U,V^*)$, $v_k=(u_i, u_j)$.

В качестве коллективной эволюционной памяти (КЭП) используется матрица $\Delta=||\delta_{ij}||_{n \times n}$, в которой для каждого ребра $v \in V^*$ графа решений $R(U,V^*)$, хранится оценка степени пригодности δ_{ij} ребра $v \in V^*$, связывающего вершины u_i и u_j . Степень пригодности ребра определяется количеством отложенного феромона на ребре. В работе используется циклический (ant-cycle) метод муравьиных систем. В этом случае феромон откладывается в матрицу $\Delta=||\delta_{ij}||_{n \times n}$ после полного формирования всеми агентами решений на текущей итерации. Для накопления феромона, откладываемого агентами за одну итерацию, используется накопительная матрица $\Delta^*=||\delta_{ij}^*||_{n \times n}$. Другими словами, в процессе текущей итерации агенты по очереди откладывают феромон в $\Delta^*=||\delta_{ij}^*||_{n \times n}$, а после завершения итерации феромон, накопленный агентами в $\Delta^*=||\delta_{ij}^*||_{n \times n}$, откладывается в матрице $\Delta=||\delta_{ij}||_{n \times n}$.

Формирование ВУМ выполняется агентом z_c по шагам на базе опорного упорядоченного вектора $V=<1,2,3,4,\dots,n>$ номеров блоков с 1 по n с динамически изменяемым составом. На каждом шаге t агентом z_c формируется вершина $u_k \in U$, $u_k=(x_i, x_j)$ графа $G_r=(U,V)$, которая включается в ВУМ $U_{lc}(t)$ графа $G_r=(U,V)$, а номера блоков x_i и x_j исключаются из V . $U_{lc}(t) \subset U$. $U_{lc}(t+1)=U_{lc}(t) \cup u_k$. Отметим, что, если число блоков равно n , то число пар блоков равно $n/2$. Следовательно число вершин, входящих в ВУМ полного графа $G_r=(U,V)$ равно $n/2$. Каждый блок входит в состав только одной вершины $u_k \in U$.

Выбор вершины $u_k \in U$ включает две стадии. На первой стадии каждого шага t агентом z_c осуществляется формирование множества $P(t) \subset U$ вершин u_k графа G_r (пар $u_k=(b_i, b_j)$ блоков), рассматриваемых в качестве претендентов для включения в $U_{lc}(t)$, отличающихся тем, что в состав каждой пары блоков $u_k \in P(t)$ входит блок $b_i = S_1(x_i)$.

Формирование выполняется следующим образом. В векторе $V(t)$ выбирается первый по порядку номер элемента b_i в составе $V(t)$, который будет в составе всех пар множества $P(t)$ блоков первым элементом. Вторыми элементами у множества претендентов $P(t)$ являются оставшиеся элементы вектора $V(t) \setminus b_i$.

Каждая пара $u_k \in P(t)$ может быть включена в $U_{lc}(t)$, но включена будет только одна из них из них.

На второй стадии шага t , выбирается пара блоков $u_k \in P(t)$ которая вносится в состав ВУМ $U_{lc}(t)$, при этом блоки (b_i, b_j) , входящие в u_k удаляются из состава вектора $V(t)$.

Выбор пары блоков $u_k=(b_i \text{ и } b_j)$, входящей в список претендентов $P(t)$, осуществляется вероятностным образом.

Для каждой $u_k \in P(t)$ на графе $R(U, V^*)$ определяется подмножество U_k вершин множества $U_{Ic}(t)$, смежных u_k . $U_k \subset U_{Ic}(t)$.

Рассчитываются стоимость S_k связи вершины $u_k \in P(t)$ с вершинами множества $U_I(t)$ по одной из формул:

$$S_k = (\lambda I_k)^\alpha / (\lambda 2_k + 1)^\beta \quad \text{при мультипликативной свертке,} \quad (1)$$

$$S_k = \alpha(\lambda I_k) + \beta \lambda 2_k \quad \text{при аддитивной свертке,} \quad (2)$$

где λI_k – суммарный объем феромона на ребрах графа $R(U, V^*)$, связывающих u_k с вершинами подмножества $U_{Ic}(t)$; $\lambda 2_k$ – вес вершины u_k .

Настройка значений параметров α и β производится экспериментально.

Далее для каждой вершины $u_k \in P(t)$ рассчитывается вероятность π_k включения в формируемое подмножество $U_I(t)$:

$$\pi_k = S_k / \sum_k (S_k). \quad (3)$$

Выбранная с вероятностью π_k вершина $u_k \in P(t)$ (пара $u_k = (b_i, b_j)$ блоков) включается в множество $U_{Ic}(t)$, а блоки b_i и b_j , входящие в $u_k = (b_i, b_j)$, удаляются из вектора $V(t)$.

При $\alpha = 0$ вероятность выбора вершины u_k зависит только от веса вершины, без учета связей подмножества $U_I(t)$.

При $\beta = 0$ вероятность выбора вершины u_k зависит только от объема феромона на связях с $U_I(t)$, что приводит к субоптимальным решениям.

Соотношение между значениями α и β находится экспериментально.

Пример: число блоков $n=10$. Необходимо сформировать 5 пар. Формируется вспомогательный вектор $V(1) = \langle 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \rangle$. На первом шаге ($t=1$) выбирается первый элемент вектора $V(1) = 1$.

На первой стадии формируется множество пар претендентов для включения в $U_{Ic}(1)$.

$$P_1 = (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (1, 10).$$

Пусть на второй стадии с помощью приведенного выше правила выбрана пара $u_1 = (1, 10)$.

u_1 включается в $U_{Ic}(1)$, а блоки 1, 10 исключаются из вектора $V(1)$.

Вектор $V(2) = \langle 2, 3, 4, 5, 6, 7, 8, 9 \rangle$.

На втором шаге ($t=2$) выбирается первый элемент вектора $V(2) = 2$.

На первой стадии формируется множество пар претендентов для включения в $U_{Ic}(2)$:

$$P_2 = (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (2, 8), (2, 9).$$

Пусть на второй стадии с помощью приведенного выше правила выбрана пара $u_2 = (2, 3)$.

u_2 включается в $U_{Ic}(2)$, а блоки 2, 3 исключаются из вектора $V(2)$.

Вектор $V(3) = \langle 4, 5, 6, 7, 8, 9 \rangle$.

На третьем шаге ($t=3$) выбирается первый элемент вектора $V(3) = 4$.

На первой стадии формируется множество пар претендентов для включения в $U_{Ic}(3)$:

$$P_3 = (4, 5), (4, 6), (4, 7), (4, 8), (4, 9).$$

Пусть на второй стадии с помощью приведенного выше правила выбрана пара $u_3 = (4, 6)$.

u_3 включается в $U_{Ic}(3)$, а блоки 4, 6 исключаются из вектора $V(3)$.

Вектор $V(4) = \langle 5, 7, 8, 9 \rangle$.

На четвертом шаге ($t=4$) выбирается первый элемент вектора $V(4) = 5$.

На первой стадии формируется множество пар претендентов для включения в $U_{Ic}(4)$:

$$P_4 = (5, 7), (5, 8), (5, 9).$$

Пусть на второй стадии с помощью приведенного выше правила выбрана пара $u_4 = (5, 8)$.

u_4 включается в $U_{Ic}(4)$, а блоки 5, 8 исключаются из вектора $V(4)$.

Вектор $V(5)=\langle 7,9 \rangle$ включает только два элемента.

В состав пары u_5 входят блоки 7,9.

ВУМ U_{I_k} , построенное z_k , включает множество вершин (пар):

$u_1=(1,10)$, $u_2=(2,3)$, $u_3=(4,6)$, $u_4=(5,8)$, $u_5=(7,9)$.

Для построенного агентом z_c ВУМ U_{I_c} рассчитывается оценка F_c .

$$F_c = \sum_k \varphi(u_k), (k|u_k \in U_{I_c}).$$

На построенном агентом z_c ВУМ U_{I_c} в графе $R(U, V^*)$ формируется полный подграф Q_{I_c} .

В ячейки матрицы $\Delta^* = \|\delta_{ij}^*\|_{n \times n}$, соответствующие всем ребрам подграфа Q_{I_c} графа $R(U, V^*)$, добавляется порция феромона в объеме:

$$\varepsilon_k(l) = \Theta / F_c(l). \quad (4)$$

После того, как все агенты сформировали решения и добавили каждый из них свою порцию феромона в накопительную матрицу $\Delta^* = \|\delta_{ij}^*\|_{n \times n}$, производится сложение матриц $\Delta = \|\delta_{ij}\|_{n \times n}$ и $\Delta^* = \|\delta_{ij}^*\|_{n \times n}$.

$$\Delta(t+1) = \Delta(t) + \Delta^*(t).$$

На третьем этапе происходит общее испарение феромона на ребрах полного графа R (соответственно в ячейках матрицы КЭП) в соответствии с формулой.

$$\tau_k = \tau_k(1 - \rho). \quad (5)$$

После выполнения всех действий среди решений, полученных агентами на текущей итерации, выбирается лучшее, 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 агентов m , и определяется как $O(l \cdot n^2 \cdot m)$.

Алгоритм распределения блоков по контейнерам. На начальном этапе алгоритма распределения множества блоков по контейнерам формируется стартовый набор данных:

- ◆ формируется секция исходных данных поставленной задачи, включающая параметры полосы и прямоугольных элементов, ограничения;
- ◆ формируется модели пространства поиска решений $G=(X, U)$, $G_r=(U, V)$, $R(U, V^*)$;
- ◆ для каждого ребра $u_k=(x_i, x_j)$ графа G определяется соответствующий ему контейнер q_k , включающий два блока $b_i=S_I(x_i)$ и $b_j=S_I(x_j)$;
- ◆ формируются структуры базы данных КЭП в виде матриц $\Delta(t)$ и Δ^* ;
- ◆ выполняется компакция каждого контейнера q_k , и определяется его вес $\varphi(u_k)$;
- ◆ формирование начального набора UP решений в виде максимальных паросочетаний в графе G и рассчитывается оценка каждого решения;
- ◆ формирование исходной базы данных МА, путем отложения в ячейках матрицы $\Delta(t)$ феромона, пропорционального оценкам паросочетаний набора UP ;
- ◆ задаются начальные значения управляющих параметров: C – число агентов, L – число итераций.

Агенты имеют собственную «память». На каждом шаге t в памяти агента z_c имеется:

- ◆ множество $U_{I_c}(t) \subset U$ вершин, уже вошедших в состав формируемого ВУМ;
- ◆ опорный упорядоченный вектор $V(t)$;
- ◆ матрицы $\Delta(t) = \|\delta_{ij}\|_{n \times n}$ и $\Delta^*(c) = \|\delta_{ij}^*\|_{n \times n}$;
- ◆ δ_{ij} – количество феромона, на каждом ребре $v_{iv}=(x_i, p_v)$ графа $R(U, V^*)$.

1. Формирование исходных данных.
2. Формирование моделей поиска решений $G=(X, U)$, $G_r=(U, V)$, $R(U, V^*)$.
3. Для каждого ребра $u_k=(x_i, x_j)$ графа G определяется соответствующий ему контейнер q_k , выполняется компакция контейнера q_k , и определяется его вес (высота) $\varphi(u_k)$.
4. Формирование структур коллективной эволюционной памяти – основной и вспомогательной матриц

$$\Delta(l) = \|\delta_{ij}\|_{n \times n}, \text{ и } \Delta^*(c) = \|\delta_{ij}^*\|_{n \times n} \text{ степеней пригодности решений.}$$

$$5. \text{ Обнуление матриц } \Delta(l) = \|\delta_{ij}\|_{n \times n} \text{ и } \Delta^*(c) = \|\delta_{ij}^*\|_{n \times n}.$$

Формирование начальной базы данных КЭП МА.

6. Формирование на базе графа $G=(X,U)$ начального набора UP максимальных паросочетаний U_{li} ,

$UP=\{U_{li}|i=1,2,\dots,n\}$. $U_{li}\in UP$ является ВУМ в $G_r=(U,V)$ и кликой в $R(U,V^*)$.

7. Расчет для каждого паросочетания U_{li} суммарного веса F_i входящих в его состав вершин.

$$F_i=\sum_k (\varphi(u_k)) (k|u_k\in U_{li}).$$

8. На каждом множестве вершин U_{li} графа $R(U,V^*)$ формируется полный подграф Q_{li} .

9. Для каждого Q_{li} формируется множество феромонных точек Φ_{li} , соответствующих ребрам подграфа

$$Q_{li}.$$

10. На каждом множестве феромонных точек Φ_{li} в матрице $\Delta=||\delta_{ij}||_{n \times n}$ откладывается начальное

$$\text{количество феромона } \tau_k(l)=\Theta/F_i$$

11. Задается число итераций – L . Задается число агентов – N_a . $l=1$.

12. Начало новой итерации. $l=l+1$. $c=0$. (l – номер итерации, c – номер агента).

13. Выбор очередного агента. $c=c+1$.

Начало формирования паросочетания агентом z_c .

Приводится в начальное состояние память агента z_c .

Обнуляется вспомогательная матрица $\Delta^*(c)=||\delta_{ij}^*||_{n \times n}$.

14. Формируются начальные значения параметров памяти агента z_c . $t=0$. $U_{lc}(1)=\emptyset$.

15. Формируется опорный упорядоченный вектор $V(t)=\langle 1,2,3,4,\dots,n \rangle$ номеров блоков с 1 по n с динамически изменяемым составом.

16. $t=t+1$. (t – номер шага).

В векторе $V(t)$ выбирается первый по порядку в составе $V(t)$ номер блока b_i .

17. Формирование множества $P(t)\subset U$, $|P(t)|=n_b$ вершин u_k (соответствующих парам блоков), рассматриваемых в качестве претендентов для включения в $U_{lc}(t)$, отличающихся тем, что в состав каждой пары блоков $u_k\in P(t)$ в качестве первого элемента входит блок b_i .

Вторыми элементами у множества претендентов $P(t)$ являются оставшиеся элементы вектора $V(t)\setminus b_i$.

18. Для каждой u_k из списка претендентов $P(t)$ на графе $R(U,V^*)$:

♦ Определяется множество $U_k\subset U$ смежных ей вершин $u_j\in U_{lc}(t)$.

♦ Рассчитываются показатели:

$\lambda 1_k$ – общий вес феромона на ребрах графа $R(U,V^*)$, связывающих $u_k\in P(t)$ со смежными ей вершинами $u_i\in U_{lc}(t)$;

$\lambda 2_k$ – вес вершины u_k .

♦ Определяется стоимость S_k связей вершины $u_k\in P(t)$ с вершинами множества $U_{lc}(t)$

$$S_k=\alpha(\lambda 1_k)+\beta \lambda 2_k$$

♦ Рассчитывается вероятность π_k включения вершины $u_k\in P(t)$ в формируемое подмножество $U_{lc}(t)$

$$\pi_k=S_k/\sum_k (S_k).$$

19. Агент с вероятностью $\pi_k=W_k/\sum_k (W_k)$ выбирает одну из вершин $u_k\in P(t)$, которая включается в множество $U_{lc}(t)$, а блоки, входящие в состав выбранной пары u_k , исключаются из состава вектора $V(t)$.

20. Если $t<n/2$, то переход к п. 14, иначе переход к п. 19.

21. Рассчитывается оценка F_c ВУМ U_{lc}

$$F_c=\sum_k (\varphi(u_k)) (k|u_k\in U_{lc}).$$

22. Если $F_c < F_{onm}$ то $F_{onm}=F_c$, $U_{ionm}=U_{lc}$.

23. На множестве вершин U_{lc} графа $R(U,V^*)$ формируется клика Q_{lc} .

24. На множестве феромонных точек, соответствующих ребрам клики Q_{lc} , в матрице $\Delta^*(c)=||\delta_{ij}^*||_{n \times n}$ откладывается феромон в количестве $\tau_c(l)=\Theta/F_c(l)$.

25. Если $c < N_w$, то переход к п. 13, иначе переход к п. 26.

26. Феромон, накопленный агентами популяции в накопительной матрице $\Delta^*(c) = ||\delta_{ij}^*||_{n \times n}$ на итерации l добавляется в матрицу $\Delta = ||\delta_{ij}||_{n \times n}$. $\Delta(l) = \Delta(l) + \Delta^*(c)$.

27. Этап испарения феромона на всех ребрах графа $R(U, V^*)$. Значения всех ячеек матрицы $\Delta(l) = ||\delta_{ij}||_{n \times n}$ уменьшаются в соответствии с формулой: $\delta_{ij} = \delta_{ij}(1 - \rho)$.

28. Если $l < L$, то переход к п. 12, иначе переход к п. 29.

29. Конец работа алгоритма.

Результаты эксперимента. Разработанный авторами алгоритм упаковки в полуограниченную полосу на основе декомпозиции и гибридизации биоинспирированных методов поисковой оптимизации реализован в виде программы DBK. Эксперименты проводились на основе методики, изложенной в работе [20], в которой исследовался алгоритм GBA.

Для оценки эффективности разработанных методик и программного обеспечения использовались данные из электронной библиотеки OR-library. Эта библиотека содержит примеры упаковки, для которых заранее известно оптимальное значение высоты занятой полосы. Разработанный алгоритм сравнивался с алгоритмами GAGA, PH, DDA и GBA.

Алгоритм GAGA, описанный в исследовании [21], разработан на основе гибридизации конструктивного и генетического методов. Результаты исследования показали высокую вычислительную эффективность. Алгоритм исследования PH [22] разработан на основе механизмов, инспирированных природой. Алгоритм DDA основан на эвристике динамической декомпозиции [23]. Алгоритмом GBA [20] задача упаковки реализуется при использовании гильотинного разреза ленты на контейнеры и гильотинного разреза контейнеров, реализован на основе методов роевого интеллекта.

На первом этапе агентом решается задача упаковки набора элементов прямоугольной формы в множество блоков прямоугольной формы с фиксированными размерами высоты и ширины. Все блоки имеют одинаковую ширину D , равную ширине полосы. Тестирования вышеуказанных алгоритмов проводилось на тестовых задачах SCPL1 ~ SCPL9 из OR-Benchmark. Результаты расчетов представлены в табл. 1, где N – число прямоугольников, H^* оптимальное значение.

Таблица 1

Сравнение с тестами из OR-Benchmark

Inst	N	Opt	GAGA	PH	DDA	GBA	DBK
		H^*	H	H	H	H	H
SCPL1	425	110	126.08	183.5	123	121	116
SCPL2	127	120	140.16	163.5	128	125	122
SCPL3	225	84	98.84	102	98	90	88
SCPL4	365	102	108.263	109.5	106	102	102
SCPL5	165	102	118.89	118	110	102	102
SCPL6	657	126	142.4	134	132	132	128
SCPL7	357	198	229.93	242	216	204	200
SCPL8	475	156	168.419	167	165	164	162
SCPL9	175	117	129.96	127	125	121	119

Как видно из табл. 1, результаты алгоритма DBK лучше, чем у сравниваемых алгоритмов, и некоторые результаты достигли предельной высоты.

Набор данных случайно выбран из OR-Benchmark. (<http://www.laria.upicardie.fr/hifi/OR-Benchmark>). Элементы имеют ширину 135, высота не ограничена. Используя тестовые данные OR-Library, разработанный алгоритм DBK сравнивался с алгоритмами EA, DDA и GBA. Результаты приведены в табл. 2.

Таблица 2

Сравнение с тестами из OR- Library

<i>Test</i>	<i>H*</i>	<i>Type</i>	<i>n</i>	<i>EA</i>	<i>DDA</i>	<i>GBA</i>	<i>DBK</i>
				<i>H</i>	<i>H</i>	<i>H</i>	<i>H</i>
cgcut 1	23	7	16	23	23	23	23
cgcut 2	63	10	23	67	63	63	63
cgcut 3	636	20	62	670	642	640	638
ngcut 1	20	5	10	23	20	20	20
ngcut 2	28	7	17	30	28	28	28
ngcut 3	28	10	21	28	28	28	28
ngcut 4	18	5	7	20	18	18	18
ngcut 5	36	7	14	36	36	36	36
ngcut 6	29	10	15	31	29	29	29
ngcut 7	10	5	8	20	10	10	10
ngcut 8	33	7	13	33	33	33	32
ngcut 9	49	10	18	50	49	49	49
ngcut 10	59	5	13	80	59	59	59
ngcut 11	51	7	15	52	51	51	51
ngcut12	77	10	22	87	77	77	77
beng01	30	20	20	30	30	30	30
beng02	57	40	40	58	57	57	57
beng03	84	60	60	85	84	84	84
beng04	107	80	80	108	107	107	107
beng05	134	100	100	134	134	134	134
beng06	36	40	40	37	36	36	36
beng07	67	80	80	67	67	67	67

Сравнение с результатами исследований алгоритмов *EA*, *DDA*, *GBA* показывают, что разработанный алгоритм имеет явное преимущество с точки зрения времени вычислений и точности вычислений.

По сравнению с существующими алгоритмами достигнуто улучшение результатов раскроя-упаковки полуограниченной полосы на 3-5%. Полученная временная сложность алгоритма, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет ($BCA \approx O(n^2)$).

Заключение. Предложена архитектура и методология раскроя-упаковки полуограниченной полосы на основе методов биоинспирированного поиска. В основе метода декомпозиции общей задачи упаковки и методологии формирования карт раскроя лежат эвристики уровневого подхода к упаковке полосы. Уровневые алгоритмы обладают лучшими возможностями улучшения и коррекции решения. Упаковка выполняется путем последовательного заполнения уровней полосы контейнерами. Архитектура, сформированная на основе декомпозиции общей задачи, включает 5 основных секций: управление процессом поиска; формирования блоков; формирование контейнеров; компакция контейнеров; заполнения полосы контейнерами. В состав стандартного маршрута поиска решения дополнительно включена процедура распределения блоков между контейнерами и процедура компактизации контейнера. Целью добавленных процедур является минимизация общей площади контейнера путем плотного размещения блоков. Это позволило улучшить характеристики решений, что подтверждается результатами экспериментов.

Компактизацию последовательно проводят во всех контейнерах. Анализ результатов распределения блоков по контейнерам показал, что максимальная плотность упаковки полосы, зависит от распределения блоков по контейнерам. В работе рассматриваются два возможных варианта размещения блоков в контейнере, в зависимости от сочетания типов разрезов: гильотинный и не гильотинный, и компактизации контейнера.

Разработана методика и конструктивный метод распределения блоков по контейнерам, используемый агентами, отличающийся линейной трудоемкостью. Формирование контейнеров выполняется последовательно по шагам. На первой стадии шага *t* формиру-

ется множество пар блоков-претендентов для назначения в очередной контейнер. На второй стадии вероятностным образом выбирается пара блоков, входящей в список претендентов.

Разработана методика и метод компакции контейнера. Компакция выполняется на основе графа ограничений, формируемого методом распространения тени.

Разработана методика формирования феромоновых точек и структур данных коллективной эволюционной памяти. Предложены новые модели и структуры данных для организации коллективной эволюционной памяти. Задача распределения блоков по контейнерам сведена к задачам построения паросочетания и выделения внутренне-устойчивого множества, для решения которых разработаны эффективные биоинспирированные алгоритмы.

Разработанный алгоритм формирования карты раскроя на основе биоинспирированного поиска позволяет значительно сократить отходы материала.

По сравнению с существующими алгоритмами достигнуто улучшение результат раскроя-упаковки полуограниченной полосы на 3-5%. Временная сложность алгоритма (BCA), полученная экспериментальным путем, практически совпадает с теоретическими исследованиями и для рассмотренных тестовых задач составляет ($BCA \approx O(n^2)$).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Stephen C.H., Defu Z.* A fast layer-based heuristic for non-guillotine strip packing // Expert System with Application. – 2011. – P. 13032-13042.
2. *Тимофеева О.П., Соколова Э.С., Милов К.В.* Генетический алгоритм в оптимизации упаковки контейнеров // Труды НГТУ. Информатика и системы управления. – 2013. – № 4 (101). – С. 167-172.
3. *Валеева А.Ф.* Применение конструктивных эвристик в задачах раскроя-упаковки // Приложение к журналу «Информационные технологии». – 2006. – № 11. – С. 1-24.
4. *Лебедев Б.К., Лебедев О.Б., Лебедева Е.М.* Модернизированный муравьиный алгоритм синтеза идентифицированного дерева гильотинного разреза при планировании СБИС // Известия ЮФУ. Технические науки. – 2017. – № 7. – С. 15-28.
5. *Лебедев О.Б., Зорин В.Ю.* Упаковка на основе метода муравьиной колонии // Известия ЮФУ. Технические науки. – 2010. – № 12. – С. 25-30.
6. *Defu Zhang, Yuxin C., Furong Y., Yain-Whar S., Stephen C.* A hybrid algorithm based on variable neighbourhood for the strip packing problem // Journal of Combinatorial Optimization. – 2016. – P. 513-530.
7. *Ахтямов А.А., Картак В.М.* Упаковка и оценка плотности упаковки ортогональных многоугольников в полубесконечную полосу // Информационные технологии интеллектуальной поддержки принятия решений. – 2014. – С. 117-121.
8. *Jaya T., Narendra S.* Hybrid approach for 2d strip packing problem using genetic algorithm // Proceedings of the 12th international conference on Artificial Neural Networks: advances in computational intelligence. – 2013.
9. *Месягутов М.А., Мухачева Э.А., Белов Г.Н., Шайтхауэр Г.* Упаковка одномерных контейнеров с продолженным выбором идентичных предметов: точный метод поиска оптимального решения // Автоматика и телемеханика. – 2011. – № 1. – С. 154-173.
10. *Lei W., Aihua Y.* A quasi-human algorithm for the two dimensional rectangular strip packing problem: in memory // Journal of Combinatorial Optimization. – 2016. – P. 416-444.
11. *Потарусов Р.В., Курейчик В.М.* Проблема одномерной упаковки элементов // Известия ТРТУ. – 2006. – № 8. – С. 88-93.
12. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2021. – 448 с.
13. *Лебедев О.Б.* Модели адаптивного поведения муравьиной колонии в задачах проектирования. – Таганрог: Изд-во ЮФУ, 2013. – 199 с.
14. *Лебедев Б.К., Лебедев О.Б.* Муравьиные алгоритмы разбиения, использующие представления задачи, отличные от канонического // Вестник Ростовского государственного университета путей сообщения. – 2016. – № 3 (63). – С. 42-47.
15. *Blum C., Roli A.* Metaheuristics in combinatorial optimization: overview and conceptual comparison // ACM computing surveys. – 2003. – P. 268-308.

16. Лебедев Б.К., Лебедев О.Б. Распределение ресурсов на основе гибридных моделей роевого интеллекта // Научно-технический вестник информационных технологий, механики и оптики. – 2017. – № 6. – С. 1063-1073.
17. Житников В.П. Задача прямоугольной упаковки в полубесконечную полосу: поиск решения в окрестности локальной нижней границы // Информационные технологии. – 2007. – № 6. – С. 55-61.
18. Raidl G.R. A Unified View on Hybrid Metaheuristics // Lecture Notes In CS. – 2006. – P. 1-12.
19. Лебедев Б.К., Лебедев О.Б. Роевой алгоритм планирования работы многопроцессорных вычислительных систем // Инженерный вестник Дона. – 2017. – № 3.
20. Лебедев Б.К., Лебедев О.Б., Ганжур М.А. Упаковка в полубесконечную полосу на основе декомпозиции и гибридизации биоинспирированных методов // Известия ЮФУ. Технические науки. – 2023. – № 4. – С. 109-125.
21. Cong J., Romesio M., Xie M. Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms // Proc. of the International Symposium on Physical Design. – 2003. – P. 88-94.
22. Hifi M., M'Hallah R. A hybrid algorithm for the two-dimensional layout problem: The cases of regular and irregular shapes // International Transactions in Operational Research. – 2003. – P. 195-216.
23. Zhang DeFu, Han ShuiHua, Ye WeiGuo. A Bricklaying Heuristic Algorithm for the Orthogonal Rectangular Packing Problem // Chinese Journal of Computers. – 2008. – P. 509- 515.

REFERENCES

1. Stephen C.H., Defu Z. A fast layer-based heuristic for non-guillotine strip packing, *Expert System with Application*, 2011, pp. 13032-13042.
2. Timofeeva O.P., Sokolova E.S., Milov K.V. Geneticheskiy algoritm v optimizatsii upakovki konteynerov [Genetic algorithm in optimization of container packaging], *Trudy NGTU. Informatika i sistemy upravleniya* [Proceedings of NSTU. Informatics and control systems], 2013, No. 4 (101), pp. 167-172.
3. Valeeva A.F. Primenenie konstruktivnykh evristik v zadachakh raskroya-upakovki [Application of constructive heuristics in cutting-packing problems], *Prilozhenie k zhurnalu «Informatsionnye tekhnologii»* [Supplement to the journal «Information Technologies»], 2006, No. 11, pp. 1-24.
4. Lebedev B.K., Lebedev O.B., Lebedeva E.M. Modernizirovannyi murav'inyy algoritm sinteza identifikirovannogo dereva gil'otinnogo razreza pri planirovanii SBIS [Modernized ant algorithm for synthesizing an identified guillotine cut tree when planning VLSI], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2017, No. 7, pp. 15-28.
5. Lebedev O.B., Zorin V.Yu. Upakovka na osnove metoda murav'inoi kolonii [Packaging based on the ant colony method], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2010, No. 12, pp. 25-30.
6. Defu Zhang, Yuxin C., Furong Y., Yain-Whar S., Stephen C. A hybrid algorithm based on variable neighbourhood for the strip packing problem, *Journal of Combinatorial Optimization*, 2016, pp. 513-530.
7. Akhtyamov A.A., Kartak V.M. Upakovka i otsenka plotnosti upakovki ortogonal'nykh mnogo-ugol'nikov v polubeskonechnuyu polosu [A hybrid algorithm based on variable neighbourhood for the strip packing problem], *Informatsionnye tekhnologii intellektual'noy podderzhki prinyatiya resheniy* [Information technologies for intelligent decision support], 2014, pp. 117-121.
8. Jaya T., Narendra S. Hybrid approach for 2d strip packing problem using genetic algorithm, *Proceedings of the 12th international conference on Artificial Neural Networks: advances in computational intelligence*, 2013.
9. Mesyagutov M.A., Mukhacheva E.A., Belov G.N., Shaytkhauer G. Upakovka odnomernykh konteynerov s prodolzhenym vyborom identichnykh predmetov: tochnyy metod poiska optimal'nogo resheniya [Packing one-dimensional containers with continued selection of identical items: an exact method for finding the optimal solution], *Avtomatika i telemekhanika* [Automation and Remote Control], 2011, No. 1, pp. 154-173.
10. Lei W., Aihua Y. A quasi-human algorithm for the two dimensional rectangular strip packing problem: in memory, *Journal of Combinatorial Optimization*, 2016, pp. 416-444.
11. Potarusov R.V., Kureychik V.M. Problema odnomernoy upakovki elementov [The problem of one-dimensional packing of elements], *Izvestiya TRTU* [Izvestiya TSURE], 2006, No. 8, pp. 88-93.
12. Karpenko A.P. Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature: a tutorial]. Moscow: Izd-vo MGTU im. N.E. Bauman, 2021, 448 p.
13. Lebedev O.B. Modeli adaptivnogo povedeniya murav'inoi kolonii v zadachakh proektirovaniya [Models of adaptive behavior of an ant colony in design problems]. Taganrog: Izd-vo YuFU. 2013, 199 p.

14. *Lebedev B.K., Lebedev O.B.* Murav'inye algoritmy razbieniya, ispol'zuyushchie predstavleniya zadachi, otlichnye ot kanonicheskogo [Ant partitioning algorithms using problem representations different from the canonical one], *Vestnik Rostovskogo gosudarstvennogo universiteta puty soobshcheniya* [Bulletin of the Rostov State University of Transport and Communications], 2016, No. 3 (63), pp. 42-47.
15. *Blum C., Roli A.* Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM computing surveys*, 2003, pp. 268-308.
16. *Lebedev B.K., Lebedev O.B.* Raspredelenie resursov na osnove gibridnykh modeley roevogo intellekta [Resource distribution based on hybrid models of swarm intelligence], *Nauchno-tekhicheskiy vestnik informatsionnykh tekhnologiy, mekhaniki i optiki* [Scientific and Technical Bulletin of Information Technologies, Mechanics and Optics], 2017, No. 6, pp. 1063-1073.
17. *Zhitnikov V.P.* Zadacha pryamougol'noy upakovki v polubeskonechnuyu polosku: poisk resheniya v okrestnosti lokal'noy nizhney granitsy [The problem of rectangular packing in a semi-infinite strip: searching for a solution in the neighborhood of the local lower boundary], *Informatsionnye tekhnologii* [Information Technologies], 2007, No. 6, pp. 55-61.
18. *Raidl G.R.* A Unified View on Hybrid Metaheuristics, *Lecture Notes In CS*, 2006, pp. 1-12.
19. *Lebedev B.K., Lebedev O.B.* Roevoy algoritm planirovaniya raboty mnogoprotsessornykh vychislitel'nykh sistem [Swarm algorithm for scheduling the operation of multiprocessor computing systems], *Inzhenernyy vestnik Dona* [Engineering Bulletin of the Don], 2017, No. 3.
20. *Lebedev B.K., Lebedev O.B., Ganzhur M.A.* Upakovka v polubeskonechnuyu polosku na osnove dekompozitsii i gibridizatsii bioinspirovannykh metodov [Packaging in a semi-infinite strip based on decomposition and hybridization of bioinspired methods], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2023, No. 4, pp. 109-125.
21. *Cong J., Romesis M., Xie M.* Optimality, Scalability and Stability Study of Partitioning and Placement Algorithms, *Proc. of the International Symposium on Physical Design*, 2003, pp. 88-94.
22. *Hifi M., M'Hallah R.* A hybrid algorithm for the two-dimensional layout problem: The cases of regular and irregular shapes, *International Transactions in Operational Research*, 2003, pp. 195-216.
23. *Zhang DeFu, Han ShuiHua, Ye WeiGuo.* A Bricklaying Heuristic Algorithm for the Orthogonal Rectangular Packing Problem, *Chinese Journal of Computers*, 2008, pp. 509-515.

Статью рекомендовал к опубликованию д.т.н., профессор А.В. Боженюк.

Лебедев Борис Константинович – Южный федеральный университет; e-mail: lebedev.b.k@gmail.com; г. Таганрог, Россия; тел.: 89282897933; кафедра систем автоматизированного проектирования; д.т.н.; профессор.

Лебедев Олег Борисович – МИРЭА – Российский технологический университет; e-mail: lebedev.ob@mail.ru; г. Москва, Россия; тел.: 89085135512; кафедра информатики; д.т.н.; профессор.

Ганжур Марина Александровна – Донской государственный технический университет; e-mail: mganzhur@yandex.ru; г. Ростов-на-Дону; тел.: 89081819111; кафедра вычислительных систем и информационной безопасности; старший преподаватель.

Lebedev Boris Konstantinovich – Southern Federal University; e-mail: lebedev.b.k@gmail.com; Taganrog, Russia; phone: +79282897933; the department of Computer Aided Design; dr. of eng. sc.; professor.

Lebedev Oleg Borisovich – MIREA – Russian Technological University; e-mail: lebedev.ob@mail.ru; Moscow, Russia; phone: +79085135512; the department of Computer Science; dr. of eng. sc.; professor.

Ganzhur Marina Aleksandrovna – Don State Technical University; e-mail: mganzhur@yandex.ru; Rostov-on-Don; phone: +79081819111; the department of Computing Systems and Information; security senior lecturer.