Управляемые системы, ИМ СО АН СССР, 1984 г., вып. 25

УДК 519.1

# ЗАДАЧИ ОБ УПАКОВКЕ ПРЯМОУГОЛЬНИКОВ В ПОЛОСУ

(Обзор)

А.Д.Вайнштейн

## 1. Постановка задач и основные определения

Рассмотрим следующую задачу. На плоскости задана область, ограниченная двумя параллельными лучами $O_1X_1$, $O_2X_2$ и перпендикулярным к ним отрезком $O_1O_2$. В дальнейшем эту область будем называть полосой, лучи $O_1X_1$ и $O_2X_2$ – сторонами полосы, отрезок $O_1O_2$ – дном полосы, его длину $W$ – шириной полосы (рис. 1). Дно полосы для удобства условимся изображать горизонтальной линией. Задан также набор из $n$ прямоугольников, боковые стороны которых параллельны сторонам полосы, а основания параллельны дну. Каждый прямоугольник характеризуется длиной $w_i$ (длина основания) и высотой $h_i$ (длина боковой стороны).

Упаковкой набора прямоугольников в полосу будем называть множество прямоугольников, расположенных внутри полосы и не пересекающихся друг с другом. Высотой упаковки $H$ назовем максимум расстояния между точками упаковки и дном полосы (см. рис. 1). Задача состоит в построении упаковки заданного набора, имеющей минимальную высоту.

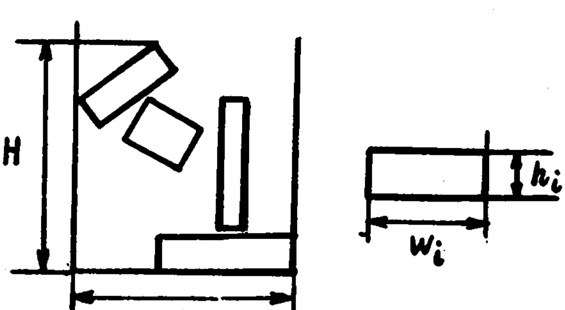



Рис. 1.

В общем случае (когда допускаются произвольные движения прямоугольников набора) задача построения упаковки минимальной высоты практически не исследовалась. Известны только примеры наборов, у которых в оптимальной упаковке углы поворота входящих в них квадратов (!) не кратны 90° (см. [31]). Оптимальная упаковка одного из таких наборов изображена на рис. 2. Отметим также работу [28], в которой исследуются асимптотические оценки для упаковок наборов из равных квадратов.

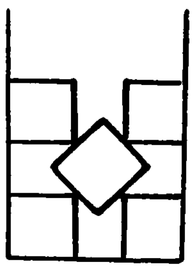



Рис. 2.

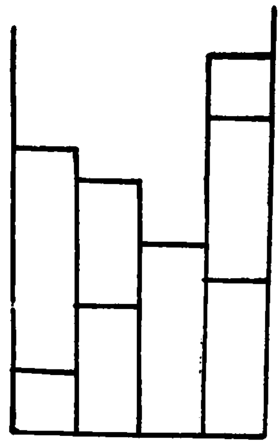


Рис. 3.

Гораздо более подробно исследованы ортогональные упаковки, в которых стороны прямоугольников остаются параллельными сторонам и дну полосы, а в особенности их частный случай, когда допускаются только параллельные переносы прямоугольников набора. В дальнейшем мы будем рассматривать только упаковки, полученные с помощью параллельных переносов.

Сформулируем несколько задач.

З а д а ч а 1. Построить упаковку заданного набора прямоугольников, имеющую минимальную высоту, пользуясь только параллельными переносами.

З а д а ч а 2. То же, что и в задаче 1, но все прямоугольники имеют одинаковую высоту (для определенности $h_i = 1$).

З а д а ч а 3. То же, что и в задаче 1, но все прямоугольники имеют одинаковую длину (для определенности $w_i = 1$), и ширина полосы кратна длине прямоугольников, $w = m \in \mathbb{Z}$.

З а д а ч а 4. То же, что и в задаче 1, но все прямоугольники – квадраты.

Сформулированные задачи (в особенности задачи 2 и 3) могут быть естественным образом проинтерпретированы в рамках теории расписаний (см., например, [7]), однако здесь мы не будем на этом останавливаться.

Задачи 2 и 3, а следовательно, и задача 1, являются $NP$-полными [4, 41, 54], поэтому отыскание эффективных алгоритмов их решения – дело практически безнадежное. В связи с этим основное внимание ниже уделяется построению приближенных алгоритмов и априорной оценке качества получаемых решений. Этому вопросу посвящен также труднодоступный обзор [34]. О других методах решения рассматриваемых задач см. [6].

Для дальнейшего изложения нам потребуются следующие определения и обозначения. Через $\mathcal{L}_j$, $j = 1, 2, 3, 4$ , будем обозначать множества всех входных данных задач 1 - 4 соответственно. Так, $\mathcal{L}_1$ - множество наборов пар $(w_i, h_i) \in R^2_+$, $\mathcal{L}_2$ - множество наборов чисел $w_i \in R_+$, $\mathcal{L}_3$ - множество наборов чисел $h_i \in R_+$, $\mathcal{L}_4$ - множество наборов чисел $x_i \in R_+$ . Через $\mathcal{L}_j(n)$ обозначим подмножество $\mathcal{L}_j$ , в которое входят наборы, содержащие не менее $n$ элементов.

Пусть $A$ - некоторый алгоритм решения задачи $j$ . Верхней оценкой алгоритма $A$ будем называть величину

$$U(A) = \sup_{L \in \mathcal{L}_j} \frac{H_A(L)}{H_{OPT}(L)},$$

где $H_A(L)$ - высота упаковки набора $L$ , полученной алгоритмом $A$ , $H_{OPT}(L)$ - высота оптимальной упаковки набора $L$ . Далее, асимптотической верхней оценкой алгоритма $A$ назовем величину

$$V(A) = \lim_{n \to \infty} \sup_{L \in \mathcal{L}_j(n)} \frac{H_A(L)}{H_{OPT}(L)}.$$

Наконец, слабую асимптотическую верхнюю оценку алгоритма $A$ определим как

$$W(A) = \lim_{n \to \infty} \sup_{L \in \mathcal{L}^0_j(n)} \frac{H_A(L)}{H_{OPT}(L)},$$

где $\mathcal{L}^0_j(n)$ - подмножество $\mathcal{L}_j(n)$ , характеризующееся соотношением

$$\lim_{n \to \infty} \sup_{L \in \mathcal{L}^0_j(n)} \frac{\max\{h_i : h_i \in L\}}{H_{OPT}(L)} = 0.$$

Слабая асимптотическая оценка определяет поведение алгоритма на множестве наборов с ограниченно изменяющейся высотой.

Пусть теперь $\mathcal{A}$ - некоторый класс алгоритмов решения задачи $j$ (в дальнейшем особое внимание будет уделяться классу РВ - алгоритмов реального времени, т.е. таких, в которых информация о размерах очередного прямоугольника становится доступной лишь после того, как будут упакованы все предыдущие прямоугольники). Нижней оценкой задачи $j$ в классе алгоритмов $\mathcal{A}$ назовем величину

$$\tilde{U}(\mathcal{A}) = \inf_{A \in \mathcal{A}} U(A).$$

Аналогично определим асимптотическую нижнюю и слабую асимптотическую нижнюю оценки:

$$\tilde{V}(\mathcal{A}) = \inf_{A \in \mathcal{A}} V(A) \; ; \qquad \tilde{W}(\mathcal{A}) = \inf_{A \in \mathcal{A}} W(A) \, .$$

Алгоритм $A \in \mathcal{A}$ назовем наилучшим в классе $\mathcal{A}$, если $\tilde{U}(\mathcal{A}) = U(A)$ асимптотически наилучшим, если $\tilde{V}(\mathcal{A}) = V(A)$, и слабо асимптотически наилучшим, если $\tilde{W}(\mathcal{A}) = W(A)$.

Далее, семейство $\{A_\varepsilon\}$ алгоритмов называется полиномиальной аппроксимационной схемой (ПАС), если

$$U(A_\varepsilon) \leq 1 + \varepsilon \, , \; \forall \varepsilon > 0 \, . \qquad (1)$$

и сложность любого алгоритма $A_\varepsilon$ есть полином от $n$ -длины исходного набора (см. [4, 32]). Аппроксимационная схема называется полностью полиномиальной (ППАС), если соотношение (1) выполняется при условии, что сложность $A_\varepsilon$ ограничена полиномом от $n$ и $1/\varepsilon$.

Понятия ПАС и ППАС естественным образом переносятся на асимптотические и слабые асимптотические оценки.

## 2. Упаковка прямоугольников одинаковой длины

Задача 3 стала первой из перечисленных задач, для которой удалось получить приближенные алгоритмы с оценкой. Простейший такой алгоритм - НЛ (нижний левый - был предложен в [36] и состоит в следующем.

Пусть $L$ - исходный список (т.е. набор прямоугольников, упорядоченный каким-либо образом). Очередной прямоугольник из $L$ размещается на самом нижнем из свободных уровней (отрезков, образованных дном полосы и верхними основаниями упакованных ранее прямоугольников) вплотную к его левой границе.

Очевидно, алгоритм НЛ работает в режиме реального времени, его сложность - $O(n)$. Следующая теорема доказана в [36] (см. также [7]).

Т е о р е м а 1. Для алгоритма НЛ решения задачи 3 справедлива оценка

$$U = 2 - 1/m \, .$$

Д о к а з а т е л ь с т в о. Пусть $H_{НЛ}(L)$ - высота упаковки списка $L$, построенной алгоритмом НЛ. Выберем произвольный прямоугольник, чье верхнее основание лежит на высоте $H_{НЛ}(L)$. Пусть $h_j$ - высота этого прямоугольника, тогда, очевидно, часть полосы от дна до уровня $H_{НЛ}(L) - h_j$ целиком заполнена прямоугольниками упаковки. Сравнивая площадь этой части упаковки с площадью всей упаковки, получаем неравенство

$$\sum_{i=1}^{n} h_i \geqslant h_j + m\,(H_{НЛ}(L) - h_j),$$

откуда

$$H_{НЛ}(L) \leqslant \frac{1}{m} \sum_{i=1}^{n} h_i + \frac{m-1}{m} h_j. \qquad (2)$$

Очевидно,

$$H_{OPT}(L) \geqslant \frac{1}{m} \sum_{i=1}^{n} h_i, \quad H_{OPT}(L) \geqslant h_j, \qquad (3)$$

откуда получаем оценку $U \leqslant 2 - 1/m$.

Неравенство $U \geqslant 2 - 1/m$ гарантирует список $L$ из $2m-1$ прямоугольников $L = \{m-1, \dots, m-1, 1, \dots, 1, m\}$, для которого

$$H_{НЛ}(L) = 2m - 1, \quad H_{OPT}(L) = m.$$

Улучшить верхнюю оценку удается путем предварительного упорядочения прямоугольников. Это приводит к некоторому увеличению сложности и выводит за пределы класса РВ. Так, в [37] рассмотре. алгоритм НЛУ (нижний левый с упорядочением по убыванию), отличающийся от НЛ тем, что список предварительно упорядочивается по убыванию высоты. Доказано, что для этого алгоритма $U = 4/3 - 1/3m$. Доказательство этой теоремы достаточно громоздко; оно может быть найдено, например, в [7].

Дальнейшее понижение верхней оценки связано с рассмотрением связи между задачами 2 и 3. Заметим, что задача 3 может быть сформулирована как задача отыскания минимальной ширины полосы, для которой оптимальная упаковка прямоугольников высоты 1 имеет высоту не более $m$ (это сразу становится ясным, если посмотреть "сбоку" на рис. 3, изображающий упаковку для задачи 3). Основываясь на этой идее, Коффман, Гэри и Джонсон предложили в [21] алгоритм, для которого удается получить оценки $1,176 \leqslant U \leqslant 1,22$. В основе этого алгоритма лежит алгоритм ППУ решения задачи 2, описанный ниже в разделе 3.

Рассмотрим теперь нижние оценки алгоритмов, работающих в режиме реального времени.

Т е о р е м а 2. Для класса РВ алгоритмов решения задачи 3 справедливы оценки

$$\tilde{U} = 3/2 \quad \text{при} \quad m = 2;$$
$$\tilde{U} = 5/3 \quad \text{при} \quad m = 3;$$
$$\tilde{U} \geqslant 1 + \sqrt{2}/2 \quad \text{при} \quad m \geqslant 4.$$

Доказательство. Пусть $m = 2$. Рассмотрим два списка: $L = \{1, 1\}$, $L' = \{1, 1, 2\}$. Пусть $A$ – произвольный алгоритм из класса РВ. Тогда, очевидно, либо $H_A(L) \geq 2$, либо $H_A(L') \geq 3$. Поскольку $H_{OPT}(L) = 1$, $H_{OPT}(L') = 2$, получаем $\tilde{U} \geq \min\left\{\frac{H_A(L)}{H_{OPT}(L)}, \frac{H_A(L')}{H_{OPT}(L')}\right\} = \frac{3}{2}$. С другой стороны, как следует из теоремы 1, алгоритм НЛ, принадлежащий классу РВ, гарантирует оценку $\tilde{U} \leq 2 - 1/2 = 3/2$. Следовательно, $\tilde{U} = 3/2$.

Пусть $m = 3$. Рассмотрим следующие четыре списка: $L_1 = \{1, 1, 1\}$, $L_2 = \{1,1,1,2,2,2\}$, $L_3 = \{1,1,1,2,2,2,7,7,7\}$, $L_4 = \{1,1,1,2,2,2,7,7,7,15\}$. Пусть $A$ – произвольный алгоритм из класса РВ. Нетрудно видеть, что либо $H_A(L_1) \geq 2$, либо $H_A(L_2) \geq 5$, либо $H_A(L_3) \geq 17$, либо $H_A(L_4) \geq 25$. Кроме того, имеем $H_{OPT}(L_1) = 1$, $H_{OPT}(L_2) = 3$, $H_{OPT}(L_3) = 10$, $H_{OPT}(L_4) = 15$, откуда $\tilde{U} \geq \min\{2, 5/3, 17/10, 25/15\} = 5/3$. Противоположное неравенство снова следует из теоремы 1.

Пусть теперь $m > 3$. Рассмотрим следующие четыре списка: $L_1 = \{1, \ldots, 1\}$, $L_2 = \{1, \ldots, 1, c, \ldots, c\}$, $L_3 = \{1, \ldots, 1, c, \ldots, c, c(c+1), \ldots, c(c+1)\}$, $L_4 = \{1, \ldots, 1, c, \ldots, c, c(c+1), \ldots, c(c+1), 2c(c+1)\}$. Здесь списки $L_1$, $L_2$, $L_3$ и $L_4$ содержат, соответственно, $m, 2m, 3m$ и $3m+1$ прямоугольников, число $c > 1$ будет определено ниже. Как и ранее, имеем следующие возможности: либо $H_A(L_1) \geq 2$, либо $H_A(L_2) \geq 1 + 2c$, либо $H_A(L_3) \geq (1+c)(1+2c)$, либо $H_A(L_4) \geq (1+c)(1+3c)$, в то время как $H_{OPT}(L_1) = 1$, $H_{OPT}(L_2) = 1 + c$, $H_{OPT}(L_3) = (1+c)^2$, $H_{OPT}(L_4) = 2c(1+c)$. Отсюда $\tilde{U} \geq \min\{2, (1+2c)/(1+c), (1+3c)/2c\}$. Наилучшая нижняя оценка здесь достигается при $(1+2c)/(1+c) = (1+3c)/2c$, т.е. при $c = 1 + \sqrt{2}$, откуда $\tilde{U} \geq 1 + \sqrt{2}/2$.

Кроме этого результата, в [2] получена также следующая асимптотическая нижняя оценка для класса РВ: $\tilde{V} \geq m^2/(m^2 - m + 1)$. Наконец, из соотношений (2) и (3) и определения слабой асимптотической оценки следует, что в классе РВ $\tilde{W} = 1$.

Предыдущие результаты могут быть усилены при некоторых фиксированных значениях $m$. В частности, для $m = 2$ в [50] предложена полностью полиномиальная аппроксимационная схема, основанная на стандартных идеях $\varepsilon$ - приближенного динамического программирования (см., например, [5]).

Из других работ отметим интересный результат [46], где построен полиномиальный алгоритм решения задачи 3 при условии, что высоты упаковываемых прямоугольников принимают заданное количество различных значений. Оценка сложности этого алгоритма $O(\log h \log m n^{2(K-1)})$, где $h = max\{h_i : h_i \in L\}$, $K$ – число различных значений, принимаемых высотами прямоугольников.

## 3. Упаковка прямоугольников одинаковой высоты

Рассмотрим теперь задачу 2, причем ширину полосы для удобства примем равной единице. Заметим, что поскольку высоты всех прямоугольников одинаковы и равны 1, имеет смысл изучать только такие алгоритмы, в которых упаковка производится по уровням, т.е. нижние основания прямоугольников располагаются на высотах 0,1,2,... . Такие алгоритмы были обстоятельно исследованы в [40, 41] (см. также [7]).

Простейший из них – алгоритм ПП (первый подходящий) – аналогичен алгоритму НЛ для задачи 3. Он заключается в том, что очередной прямоугольник из списка располагается на самом нижнем из тех уровней, на которых он может быть упакован, и при том как можно левее.

Алгоритм ЛП (лучший подходящий) отличается от ПП тем, что уровень, на котором располагается очередной прямоугольник, выбирается из условия минимума разности ширины полосы и суммарной длины упаковки в этом уровне.

Оба алгоритма – ПП и ЛП – принадлежат классу РВ. Их оценки исследовались в [33, 40, 41], где показано, что для обоих алгоритмов справедливы соотношения

$$U \le 2{,}7\,, \quad V = 1{,}7 \tag{4}$$

(заметим, что для задачи 2 понятия асимптотической и слабой асимптотической оценок совпадают, поэтому неравенства для $W$ в этом разделе не рассматриваются). Представляет интерес идея доказательства выписанных соотношений (см. также [19]).

Пусть нам удалось построить функцию $S(x)$, обладающую следующими двумя свойствами:

1) Пусть $\sum_{i=1}^{K} x_i \le 1$. Тогда $\sum_{i=1}^{K} S(x_i) \le \alpha$.

2) Пусть $\mathcal{B} = \{B_1, B_2, \dots, B_t\}$ – разбиение множества положительных чисел $X = \{x_1, \dots, x_\ell\}$, $0 < x_i \le 1$, на $t$ непересекающихся подмножеств такое, что для любых $i, j$, $1 \le i < j \le t$, и $\bar{x} \in B_j$ выполняется неравенство $\bar{x} > 1 - \sum_{x \in B_i} x$. Тогда

$$\sum_{x\in X} S(x) > t-1+\sum_{i=1}^{t} max\,\{0,\ (\sum_{x\in B_i} S(x)-1)\}.$$

В этом случае для алгоритма ПП справедливы оценки

$$V \le \alpha, \quad U \le \alpha+1. \qquad (5)$$

Действительно, оптимальная упаковка списка $L$ определяет разбиение списка на $t^* = H_{OPT}(L)$ непересекающихся подмножеств $B_1, \dots, B_{t^*}$, в каждом из которых сумма длин прямоугольников не превосходит единицы: $\sum_{w_i\in B_j} w_i \le 1$. Следовательно, из свойства (1) вытекает

$$\sum_{w_i\in L} S(w_i) \le \alpha\, H_{OPT}(L). \qquad (6)$$

С другой стороны, ПП-упаковка списка $L$ определяет разбиение списка на $t = H_{ПП}(L)$ непересекающихся подмножеств, для которых в силу описания алгоритма выполняется посылка свойства 2 . Отсюда

$$H_{ПП}(L) \le \sum_{w_i\in L} S(w_i) + 1. \qquad (7)$$

Сопоставление (6) и (7) дает оценки (5).

Нетрудно показать, что свойствами 1 и 2 обладает функция $S(x) = 2x$ ; это дает оценки $V \le 2,\ U \le 3$ . Оценку (4) удалось получить в [33] с помощью кусочно-линейной функции, определенной следующим образом:

$$S(x) = \begin{cases} 6x/5, & 0 \le x \le 1/6\ ; \\ 9x/5 - 1/10, & 1/6 \le x \le 1/3\ ; \\ 6x/5 + 1/10, & 1/3 < x \le 1/2\ ; \\ 6x/5 + 4/10, & 1/2 < x \le 1\ . \end{cases}$$

Функция $S(x)$ , определенная этими формулами, тоньше учитывает индивидуальные особенности прямоугольников, чем грубая функция $S(x) = 2x$ , что позволяет получить хорошую оценку (4). Если же учитывать эти особенности уже в процессе упаковки прямоугольников, то можно получить алгоритм класса РВ с лучшей асимптотической характеристикой. Это было сделано в [55], где предложен алгоритм УПП (улучшенный первый подходящий), заключающийся в следующем.

Прямоугольник имеет тип I, если для него $1/2 < w_i \le 1$ , тип II, если $2/5 < w_i \le 1/2$ , тип III, если $1/3 < w_i \le 2/5$ , и тип IV, если $0 < w_i \le 1/3$ . Уровням упаковки также приписывается тип, определяемый типом первого упакованного на этом уровне прямоугольника. Очередной прямоугольник типа $i$ пакуется на первом подходящем уровне типа $i$, за исключением $K$ $n$-х прямоугольников типа III (здесь $n$ = 1,2,... ; $K$ – параметр алгоритма), которые пакуются на первом подходящем уровне типа I.

В [55] доказано, что при $K$ = 6,7,8,9 для алгоритма УПП справедливы оценки $U \le 20/3$, $V = 5/3$. Кроме того, там же исследована асимптотическая нижняя оценка задачи 2 в классе РВ и доказано, что для этого класса $\tilde{V} \geq 3/2$. Эта оценка улучшена в [47], где получен следующий результат: пусть

$$m_0 = 1, \quad m_{i+1} = m_i(m_i + 1), \quad i = 0, 1, 2, \dots, \tag{8}$$

тогда для класса РВ

$$\tilde{V} \geq \sup_{K \geq 0} \Big( \sum_{i=0}^{K} \frac{i+1}{m_i} \Big) \Big/ \Big( \sum_{i=0}^{K} \frac{1}{m_i} \Big).$$

Непосредственные вычисления для $0 \le K \le 5$ показывают, что $\tilde{V} \geq 1{,}536$.

Так же, как и для задачи 3, оценки (4) можно существенно улучшить, разрешив предварительное упорядочение прямоугольников. Так, в [40,41] были рассмотрены алгоритмы ППУ (первый подходящий с упорядочением по убыванию) и ЛПУ (лучший подходящий с упорядочением по убыванию), отличающиеся от ПП и ЛП тем, что список предварительно упорядочивается по убыванию длин прямоугольников. Методами, сходными с изложенными выше, удалось показать, что для обоих указанных алгоритмов справедливы соотношения

$$U \le 5\,\frac{2}{9}, \quad V = \frac{11}{9}.$$

Эти результаты могут быть улучшены, если априори известно, что длины прямоугольников изменяются в более узких пределах. Например, для $w_i \in (0; 1/2]$ удается доказать оценку $V = 71/60$.

Принципиальная возможность получения более сильных оценок была отмечена в [55]. Там высказана следующая идея: если удается описать "плохие" входы для данного алгоритма и построить для таких входов алгоритм с лучшими характеристиками, то исходный алгоритм будет улучшен. На этом пути удалось модифицировать алгоритм ППУ так, что его асимптотическая верхняя оценка снизилась до $11/9 - \varepsilon$, где $\varepsilon \simeq 10^{-7}$.

Полученное снижение оценки настолько незначительно, что была даже высказана гипотеза о том, что в классе алгоритмов с достаточно невысокой сложностью (например, порядка $O(n)$, $O(n \log n)$) асимптотическая нижняя оценка задачи 2 отлична от единицы.

Эта гипотеза была опровергнута в [30], где для задачи 2 построена линейная (по числу прямоугольников) аппроксимационная схема. А именно доказано, что для любого $\varepsilon > 0$ существует алгоритм упаковки $S_\varepsilon$ с $V(S_\varepsilon) \le 1 + \varepsilon$, работающий в течение $C_\varepsilon + Cn \log(1/\varepsilon)$, где $C_\varepsilon$ зависит только от $\varepsilon$, $C$ – абсолютная константа. (Другая такая схема предложена в [42].)

В основе доказательства этого результата лежит следующий факт. Рассмотрим задачу 2', отличающуюся от исходной задачи 2 тем, что все прямоугольники имеют длину не меньше заданной величины $\eta$, причем длины прямоугольников принимают всего $t$ различных значений (общее число прямоугольников по-прежнему $n$). В [30] доказано, что при фиксированных $\eta$ и $t$ задача 2' может быть решена с точностью до аддитивной константы за время, не зависящее от $n$.

Далее осуществляются следующие шаги. По заданным $n$ и $\varepsilon$ определяют $\varepsilon_1 = \varepsilon/(\varepsilon+2)$, $t = \lceil \varepsilon_1^{-2} \rceil$, $\ell$ – количество прямоугольников, длина которых меньше $\varepsilon_1$, $h = \lfloor (n-\ell)/t \rfloor$. Затем исходный список $L \in \mathcal{L}_2$ разбивается на три подсписка: $K_0$, $K$ и $R$. В $K_0$ входят $\ell$ самых коротких, в $R$ входят $z = n - \ell - th$ самых длинных (имеем $0 \le z \le m - 1$), в $K$ – $th$ оставшихся прямоугольников. В свою очередь, список $K$ разбивается на $t$ подсписков $K_1, \ldots, K_t$ длины $h$ каждый, причем все прямоугольники из $K_j$ не короче любого прямоугольника из $K_i$, $1 \le i < j \le t$, и в каждом из подсписков $K_i$ выбирается прямоугольник минимальной длины $\alpha_i$. Нетрудно видеть, что вся эта процедура может быть осуществлена за время $O(n \log t) = Cn \log(1/\varepsilon)$ (с использованием алгоритма [1], отыскивающего $k$-й по величине элемент массива за линейное время).

После этого строятся списки $L'$ и $L''$: $L'$ состоит из $K_0, R, h$ прямоугольников длины $\alpha_1$, $h$ прямоугольников длины $\alpha_2, \ldots, h$ прямоугольников длины $\alpha_t$; $L''$ отличается от $L'$ тем, что $h$ прямоугольников длины $\alpha_1$ заменяются на такое же количество прямоугольников длины 1. Нетрудно убедиться, что выполняются неравенства $H_{OPT}(L') \le H_{OPT}(L) \le H_{OPT}(L'')$, $H_{OPT}(L'') \le (1+\varepsilon) H_{OPT}(L')$. Таким образом, достаточно упаковать список $L''$, что можно сделать, решив задачу 2'.

Заметим, что описанная аппроксимационная схема имеет чисто теоретическое значение, поскольку константа $C_\varepsilon$ в оценке сложности алгоритма зависит от $\varepsilon$ суперэкспоненциально. Наилучшими с практической точки зрения по-прежнему остаются алгоритмы ППУ и ЛПУ.

Из других работ отметим [10], где еще раз (уже в совершенно другой ситуации) возникает последовательность (8). Там исследуется алгоритм СПУ (следующий подходящий с упорядочением по убыванию), в котором очередной прямоугольник располагается рядом с предыдущим, а если это невозможно – то на следующем уровне. В [10] доказано, что для алгоритма СПУ справедлива оценка $$V = \sum_{i=0}^{\infty} 1/m_i \simeq 1{,}691 .$$

Другие вопросы, связанные с задачей 2 и ее модификациями, рассматриваются в [16,52].

4. Общий случай

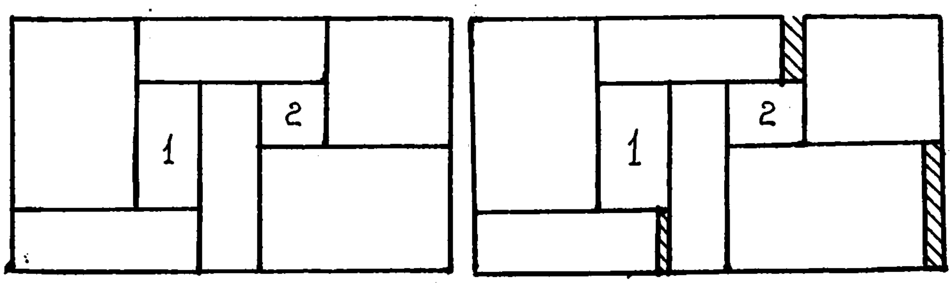
Задача об ортогональной упаковке произвольных прямоугольников (задача 1) впервые рассматривалась в [11], где изучались алгоритмы класса НЛ. В противоположность задаче 3, здесь высота НЛ-упаковки при неудачном начальном упорядочении может отличаться от оптимальной на произвольную мультипликативную константу. В частности, таким свойством обладают упорядочения по увеличению длины и по уменьшению высоты [11]. Таким образом, алгоритм НЛ не может приемлемо работать в режиме реального времени.

Более удачным оказывается упорядочение по уменьшению длины. А именно в [11] доказано, что для алгоритма НЛУД (нижний левый с упорядочением по уменьшению длины) выполняется оценка $U = 3$ . Доказательство неравенства $U \leq 3$ основано на подсчете площадей. Аналогично теореме 1 рассматривается самый высокий из прямоугольников упаковки, верхнее основание которого лежит на высоте $H_{\text{НЛУД}}(L)$ , и проводится горизонтальная прямая через его нижнее основание. Нетрудно показать, что часть полосы, лежащая ниже этой прямой, заполнена прямоугольниками упаковки не менее чем наполовину. Отсюда вытекает искомое неравенство. Обратное неравенство доказывается конструктивно.

Для задачи 4 аналогичными методами удается доказать оценку $U = 2$ .

В [17] приводится подробное описание структуры данных, обеспечивающей для алгоритма НЛУД оценку сложности $O(n^2)$ .

Как уже отмечалось ранее, в задаче 3 всегда можно найти такое упорядочение каждого конкретного списка, для которого НЛ-упаковка будет оптимальной. Для задачи 1 это оказывается неверным. Так, в [14] доказано, что в классе всех




а) б)

алгоритмов НЛ (с различным предварительным упорядочением) справедлива оценка $\tilde{U} \geq 5/4$. Идея этого результата состоит в следующем.

Рассмотрим упаковку на рис. 4 а. Очевидно, это – оптимальная упаковка заданного набора в полосу ширины 7. Более того, можно показать, что любая оптимальная упаковка этого набора в полосу такой ширины может быть получена из данной с помощью отражений относительно осей симметрии получившегося прямоугольника. Это означает, что в любой оптимальной упаковке прямоугольники 1 и 2 не попадут ни в нижний, ни в верхний ряд упаковки. Увеличим теперь длины прямоугольников 1 и 2 на $\varepsilon$ и рассмотрим упаковку полученного набора в полосу шириной $7+2\varepsilon$. Если $\varepsilon$ достаточно мало, то оптимальная упаковка модифицированного набора имеет такую же структуру (см. рис. 4 б). Но такая упаковка не может быть результатом работы НЛ-алгоритма, так как у нее в нижнем ряду обязательно будет зазор, что противоречит правилу НЛ. Легко убедиться, что для изображенного набора любой НЛ-алгоритм построит упаковку с высотой не менее 5, в то время как высота оптимальной упаковки равна 4, что и доказывает требуемую оценку.

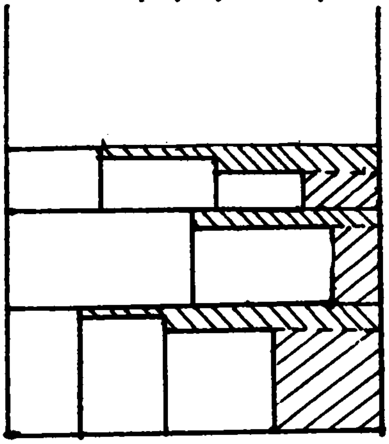


Рис. 5.

Другой подход к задаче 1 предложен в [23], где изучались так называемые уровневые алгоритмы. Рассматривались два алгоритма – СПУВ (следующий подходящий с упорядочением по убыванию высоты) и ППУВ (первый подходящий с упорядочением по убыванию высоты).

Алгоритм СПУВ заключается в следующем. Список упорядочивается по убыванию высоты прямоугольников. Первый прямоугольник пакуется в левом нижнем углу, следующий – вплотную к нему справа и т.д. до тех пор, пока очередной прямоугольник не окажется слишком длинным. В этом случае через верхнее основание первого прямоугольника проводится горизонтальная линия $\ell$ и оставшаяся часть списка пакуется аналогичным образом в полосу, дном которой служит $\ell$. Получившаяся упаковка имеет вид, изображенный на рис. 5. Часть полосы, заключенная между двумя последовательными горизонтальными линиями, называется блоком. Алгоритм ППУВ отличается от СПУВ тем, что очередной прямоугольник пакуется в самый нижний из блоков, в которые он подходит.

Справедлива следующая

Т е о р е м а 3 [23]. Для алгоритма СПУВ выполняются оценки

$$U \leq 3, \quad W = 2.$$

Д о к а з а т е л ь с т в о. Пусть $B_1, \ldots, B_t$ - блоки упаковки, $x_i$ - длина самого левого прямоугольника в $B_i$, $y_i$ - суммарная длина всех прямоугольников из $B_i$. Тогда, очевидно, $y_i + x_{i+1} > w$, $1 \le i < t$. Пусть $A_i$ - суммарная площадь всех прямоугольников из $B_i$, $H_i$ - высота $B_i$. Поскольку все прямоугольники из $B_i$ не ниже, чем $H_{i+1}$, а высота самого левого прямоугольника в $B_{i+1}$ равна $H_{i+1}$, то $A_i + A_{i+1} \ge H_{i+1}(y_i + x_{i+1}) > w H_{i+1}$. Если теперь $A = \sum_{i=1}^{t} A_i$, то

$$H_{СПУВ}(L) = \sum_{i=1}^{t} H_i \le H_1 + \frac{1}{w} \sum_{i=1}^{t-1} A_i + \frac{1}{w} \sum_{i=2}^{t} A_i \le$$

$$\le H_1 + \frac{2A}{w} \le \max_{1 \le i \le n} h_i + 2H_{ОРТ}(L).$$

Отсюда следуют неравенства $U \le 3$ и $W \le 2$. Неравенство $W \ge 2$ вытекает из следующего примера: $n = 4K$, $h_i = 1$, $i = 1, \ldots, n$, $w_{2m} = w/2$, $m = 1, \ldots, 2K$, $w_{2m-1} = w/2^n$, $m = 1, \ldots, 2K$. Здесь, очевидно, $H_{СПУВ}(L) = n$, $H_{ОРТ}(L) = n/2 + \varepsilon$, $\varepsilon \to 0$.

Заметим, что хотя асимптотически алгоритм СПУВ лучше, чем НЛУД, абсолютная оценка для него равна 3, так же, как и для НЛУД. Улучшения как асимптотической, так и абсолютной оценок позволяет добиться применение алгоритма ППУВ: в [23] доказано, что для алгоритма ППУВ справедливы соотношения $U \le 2{,}7$, $W = 1{,}7$. Доказательство этого факта проводится аналогично доказательству оценок для алгоритмов ППУ и ЛПУ в задаче 2 с использованием той же весовой функции.

Если все длины ограничены величиной $w/m$, $m \ge 2$, то сформулированный результат можно усилить [23]: выполняется оценка

$$H_{ППУВ}(L) \le \left(1 + \frac{1}{m}\right) H_{ОРТ}(L) + 1.$$

Отметим также результат для задачи 4: если все прямоугольники из списка - квадраты, то для алгоритма ППУВ $U \le 5/2$, $W = 3/2$.

Таким образом, уровневые алгоритмы превосходят по качеству НЛ-алгоритмы, хотя, казалось бы, при уровневой упаковке в полосе остается очень много неиспользованного места. Частичным объяснением этого явления может служить тот факт, что суммарная незанятая площадь, лежащая в блоках над прямоугольниками упаковки, не так уж велика:

$$S \le (H_1 - H_2)w + (H_2 - H_3)w + \dots + (H_{t-1} - H_t)w + H_t w = H_1 w.$$

Отметим, что уровневые алгоритмы, в отличие от НЛ-алгоритмов, удается приспособить к работе в режиме реального времени. Этому вопросу посвящена [12], где описаны единственные на сегодняшний день алгоритмы решения задачи 1 в режиме реального времени с конечной верхней оценкой $U$. Идея этих алгоритмов заключается в следующем. До начала работы алгоритма фиксируется параметр $\tau$, $0 < \tau < 1$. Высотой блока может стать только число вида $\tau^i$, $i \in \mathbb{Z}$. Для очередного прямоугольника определяется целое (возможно, отрицательное) число $K$ такое, что $\tau^{K+1} < h \le \tau^K$. Алгоритм СП$_\tau$ упаковывает этот прямоугольник как можно левее в самом верхнем из блоков высоты $\tau^K$ (если упаковка ни в один из таких блоков невозможна, организуется новый блок высоты $\tau^K$). Алгоритм ПП$_\tau$ отличается от него лишь тем, что упаковка производится в самый нижний из блоков высоты $\tau^K$. С помощью сравнения площадей для алгоритма СП$_\tau$ легко получить оценки $W = 2/\tau$, $U = 2/\tau + 1/\tau(1-\tau)$; тем самым $U \simeq 7,36$ при соответствующем $\tau$. Для алгоритма ПП$_\tau$ с помощью весовой функции в [12] получены более сильные оценки $W = 1,7/\tau$, $U = 1,7/\tau + 1/\tau(1-\tau)$, откуда $U \simeq 6,99$. В применении к задаче 4 оба описанных алгоритма дают оценку $U = 1/\tau + 1/\tau(1-\tau)$.

Приведенные выше верхние оценки алгоритмов из класса РВ пока еще очень далеки от нижней оценки $\tilde{U} \ge 2$, доказанной в [15]. Кроме этого результата, в [15] получены также интересные нижние оценки для классов алгоритмов с различным предварительным упорядочением и одним просмотром списка. Эти алгоритмы работают так же, как алгоритмы из класса РВ, но на входе рассматриваются лишь списки, упорядоченные в соответствии с заданным правилом. В [15] для упорядочения по убыванию ширины получена нижняя оценка $\tilde{U} \ge 1 + \sqrt{6}/3$, для упорядочения по убыванию высоты - $\tilde{U} \ge 5/3$, для упорядочений по возрастанию ширины либо высоты - $\tilde{U} \ge (1+\sqrt{7})/2$. Для задачи 4 в случае упорядочения по убыванию размера - $\tilde{U} \ge 3/2$, в случае упорядочения по возрастанию размера - $\tilde{U} \ge 7/4$.

Третий метод решения рассматриваемой задачи - метод расщепления полосы - был впервые предложен в [35]. Для осуществления алгоритма расщепления (Р) прямоугольники предварительно упорядочиваются по убыванию длины. Упаковка начинается с нижнего левого угла полосы. Если очередной прямоугольник удается упаковать вплотную с каким-либо из предыдущих на одном горизонтальном уровне, то полоса расщепляется на две вертикальной прямой, продолжающей общую границу этих прямоугольников. После этого упаковка происходит в пределах этих полос, которые в свою очередь могут быть расщеплены, и т.д.

В [35] доказано, что для алгоритма Р справедливы оценки $U \le 3$, $W = 2$. Доказательство этого результата основано на подсчете площади, занимаемой упаковкой в текущий момент работы алгоритма. Заметим, что в задаче 4 алгоритма Р можно доказать оценку $U \le 2$ (см. [35]).

Дальнейшее улучшение абсолютных и асимптотических оценок в задаче 1 связано с применением смешанных алгоритмов, использующих несколько стратегий упаковки в зависимости от размеров прямоугольников.

Так, лучшая абсолютная оценка, известная на сегодняшний день (январь 1984 г.), получается комбинацией идей расщепления полосы и уровневой упаковки. Рассмотрим следующий ал оритм УР (уровневый с расщеплением), предложенный в [53]. Выберем прямоугольники с длиной не менее $W/2$ и упакуем их один на другой. Пусть $h_0$ – высота этой упаковки. Упорядочим оставшиеся прямоугольники по убыванию высоты и уложим алгоритмом СПУВ один блок на высоте $h_0$. Если список еще не исчерпан, разбиваем полосу вертикальной прямой на две полосы равной ширины. В каждой из этих полос определим текущий уровень – по самой высокой точке упаковки, расположенной в этой полосе. Оставшиеся прямоугольники пакуются алгоритмом СПУВ по блокам в ту из полос, в которой очередной текущий уровень (к моменту начала нового блока) окажется ниже.

Т е о р е м а 4. Для алгоритма УР справедливы оценки $U \le 5/2$, $W = 2$.

Д о к а з а т е л ь с т в о. Предположим для удобства, что ширина полосы равна единице. Пусть $A_0$ – площадь упаковки ниже уровня $h_0$, $A_1$ – площадь прямоугольников, попавших в левую полосу, нижние основания которых расположены на уровне $h_0$, $A_2$ – площадь всех остальных прямоугольников, $h_1$ – высота самого нижнего блока в левой полосе, $h_2$ – высота самого нижнего блока в правой полосе, $H_2$ – высота упаковки прямоугольников, составляющих $A_2$, $h$ – разность высот упаковки в правой и левой полосах к концу работы алгоритма. Тогда, очевидно, $2H_{УР}(L) = 2h_0 + h_1 + H_2 + h$. Аналогично доказательству теоремы 3 можно получить $H_2 \le 4A_2 + h_2$. Далее, $h_0 \le 2A_0$, $h_2 \le 2A_1$. Отсюда

$$H_{УР}(L) \le \frac{1}{2}\left(4A_0 + 4A_2 + h_2 + h + h_1\right) = 2(A_0 + A_1 + A_2) + \frac{h + h_1 - h_2}{2}. \quad (9)$$

Очевидно, $A_0 + A_1 + A_2 \le H_{OPT}(L)$. Далее, если во время работы алгоритма текущий уровень левой полосы всегда превосходил текущий уровень правой, то $H_{УР}(L) = h_0 + h_1 \le 2H_{OPT}(L)$. В противном случае $h \le h_2$, и из (9) и $h_1 \le max\{h_i : h_i \in L\} \le H_{OPT}(L)$ получаем утверждение теоремы.

В алгоритме, предложенном в [23], исходный список разбивается на две части: в $L_1$ входят прямоугольники с длиной более $W/2$, в $L_2$ – остальные. Затем список $L_1$ пакуется с помощью алгоритма СПУВ. Блоки полученной упаковки переупорядочиваются по возрастанию длины. В результате этого в правом нижнем углу полосы образуется незаполненная область, в которую вписывается прямоугольник длины $W/3$. Список $L_2$ пакуется с помощью СПУВ сначала в этот прямоугольник, а потом выше упаковки списка $L_1$. Как доказано в [23], для описанного алгоритма справедливы оценки $U \le 7/2$, $W = 3/2$.

В смешанном алгоритме из [35] исходный список разбивается на пять частей: $L_1 = \{(h_i, w_i) \in L : w_i > w/2\}$, $L_2 = \{(h_i, w_i) \in L : w/3 < w_i \le w/2\}$, $L_3 = \{(h_i, w_i) \in L : w/4 < w_i \le w/3\}$, $L_4 = (h_i, w_i) \in L : 5w/24 < w_i \le w/4$, $L_5 = \{(h_i, w_i) \in L : w_i \le 5w/24\}$. Прямоугольники из $L_1$ пакуются один над другим у левой стороны полосы. В оставшуюся незаполненной часть полосы, примыкающую к правой стороне, вписываются на соответствующих высотах прямоугольники длины $w/4$, $w/3$, $5w/12$. Подсписки $L_2$, $L_3$, $L_4$, $L_5$ пакуются в эти прямоугольники алгоритмами ППУВ и НПУД (нижний правый с упорядочением по убыванию длины). В [35] доказано, что $U \le 151/18$, $W = 4/3$.

В алгоритме работы [8] исходный список также разбивается на пять частей с учетом длины входящих в него прямоугольников. Часть прямоугольников пакуется с помощью алгоритмов НЛУД или НПУД, либо просто один над другим, в области, примыкающие к левой и правой сторонам полосы. Оставшиеся прямоугольники пакуются в область сложной конфигурации, расположенную между правой и левой упаковками, для чего разработана специальная модификация алгоритма СПУВ. В [8] сконструирована весовая функция, с помощью которой удается показать, что для описанного алгоритма справедливы оценки $U \le 63/8$, $W = 5/4$. Таким образом, этот алгоритм имеет наилучшую слабую асимптотическую оценку из всех, известных на сегодняшний день.

Отметим, что абсолютные оценки трех последних алгоритмов сравнительно высоки и, по-видимому, могут быть улучшены.

## 5. Другие результаты

Остановимся вкратце на нескольких задачах, примыкающих в той или иной степени к изложенным выше.

1) Многомерные упаковки. Естественным обобщением рассмотренных задач является следующая: упаковать заданный набор $d$-мерных параллелепипедов в $d$-мерную "полосу" - фигуру, представляющую собой прямое произведение $(d-1)$-мерного параллелепипеда на ортогональный к нему луч. Такая задача рассматривалась в [13,18,30,34,55] для случая параллелепипедов одинаковой высоты (под высотой здесь подразумевается то измерение, в направлении которого полоса бесконечна). Были исследованы абсолютные и асимптотические оценки для $d$-мерных аналогов алгоритмов ПП и ППУ. Так, в [34] построен алгоритм с оценкой $V \le d + 7/10$. Для других алгоритмов также наблюдается линейная (с коэффициентом единица) зависимость оценки от размерности пространства. Тот же самый эффект возникает и при попытке построения ПАС: в $d$-мер-

ном пространстве можно лишь гарантировать для любого $\varepsilon > 0$ наличие линейного алгоритма с оценкой $V \leq d + \varepsilon$ (см. [30]).

Несколько лучше обстоят дела для $d = 3$. В этом случае имеется алгоритм решения указанной задачи, для которого $91/45 \leq V \leq 17/8$ (см. [18]). Наконец, отметим работу [13], где предложен точный алгоритм для случая, когда все размеры параллелепипедов кратны $1/p$, где $p$ - фиксированное целое число.

2) Задача максимизации числа упакованных прямоугольников. Следующая задача тесно связана с рассматривавшимися выше: упаковать как можно больше прямоугольников из заданного списка в фиксированный прямоугольник (т.е. в полосу, ограниченную также и сверху). В [24-26] эта задача изучалась для случая прямоугольников одинаковой высоты. При этом естественно считать, что список прямоугольников упорядочен по неубыванию длины. Поскольку рассматриваемая задача, так же, как и все предыдущие, является $NP$-полной, для исследования качества приближенных алгоритмов ее решения используются оценки, аналогичные введенным ранее, с тем исключением, что число прямоугольников в оптимальной упаковке стоит теперь в числителе, а число прямоугольников в упаковке, полученной данным алгоритмом, - в знаменателе (это связано с тем, что рассматривается задача на максимум).

В упомянутых работах исследован класс префиксных алгоритмов, т.е. таких, в которых упаковка производится в порядке возрастания длины и прекращается, как только появляется прямоугольник, упаковать который невозможно. Доказано, что для всех алгоритмов этого класса $U \leq 2 - 1/m$, где $m$ - высота прямоугольника, в который ведется упаковка (высота прямоугольников из списка полагается равной 1). Далее, исследуется алгоритм ППВ (первый подходящий с упорядочением по возрастанию), для которого доказана оценка $U = 4/3$. Наконец, предложена следующая модификация алгоритма ППУ. Из упорядоченного по неубыванию длины списка выбрать подряд максимальный по включению подсписок, суммарная площадь которого не превосходит площади прямоугольника, в который ведется упаковка. Переупорядочить выбранный подсписок по невозрастанию длины и паковать в полосу алгоритмом ППУ. Если высота полученной упаковки не превосходит $m$, то имеем оптимальную упаковку для исходной задачи. В противном случае необходимо удалить из подсписка самый длинный прямоугольник и снова применить алгоритм ППУ. В [24-26] доказано, что для этого алгоритма $8/7 \leq V \leq 7/6$.

Аналогичная задача для списка, состоящего из квадратов, исследована в [9]. Там рассмотрен алгоритм СПВ (следующий подходящий с упорядочением по возрастанию размера) и доказано, что для него $V = 4/3$.

3) Задачи упаковки с дополнительными условиями. Отметим в первую очередь интересную работу [22], в которой рассматривается динамический вариант задачи 2. А именно, для каждого прямоугольника задаются числа $d < b \leq 0$,

так что в момент времени $b$ прямоугольник должен быть упакован, а в момент времени $d$ – изъят из упаковки. Таким образом, высота упаковки заданного списка становится функцией времени, и задача заключается в том, чтобы минимизировать максимум этой функции. В [22] изучаются верхние и нижние оценки для этой задачи в режиме реального времени.

Другой вид дополнительных условий возникает, когда прямоугольники считаются окрашенными в несколько цветов и накладываются ограничения на число прямоугольников одного цвета, находящихся на одном уровне (см. [7,49]).

4) Вероятностный анализ. Выше был описан только один из подходов к исследованию приближенных алгоритмов решения $NP$-полных задач – априорные оценки поведения в худшем случае. Другим таким подходом является вероятностный анализ – оценка поведения в большинстве случаев. Этому направлению применительно к рассматриваемым задачам посвящены работы [19,20,27,29,38,45, 51]. Из полученных там результатов отметим доказательство оптимальности почти всегда алгоритмов ППУ и ЛПУ. Доказательства подобных фактов проводятся по следующей схеме:

а) строится некоторый достаточно простой алгоритм и доказывается, что он на всех списках работает не лучше, чем интересующий нас алгоритм (например, ППУ);

б) доказывается, что построенный алгоритм почти всегда оптимален. Недостаток вероятностного подхода заключается в отсутствии естественного распределения вероятностей на множестве всех списков.

5) Другие задачи, связанные с упаковкой прямоугольников в полосу, обсуждаются также в [3, 39, 43. 44. 48].

б) доказывается, что построенный алгоритм почти всегда оптимален. Недостаток вероятностного подхода заключается в отсутствии естественного распределения вероятностей на множестве всех списков.

5) Другие задачи, связанные с упаковкой прямоугольников в полосу, обсуждаются также в [3,39,43,44,48] .

Поступила в ред.-изд.отдел
11 апреля 1984 г.

## Литература

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979.

2. Вайнштейн А.Д. Нижние оценки для задачи упаковки прямоугольников в полосу в режиме реального времени. – В кн.: Теория и методы автоматизации проектирования сложных систем и автоматизации научных исследований. – Минск, 1984.

3. Вайнштейн А.Д., Кадушин А.И. Метод устранения фрагментации памяти при детерминированной пакетной обработке. – Кибернетика, 1984, № 3.

4. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи.– М.: Мир, 1982.

5. Левнер Е.В., Генс Г.В. Дискретные оптимизационные задачи и эффективные приближенные алгоритмы. – М.: ЦЭМИ, 1978.

6. Стоян Ю.Г., Соколовский В.З. Решение некоторых многоэкстремальных задач методом сужающихся окрестностей. – Киев: Наукова думка, 1980.

7. Теория расписаний и вычислительные машины/ под ред. Э.Г.Кофмана. - М.: Наука, 1984.

8. Baker B., Brown D., Katseff H. A 5/4 algorithm for two-dimensional packing. - J. of Algorithms, 1981, v.2, N 4.

9. Baker B., Calderbank A., Coffman E., Lagarias J. Approximation algorithms for maximizing the number of squares packed into a rectangle. - SIAM J. Algebr. and Discr. Meth., 1983, v.4, N 3.

10. Baker B., Coffman E. A tight asymptotic bound for next-fitdecreasing bin-packing. - SIAM J. Algebr. and Discr. Meth., 1981, v.2, N 2.

11. Baker B., Coffman E., Rivest R. Orthogonal packings in two dimensions. - SIAM J. Comput., 1980, v.9, N 4.

12. Baker B., Schwarz J. Shelf algorithms for two-dimensional packing problems. - SIAM J. Comput., 1983, v. 12, N 3.

13. Blazewicz J., Ecker K. A linear time algorithm for restricted bin packing and scheduling problems. - Oper. Res. Lett., 1983, v.2, N 2.

14. Brown D. An improved BL lower bound. - Inform. Process. Lett., 1980, v. 11, N 1.

15. Brown D., Baker B., Katseff H. Lower bounds for on-line twodimensional packing algorithms. - Acta Informatica, 1982, v.18, N 2.

16. Chandra A., Hirschberg D., Wong C. Bin packing with geometric constraints in computer network design. - Oper. Res., 1978, v.26, N 5.

17. Chazelle B. The bottom-left bin-packing heuristic: an efficient implementation. - IEEE Trans. on Computers, 1983, v.32, N 8.

18. Chung F., Garey M., Johnson D. On packing two-dimensional bins.- SIAM J. Algebr. and Discr. Meth., 1982, v.3, N 1.

19. Coffman E. An introduction to proof techniques for bin-packing approximation algorithms. - In: Deterministic and stochastic scheduling/ ed. M.Dempster, J.Lenstra, A.Rinnooy Kan. - Dordrecht, 1982.

20. Coffman E., Frederickson G., Lueker G. Probabilistic analysis of the LPT processor scheduling heuristic. - In: Deterministic and stochastic scheduling/ ed. M.Dempster, J.Lenstra, A.Rinnooy Kan. Dordrecht, 1982.

21. Coffman E., Garey M., Johnson D. An application of bin-packing to multiprocessor scheduling. - SIAM J. Comput., 1978, v.7, N 1.

22. Coffman E., Garey M., Johnson D. Dynamic bin Packing.- SIAM J. Comput., 1983, v.12, N 2.

23. Coffman E., Garey M., Johnson D., Tarjan R. Performance bounds for level-oriented two-dimensional packing algorithms. - SIAM J. Comput., 1980, v.9, N 4.

24. Coffman E., Leung J. Combinatorial analysis of an efficient algorithm for processor and storage allocation. - SIAM J. Comput., 1979, v.8, N 2.

25. Coffman E., Leung J., Ting D. Bin-packing problems and their applications in storage and processor allocation. - In: Computer performance/ ed. K.Chandy, M.Reiser. North-Holland, 1977.

26. Coffman E., Leung J., Ting D. Bin-packing: maximizing the number of pieces packed. - Acta Informatica, 1978, v.9, N 3.

27. Coffman E., So K., Hoffri M., Yao A. A stochastic model of bin-packing. - Inform. and Control, 1980, v.44, N 2.

28. Erdos P., Graham R. On packing squares with equal squares.- J. Combin. Theory, ser.A, 1975, v.19, N 1.

29. Frederickson G. Probabilistic analysis for simple one- and two-dimensional bin packing algorithms. - Inform. Process.Lett.,1980, v. 11, N 4-5.

30. Fernandez de la Vega W., Lueker G. Bin packing can be solved within 1+ε in linear time. - Combinatorica, 1981, v.1, N 4.

31. Gardner M. Some packing problems that cannot be solved by sitting on the suitcase. - Sci. Amer., 1979, v.241, N 4.

32. Garey M., Graham R., Johnson D. Performance guarantees for scheduling algorithms. - Oper. Res., 1978, v.26, N 1.

33. Garey M., Johnson D., Graham R., Yao A. Resource constrained scheduling as generalised bin packing. - J. Combin. Theory, ser. A, 1976, v.21, N 3.

34. Garey M., Johnson D. Approximation algorithms for bin-packing. - In: Analysis and design of algorithms in combinatorial optimization/ ed. G.Ausiello, M.Lucertini. New-York, 1981.

35. Golan I. Performance bounds for orthogonal oriented two-dimensional packing algorithms. - SIAM J. Comput., 1981, v.10, N 3.

36. Graham R. Bounds for certain multiprocessing anomalies. - Bell Syst., Tech., J., 1966, v.45, N 9.

37. Graham R. Bounds for multiprocessor timing anomalies.- SIAM J. Appl. Math., 1969, v.17, N 2.

38 Hoffman U. A class of simple stochastic online bin packing algorithms. - Computing, 1982, v.29, N 3.

39. Holton D., Richard J. Brick packing. - Lect. Notes Math., 1978, v.686.

40. Johnson D. Fast algorithms for bin packing.- J. Comput. and Syst. Sci., 1974, v.8, N 3.

41. Johnson D., Demers A., Ullman J., Garey M., Graham R. Worst-case performance bounds for simple one- dimensional packing algorithms.- SIAM J. Comput., 1974, v.3, N 4.

42. Karp R., Karmarkar N. An efficient approximation scheme for the one-dimensional bin packing problem.- In: Proc. 23-rd Ann. Symp. on Foundations of Comput. Sci., 1982.

43. Kleitman D., Krieger M. Packing squares in rectangles, I. - Ann. of the New-York Acad. Sci., 1970, v.175, Art.1.

44. Kleitman D., Krieger M. An optimal bound for two-dimensional bin packing.- In: Proc. 16th Ann. Symp. on Foundations of Comput.Sci., 1975.

45. Knodel W. A bin packing algorithm with complexity O (nlogn) and performance 1 in the stochactic limit.- Lect. Notes Comput. Sci., 1981, v.118.

46. Leung G. On scheduling independent tasks with restricted execution times.- Oper. Res., 1982, v.30, N 1.

47. Liang F. A lower bound for on-line bin packing. - Inform. Process. Lett., 1980, v.10, N 2.

48. Meir A., Moser L. On packing of squares and cubes. - J. Combin. Theory, 1968, v.5, N 2.

49. Morihara I., Ibaraki T., Hasegawa T. Bin packing and multiprocessor scheduling problems with side constraints on job types. - Discr. Appl. Math., 1983, v.6, N 2.

50. Sahni S. Algorithms for scheduling independent tasks.- J. of ACM, 1976, v.23, N 1.

51. Shapiro S. Preformance of heuristic bin packing algorithms with segments of random length.- Inform. and Control, 1977, v.35, N 2.

52. Shearer J. A counterexample to a bin packing conjecture. - SIAM J. Algebr. and Discr. Meth., 1981, v.2, N 3.

53. Sleator D. A 2.5 times optimal algorithm for packing in two dimensions.- Inform., Process. Lett., 1980, v.10, N 1.

54. Ullman J. Polynomial complete scheduling problems.- In: Proc. 4th Symp. on Operating Systems Principles, 1973.

55. Yao A. New algorithms for bin packing.- J. of ACM, 1980, v.27, N 2.