

УДК 004.3

В. М. КАРТАК

## ЗАДАЧА УПАКОВКИ ПРЯМОУГОЛЬНИКОВ: ТОЧНЫЙ АЛГОРИТМ НА БАЗЕ МАТРИЧНОГО ПРЕДСТАВЛЕНИЯ

Рассматривается NP-трудная задача упаковки прямоугольников в полубесконечную полосу, которая имеет широкое применение в промышленности. Рассматривается матричное представление прямоугольной упаковки и ее различные свойства, которые позволяют создать эффективный алгоритм. Алгоритм трансформируется на задачу упаковки  $n$ -мерных параллелепипедов. *Метод ветвей и границ; раскрой-упаковка*

### ВВЕДЕНИЕ

Задача упаковки прямоугольников в полубесконечную полосу (также двумерная задача раскрой-упаковки, two-dimensional strip packing problem, 2DSPP) состоит в следующем: требуется разместить заданные прямоугольники в полубесконечной полосе так, чтобы длина занятой части полосы достигла минимума. Согласно Дикхоффов (Dyckhoff's) типологии раскрой и упаковки [1] двумерная задача раскрой-упаковки имеет тип 2/N/O. Задача 2DSPP возникла в середине XX века и с тех пор не теряет актуальности, поскольку к ее решению сводятся прикладные задачи распределения двумерного ресурса, такие как: задачи планирования занятости и составления расписаний; задачи распределения памяти ЭВМ; собственно проблемы упаковки и раскрой. Вместе с тем большинство работ в указанной области посвящены приближенным методам решения данной задачи. Это обусловлено тем, что 2DSPP является NP-трудной задачей. На данный момент в мире существует лишь несколько подходов, позволяющих решать двумерную задачу раскрой-упаковки точно [2].

Большинство точных методов решения 2DSPP сводится к перебору всего множества допустимых решений и нахождения среди них оптимума. Такой перебор может быть доведен до эффективных алгоритмов [2] разного вида улучшениями. Значительная группа методов улучшенного перебора известна под названием «метода ветвей и границ» (МВВ). S. Martello и D. Vigo применили МВВ для решения 2DSPP [3].

Точный метод решения 2DSPP, ориентированный на небольшое количество прямоугольников, был так же разработан А. И. Ли-

повецким (УрО АН СССР). В работе [4] он ввел понятие «зоны» таким образом, что решение двумерной задачи раскрой-упаковки стало сводиться к перебору допустимых зон, а также показал, что задача упаковки прямоугольников в полубесконечную полосу имеет сложность  $O(m!^2)$ , где  $m$  — число прямоугольников. Использование ряда правил отсечения бесперспективных ветвей позволило сократить перебор. Метод исследован и реализован В. В. Бухваловой [5].

S. P. Fekete, J. Schepers [6] предложили представлять упаковки в виде набора интервальных графов. Разработанный ими алгоритм осуществляет перебор указанных графов. В работе показано, что одному интервальному графу соответствует некоторое семейство упаковок. Они использовали этот подход для решения рюкзачной задачи.

Другой подход к решению 2DSPP был предложен Ю. Г. Стояном [7] и М. В. Новожиловой [8], которые предлагают использовать методы поиска глобального минимума функции цели для задачи 2DSPP. Условия размещения прямоугольников описываются системой из  $2m(m-1)$  линейных неравенств относительно координат полюсов прямоугольников и таким же числом целочисленных переменных. Полученная система неравенств может быть решена с использованием методов линейного программирования. Недостатком данного подхода является то обстоятельство, что число неравенств и целочисленных переменных быстро растет с увеличением числа прямоугольников.

В данной статье описан новый подход к нахождению оптимальной упаковки, базирующийся на ее матричном представлении. Использование свойств данной модели позволя-

ет создавать высокоэффективные алгоритмы решения рассматриваемой задачи.

Первоначально упаковка разбивается на вертикальные и горизонтальные полосы. Вертикальное разбиение впервые было предложено Э. А. Мухачевой и позже применено для эвристического поиска А. С. Мухачевой [9]. Далее каждому разбиению сопоставляется бинарная матрица с элементами  $a_{ij} = 1$ , если  $i$ -й прямоугольник пересекается с  $j$ -м полосой, в противном случае  $a_{ij} = 0$ . Алгоритм сводится к построению представленных бинарных матриц и выбора из них тех, которые отвечают оптимальному результату.

## 1. ПОСТАНОВКА ЗАДАЧИ

Имеется полубесконечная полоса  $S$  фиксированной ширины  $W$  и множество прямоугольников  $R = \{R_i, i = \overline{1, m}\}$  с размерами  $(l_i, w_i)$ , где  $l_i$  — длина,  $w_i$  — ширина  $i$ -го прямоугольника,  $m$  — количество прямоугольников. Обозначим исходные данные как  $(W, m, R)$ . Требуется разместить прямоугольники в полосе так, чтобы длина занятой части полосы достигла минимума.

Допустимой упаковкой прямоугольников  $R$  в полубесконечную полосу  $S$  называется размещение прямоугольников внутри полосы  $S$ , удовлетворяющее следующим условиям:

- стороны прямоугольников из  $R$  параллельны сторонам полосы  $S$ ;
- прямоугольники из  $R$  между собой не перекрываются (не имеют общих внутренних точек).

Введем прямоугольную систему координат, в которой оси  $O_x, O_y$  совпадают с нижней и левой сторонами полосы, и обозначим допустимую упаковку через  $P = \{P_1, P_2, \dots, P_m\}$ , где  $P_i = (x_i, y_i)$  — координаты левого нижнего угла  $i$ -го прямоугольника. Тогда координаты прямоугольников в допустимой упаковке должны удовлетворять следующим условиям:

$$x_i \geq 0, y_i \geq 0, y_i + w_i \leq W, i = \overline{1, m}. \quad (1)$$

$$\begin{aligned} & (x_i + l_i \leq x_j) \vee (x_j + l_j \leq x_i) \vee \\ & \vee (y_i + w_i \leq y_j) \vee (y_j + w_j \leq y_i), \quad (2) \\ & i, j = \overline{1, m}, i \neq j. \end{aligned}$$

Допустимая упаковка  $P_o$ , для которой длина занятой части полосы  $L = \max_{i=1, m} (x_i + l_i)$  достигает минимума, называется оптимальной.

## 2. МАТРИЧНОЕ ПРЕДСТАВЛЕНИЕ УПАКОВКИ

Как уже говорилось во введении, 2DSPP принадлежит к классу NP-трудных задач. В работе [11] было показано, что каждой упаковке можно сопоставить бинарные матрицы специального вида.

### 2.1. Вертикальное и горизонтальное разбиение упаковки

Пусть для некоторой задачи  $(W, m, R)$  известна упаковка  $P$ . Выполним мысленно вертикальные «резы», проходящие через правые стороны прямоугольников  $R$ . Таким образом, вся упаковка делится на вертикальные полосы. Обозначим  $k_x$  — число вертикальных полос. Аналогично выполним мысленно горизонтальные резы, проходящие через верхние стороны прямоугольников,  $k_y$  — число горизонтальных полос.

**Пример 1.** Для задачи 2DSPP с исходными данными  $(W, m, R)$ , где  $W = 55$ ,  $m = 6$ ,  $R_1 = (20, 28)$ ,  $R_2 = (30, 27)$ ,  $R_3 = (63, 10)$ ,  $R_4 = (22, 45)$ ,  $R_5 = (26, 45)$ ,  $R_6 = (15, 15)$ , известна упаковка  $P = \{P_1 = (0, 27), P_2 = (0, 0), P_3 = (20, 45), P_4 = (30, 0), P_5 = (52, 0), P_6 = (78, 0)\}$ , изображенная на рис. 1.

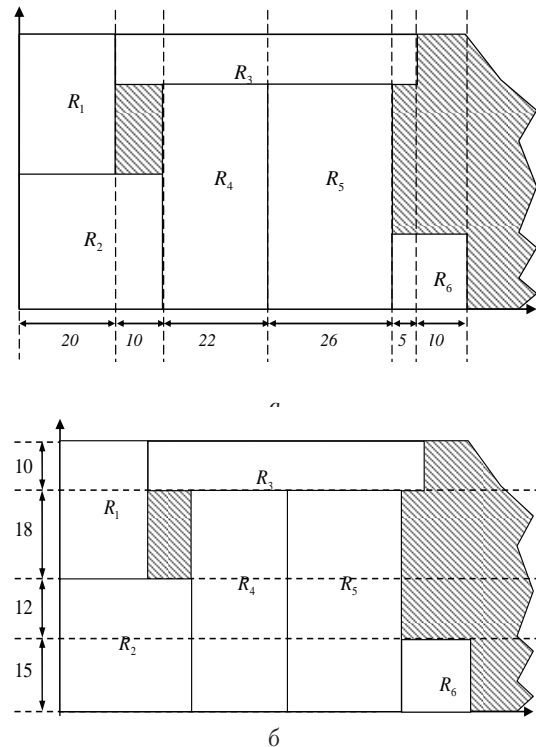


Рис. 1. Разбиение упаковки: а — вертикальная, б — горизонтальная

## 2.2. Матричное представление упаковки

Вертикальному разбиению упаковки на полосы сопоставим следующую пару:

1) Матрицу  $A^x$  размерности  $m \times k_x$  по праву:

$$a_{ij}^x = \begin{cases} 1, & \text{если } i\text{-й прямоугольник пересекает} \\ & j\text{-й вертикальной полосой,} \\ 0, & \text{в противном случае.} \end{cases}$$

2) Вектор  $z^x = (z_1^x, z_2^x, \dots, z_{k_x}^x)$ , элементами которого являются значения длин вертикальных полос.

Аналогично горизонтальному разбиению сопоставим матрицу  $A^y$  размерности  $m \times k_y$  и вектор  $z^y = (z_1^y, z_2^y, \dots, z_{k_y}^y)$ . Для упаковки  $P$ , представленной в примере 1, имеем:

$$A^x = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

$$z^x = (20, 10, 22, 26, 5, 10),$$

$$A^y = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix},$$

$$z^y = (5, 12, 18, 10).$$

Полученные указанным способом матрицы  $A^x$  и  $A^y$  будем называть *матрицами упаковки*, а векторы  $z^x$  и  $z^y$  — *векторами упаковки*.

## 2.3. Свойства матриц и векторов упаковки

В силу специфики своего построения, матрицы  $A^x$ ,  $A^y$  и векторы  $z^x$  и  $z^y$  обладают следующими свойствами:

**Свойство 1. Ограниченность.** Количество столбцов в матрицах  $A^x$  и  $A^y$  не превосходит  $m$  число прямоугольников в  $R$ .

$$k_x \leq m, k_y \leq m. \quad (3)$$

**Свойство 2. Продолженность единиц.** Для каждой строки  $i$ , ( $1 \leq i \leq m$ ) матрицы

$A^x$  найдутся такие столбцы  $j_{begin}^i$  и  $j_{end}^i$ , что

$$a_{ij}^x = \begin{cases} 1, & \text{если } j_{begin}^i \leq j \leq j_{end}^i, \\ 0, & \text{в противном случае.} \end{cases} \quad (4)$$

**Свойство 3. Положительность.** Все координаты векторов упаковки  $z^x$  и  $z^y$  — неотрицательные.

**Свойство 4.** Связь между компонентами векторов  $z^x$ ,  $z^y$  и элементами матриц  $A^x$ ,  $A^y$  с размерами прямоугольников  $R$  устанавливается следующим соотношением:

$$\sum_{i=1}^{k_x} a_{ij}^x z_i^x = l_j, \quad \sum_{i=1}^{k_y} a_{ij}^y z_i^y = w_j, \quad j = \overline{1, m} \quad (5)$$

**Свойство 5.** Пусть  $W$  — ширина полосы  $S$ , тогда связь между компонентами векторов  $z^x$ ,  $z^y$  и границами  $S$  выражается формулой:

$$\sum_{i=1}^{k_y} z_i^y \leq W \quad (6)$$

**Свойство 6. Не перекрытие прямоугольников.** Если в некотором столбце  $k_0$  матрицы  $A^x$  имеются строки  $i_1$  и  $i_2$  такие, что  $a_{i_1 k_0}^x = 1$  и  $a_{i_2 k_0}^x = 1$ , то для любого столбца  $1 \leq k \leq k_y$  матрицы  $A^y$  верно:  $a_{i_1 k}^y \cdot a_{i_2 k}^y = 0$  (т. е.  $a_{i_1 k}^y$  и  $a_{i_2 k}^y$  одновременно не могут быть равны 1).

**Утверждение 1.** Координаты  $(x_i, y_i)$  прямоугольников вычисляются по формулам

$$x_i = \sum_{k=1}^{k_i-1} z_k^x, \quad \text{где } k_i = \min(j | a_{ij}^x = 1), \quad i = \overline{1, m} \quad (7)$$

$$y_i = \sum_{k=1}^{k_i-1} z_k^y, \quad \text{где } k_i = \min(j | a_{ij}^y = 1), \quad i = \overline{1, m} \quad (8)$$

В работе [11] была доказана следующая теорема.

**Теорема.** Упаковка  $P$ , полученная по формулам (7) и (8) из матриц  $A^x$ ,  $A^y$  и векторов  $z^x$  и  $z^y$ , удовлетворяющих свойствам 1–6, является допустимой.

**Утверждение 2.** Для нахождения оптимальной упаковки достаточно найти допустимые пары  $(A^x, z^x)$  и  $(A^y, z^y)$ , такие, что

$$\sum_{i=1}^{k_x} z_i^x \rightarrow \min.$$

### 3. ОБЩИЙ МАТРИЧНЫЙ АЛГОРИТМ ПОИСКА ОПТИМАЛЬНОГО РЕШЕНИЯ 2DSP

Алгоритм последовательного улучшения допустимых матриц и векторов состоит из выполнения следующих базовых процедур:

**Шаг 1.** Нижняя граница  $L_{down} = \frac{\sum_{i=1}^m l_i w_i}{W}$ . В качестве верхней границы берется  $L_{up} = \infty$ .

**Шаг 2.** Строится пара  $(A^x, z^x)$ , удовлетворяющая свойствам 1–5 и условию  $\sum_{i=1}^{k_x} z_i^x < L_{up}$ . Если пара существует, то к Шагу 3, иначе к Шагу 4.

**Шаг 3.** Строится допустимая пара  $(A^y, z^y)$ , удовлетворяющая свойствам 1–6 и  $\sum_{i=1}^{k_y} z_i^y \leq W$ . Если пара существует, то к Шагу 4, иначе к Шагу 2.

**Шаг 4.**  $L_{up} = \sum_{i=1}^{k_x} z_i^x$ . Если  $L_{up} = L_{down}$ , то переход к Шагу 5, иначе переход к Шагу 2.

**Шаг 5.**  $L_{up}$  — оптимум.

### 4. МЕТОД ПОСТРОЕНИЯ ДОПУСТИМЫХ МАТРИЦ УПАКОВКИ

Разберем более детально метод построения допустимых пар  $(A^x, z^x)$  и  $(A^y, z^y)$ . Поскольку эта задача аналогична задаче линейного раскроя, то для ее решения можно использовать модифицированный алгоритм ветвей и границ, применяемый для поиска оптимального решения **1DCSP** [9].

Так как процедуры построения допустимых пар  $(A^x, z^x)$  и  $(A^y, z^y)$  совпадают, ограничим иллюстрацию идеи перебора только парой  $(A^x, z^x)$ .

Особенности построения пары  $(A^y, z^y)$  отметим ниже.

#### 4.1. Построение допустимой вертикальной матрицы упаковки

Введем следующие обозначения:

$PR$  — множество неразмещенных прямоугольников;

$M = \{M(1), M(2), \dots, M(k_x)\}$  — где  $M(i)$  — множество прямоугольников начинающихся в  $i$ -м блоке;

$B^{(k_t)} = \{b_1^{(k_t)}, b_2^{(k_t)}, \dots, b_m^{(k_t)}\}$  — вектор учета «остаточной длины» прямоугольников, по-

сле заполнения  $k_t$  столбцов  $b_i^{(k_t)} = l_i - \sum_{j=1}^{k_t} a_{ij} Z_j$ ,  $i = 1 \dots m$ ;

$O(j) = Z_j(W - \sum_{i=1}^m a_{ij} l_i)$  суммарный остаток после заполнения первых  $j$  столбцов.

Выполнение следующих правил позволяет отсекалть неперспективные варианты.

**Учет допустимого остатка.** Так как геометрический смысл матрицы  $A^x$  в том, что она задает расположение прямоугольников по оси  $O_x$ , условие  $\sum_{i=1}^{k_x} z_i^x < L_{up}$  эквивалентно тому, что суммарный остаток упаковки меньше, чем  $\Lambda = L_{up}W - \sum_{i=1}^m l_i w_i$ . Следовательно, для выполнения  $\sum_{i=1}^{k_x} z_i^x < L_{up}$ , необходимо, что

бы было выполнено неравенство  $\Lambda < \sum_{j=1}^{k_x} O(j)$ .

**Доминантность.** Это отсечение основано на свойстве доминантности, определенного для **1DCSP** в работе [11]. Говорят, что одна задача **1DCSP** доминирует над другой, если каждой заготовке из первой задачи можно сопоставить заготовку из второй задачи так, чтобы длины заготовок из первой задачи были не длиннее длин заготовок из второй. Доказано, что решение второй (доминируемой) задачи не может быть лучше, чем решение первой.

В процессе перебора допустимых матриц  $A^x$  после заполнения столбца  $k_t$  формируется текущее значение вектора учета «остаточной длины» прямоугольников  $B^{(k_t)}$ . Определим  $b_i^{(k_t)} = 0$ , если  $i$ -й прямоугольник полностью расположен в первых  $k_t$  вертикальных полосах;  $b_i^{(k_t)} = w_i$ , если  $i$ -й прямоугольник еще не был размещен;  $0 < b_i^{(k_t)} < w_i$ , если  $i$ -й прямоугольник пересекается с вертикальной полосой  $k_t$  и он должен присутствовать также и в  $(k_t + 1)$  полосе.

Определим число  $D_x^{(k_t)} = \sum_{j=1}^{k_t} Z_j$  — длину первых  $k_t$  блоков. Будем запоминать значения  $B^{(k_t)}$  и  $D_x^{(k_t)}$  при шаге назад, т. е. при переходе от столбца с номером  $(k_t+1)$  к столбцу с номером  $k_t$ . Пусть далее на шаге вперед мы получили новое значение  $B'^{(k_t)}$  и  $D_x'^{(k_t)}$ . Скажем, что  $B^{(k_t)}$  доминирует над  $B'^{(k_t)}$ , если  $D_x^{(k_t)} \leq D_x'^{(k_t)}$  и для всех  $i \in \{1 \dots m\}$  выпол-

нено одно из условий

$$b_i^{k_t} = 0; \quad (9)$$

$$b_i^{k_t} = b_i'^{k_t} = w_i; \quad (10)$$

$$(0 < b_i^{k_t} < w_i \text{ и } b_i^{k_t} \leq b_i'^{k_t} < w_i) \quad (11)$$

Очевидно, что если  $B^{(k_t)}$  доминирует над  $B'(k_t)$ , то не имеет смысла генерировать дальнейшие столбцы матрицы  $A^x$ , так как ожидаемый результат не может превзойти уже ранее полученного. Это следует из того, что запоминание  $B^{(k_t)}$  происходит на шаге назад и все возможные заполнения остальных столбцов уже просмотрены, а результат работы алгоритма для  $B^{(k_t)}$  не может быть лучше, так как он содержит большее число не размещенных элементов.

С учетом перечисленных выше свойств процедура построения допустимых  $(A^x, z^x)$  будет выглядеть следующим образом.

Вход:  $(W, m, R), (A^x, z^x), L_{up}$ .

Шаг 1. Рассчитывается резерв:  $\Lambda = L_{up}W - \sum_{i=1}^m l_i w_i$ ;

Шаг 2. Ищется  $k_x$  такая, что  $O(k_x - 1) \leq \Lambda < O(k_x)$ ;

Шаг 3.  $b_i^{(k_x)} = l_i - \sum_{j=1}^{k_x} a_{ij} z_j^x, i = 1..m$

Шаг 4. Ищется новое заполнение столбца  $k_x$ , которое бы отвечало условию  $\Lambda \geq O(k_x)$  и условию доминантности. Если удалось найти такое разбиение, то переход к Шагу 7, иначе — Шаг 5.

Шаг 5. Шаг назад:  $k_x = k_x - 1$ .

Шаг 6. Если  $k_x = 0$ , то **Стоп** — (матрицы не существует), иначе переход к Шагу 3.

Шаг 7.  $k_x = k_x + 1$  (шаг вперед) и заполняется  $k_x$  столбец.

Шаг 8. Если  $B^{k_x} \neq 0$ , то переход к Шагу 4.

Шаг 9. Найдено новое значение  $(A^x, z^x)$ .

#### 4.2. Особенности построения горизонтальной матрицы упаковки

После построения допустимой пары  $(A^x, z^x)$ , т.е. расположения прямоугольников по оси  $O_x$ , необходимо найти любые допустимые  $(A^y, z^y)$  с учетом свойств 1–6

и  $\sum_{i=1}^{k_y} z_i^y \leq W$ . Процесс построения  $(A^y, z^y)$  аналогичен тому, что был предложен для конструирования  $(A^x, z^x)$ . При этом для

выполнения свойства 6 необходимо ввести дополнительный список запрещенных пар. Обозначим этот список  $FR = FR(1), FR(2), \dots, FR(m)$ , где  $FR(i) = j | \exists f \in 1 \dots k_x a_{if}^x \cdot a_{jt}^x = 1$  — множество номеров прямоугольников, с которыми прямоугольник  $i$  не может присутствовать в одном горизонтальном блоке (иначе эти прямоугольники перекроются).

В дальнейшем при генерации столбцов матрицы  $A^y$  необходимо учитывать список  $FR$ : в случае размещения в столбец матрицы  $A^y$  прямоугольника  $i$  никакой другой прямоугольник из  $FR(i)$  не должен в этом столбце присутствовать. Данное ограничение существенно сокращает число допустимых  $(A^y, z^y)$ .

#### 4.3. Пример выполнения алгоритма

Возьмем исходные данные из примера 2. После выполнения процедуры начального заполнения и нахождения  $(A^x, z^x)$  найдена соответствующая пара  $(A^y, z^y)$ :

$$A^x = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix},$$

$$z^x = (28, 21, 4, 2, 19),$$

$$A^y = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix},$$

$$z^y = (20, 9, 7, 1, 12, 4).$$

Соответствующая упаковка, полученная по формулам (7)–(8) имеет вид:  $P = \{P_1 = (0, 0), P_2 = (0, 29), P_3 = (28, 0), P_4 = (28, 20), P_5 = (28, 37), P_6 = (49, 20)\}$ ,  $L_{up} = 74$ .

Следующими парами  $(A^x, z^x)$  и  $(A^y, z^y)$ , где  $\sum_{i=1}^{k_x} z_i^x < L_{up}$ , будут:



$$A^x = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix},$$

$$z^x = (25, 3, 18, 6, 4, 15),$$

$$A^y = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix},$$

$$z^y = (20, 9, 8, 8, 4, 12).$$

Полученным  $(A^x, z^x)$  и  $(A^y, z^y)$  сопоставляется упаковка  $P = \{P_1 = (0, 0), P_2 = (28, 0), P_3 = (0, 29), P_4 = (25, 29), P_5 = (25, 45), P_6 = (46, 20)\}$ .

$L_{up} = 71$  — новое значение верхней границы, и т. д.

Оптимальному решению будет соответствовать упаковка со значением  $L = 55$ , так как оно совпадает с нижней границей.

$$A_{op}^x = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$z_{op}^x = (21, 6, 1, 25, 2),$$

$$A_{op}^y = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix},$$

$$z_{op}^y = (20, 9, 8, 8, 12, 4).$$

Соответствующая упаковка  $P = \{P_1 = (0, 0), P_2 = (27, 37), P_3 = (28, 0), P_4 = (0, 29), P_5 = (0, 45), P_6 = (28, 20)\}$  изображена на рис. 2.

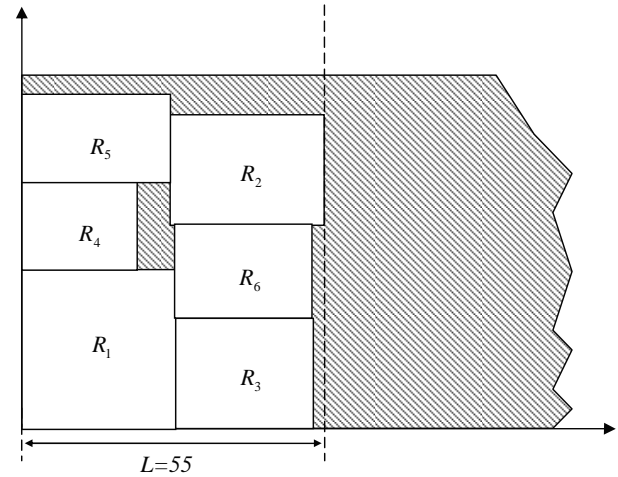


Рис. 2. Оптимальная упаковка для примера 2

## 7. ЧИСЛЕННЫЙ ЭКСПЕРИМЕНТ

Было проведено сравнение приведенного алгоритма с «методом зон» (реализация В. В. Бухваловой). Эксперимент проводился на компьютере P-IV 2000 GHz, 1 Gb на случайных тестовых задачах с параметром  $m = (12, 14, 16, 18)$ .

Для каждой группы было сгенерировано по 30 задач, время счета было ограничено 10 мин. В качестве показателя эффективности алгоритма выбран коэффициент раскроя (Cutting Coefficient, CC), равный отношению суммарной площади прямоугольников к площади занятой части полосы.

Как видно из табл. 1, матричный алгоритм за существующее временное ограничение находил лучшие упаковки (даже в случаях, когда оптимальность упаковки не удалось доказать), нежели «метод зон».

Таблица 1

Сравнение с «методом зон» (время: 10 мин)

$m$	Матричный алгоритм		«Метод зон»	
	Число задач, где был получен доказуемый оптимум	Средн. коэф. раскроя	Число задач, где был получен доказуемый оптимум	Средн. коэф. раскроя
12	30	0,85	30	0,85
14	30	0,92	3	0,92
16	24	0,88	3	0,76
18	12	0,89	0	0,73

## 7. ЗАКЛЮЧЕНИЕ

Приведенный в статье матричный подход к решению 2BDPP имеет существенные отличия от известных ранее методов решения этой задачи.

Во-первых, алгоритм «работает» с бинарными матрицами определенного типа, а не с конкретными геометрическими объектами (прямоугольниками, «зонами», блок-структурами и т. д.), что является немаловажным фактом при реализации алгоритма, а также свидетельствует о его универсальности в смысле возможности применения модификации алгоритма для решения задач упаковки более сложных геометрических объектов.

Во-вторых, алгоритм сводится к решению двух модифицированных задач линейного раскроя, трудоемкость решения которых существенно меньше трудоемкости решения 2BDPP. Базируясь на приведенном матричном построении упаковки, используя методы математического программирования, можно строить более точные нижние границы, а также использовать новые виды отсечений недопустимых вариантов.

В-третьих, данный подход обобщается на случаи, когда мерность рассматриваемых объектов больше двух. Количество матриц и векторов упаковки будет равна мерности рассматриваемых прямоугольников. Например, в трехмерном случае их будет три ( $A^x, Z^x$ ), ( $A^y, Z^y$ ), ( $A^z, Z^z$ ); в четырехмерном — четыре и т. д. При этом свойства 1–5 остаются неизменными, а свойство 6 модифицируется в зависимости от рассматриваемой мерности.

## СПИСОК ЛИТЕРАТУРЫ

1. **Dyckhoff, H.** A Typology of cutting and packing problems / H. Dyckhoff // *European J. of Operational Research*. 1990. Vol. 44. P. 145–159.
2. **Wascher, G.** An improved typology of cutting and packing problems / G. Wascher, H. Haussner, H. Schumann // *Working Paper at Faculty of Economics and Management, Magdeburg* : Otto von Guericke University, 2005. № 24.
3. **Beasley, J. E.** An exact two-dimensional non-guillotine cutting tree search procedure /

J. E. Beasley // *Operational Research*. 1985. Vol. 33. P. 49–64.

4. **Martello, S.** Exact solution of two-dimensional finite bin packing problem / S. Martello, D. Vigo // *Managment Science*. 1997. Vol. 35. P. 64–68.
5. **Липовецкий, А. И.** К оптимизации свободного размещения прямоугольников / А. И. Липовецкий // *Автоматическое проектирование в машиностроении*. Минск : ИТК АН БССР, 1985. С. 80–87.
6. **Бухвалова, В. В.** Задача прямоугольного раскроя: метод зон и другие алгоритмы / В. В. Бухвалова. СПб. : СПбГУ, 2001.
7. **Fekete, S.** A Combinatorial characterization of higher dimensional orthogonal packing / S. Fekete, J. Schepers // *Mathematics of Operations Research*. 2004. Vol. 29. P. 353–368.
8. **Стоян, Ю. Г.** Математические модели и оптимизационные методы геометрического проектирования / Ю. Г. Стоян, С. В. Яковлев. Киев : Наук. думка. 1986.
9. **Новожилова, М. В.** Решение задачи поиска глобального экстремума линейной функции цели на структуре линейных неравенств : препринт / М. В. Новожилова. Харьков : ИП АН УССР, 1988.
10. **Мухачева, Э. А.** Генетический алгоритм блочной структуры в задачах двухмерной упаковки / Э. А. Мухачева, А. С. Мухачева, А. В. Чиглинец // *Информационные технологии*. 1999. № 11. С. 13–18.
11. **Картак, В. М.** Решение задачи упаковки прямоугольников в полубесконечную полосу / В. М. Картак // *Региональная шк.-конф. : сб. тр.* Уфа, 2001. Т. 1. С. 110–121.
12. **Картак, В. М.** Модифицированный метод ветвей и границ: алгоритм и численный эксперимент для задачи одномерного раскроя / В. М. Картак, Э. А. Мухачева // *Информационные технологии*. 2000. № 9. С. 15–22.

## ОБ АВТОРЕ



**Картак Вадим Михайлович**, доц. каф. выч. математики и кибернетики. Дипл. инж.-программист (УГАТУ, 1995). Канд. физ.-мат. наук по мат. моделированию (БГУ, 1999). Иссл. в обл. задач раскроя-упаковки.