С. Н. Жук, Приближенные алгоритмы упаковки прямоугольников в несколько полос, *Дискрет. матем.*, 2006, том 18, выпуск 1, 91–105

DOI: 10.4213/dm34

Дискретная математика

том 18 выпуск 1 * 2006

УДК 519.7

Приближенные алгоритмы упаковки прямоугольников в несколько полос

© 2006 г. С. Н. Жук

Рассматривается задача упаковки прямоугольников в несколько полубесконечных полос определенной ширины. В работе предложены два достаточно просто реализуемых алгоритма, в которых прямоугольники размещаются по мере поступления. Показано, что точность первого алгоритма не аппроксимируется никакой абсолютной постоянной. Второй алгоритм гарантирует константную мультипликативную точность, причем доказанная оценка мультипликативной точности неулучшаема.

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 05–01–00798.

1. Введение

Задача упаковки прямоугольников в несколько полос заключается в следующем: имеется несколько полубесконечных полос определенной ширины и набор прямоугольников. Требуется найти ортогональное размещение этих прямоугольников по полосам (без пересечений и вращений), имеющее минимальную высоту.

Такая задача возникает, например, при распределении ресурсов в глобальных вычислительных сетях Grid [7], объединяющих разнородные вычислительные ресурсы (персональные компьютеры, рабочие станции, кластеры, суперкомпьютеры) и используемых для выполнения различных вычислительных приложений. Задачу о распределении ресурсов в Grid можно представить как задачу о размещении задач на группе кластеров. Такая модель имеет простую геометрическую интерпретацию. Кластер можно представить как бесконечную в одну сторону полосу определенной ширины (ширина характеризует число процессоров кластера, другое измерение соответствует времени), а задачу — как прямоугольник с высотой и шириной, равными времени выполнения и требуемому числу процессоров соответственно. Тогда естественно возникает задача упаковки данного множества задач (прямоугольников) в несколько полос (кластеров): требование, что прямоугольники не должны пересекаться означает, что каждая задача на некоторое время (равное высоте прямоугольника) занимает некоторое число процессоров (равное ширине прямоугольника).

Даже для случая одной полосы, рассматриваемая задача является NP-трудной. Например, к ней сводится одномерная задача об упаковке в контейнеры. Если высоты всех прямоугольников равны 1, то слои, которые будут получаться при их размещении, как раз и соответствуют этим контейнерам. Следовательно, вряд ли можно ожидать, что существует эффективный алгоритм, позволяющий решить эту задачу точно. Поэтому рассматривают различные приближенные алгоритмы, которые позволяют достаточно быстро найти некоторое решение с определенной точностью.

Наиболее исследованы частные случаи задачи. Один из них — уже упомянутая классическая задача об упаковке в контейнеры [8]. Другой частный случай, когда имеется всего одна полоса, — также известная и хорошо исследованная задача [1, 2, 3]. Для нее разработаны различные приближенные алгоритмы, в частности, построен алгоритм, позволяющий за полиномиальное время находить почти оптимальное решение (с точностью $1+\varepsilon$ для любого $\varepsilon > 0$) [2]. Тем не менее, этот алгоритм достаточно сложен и на практике обычно применяют более простые эвристики.

Как обычно, считаем, что алгоритм гарантирует точность r (мультипликативную), если каждое допустимое решение, построенное с помощью этого алгоритма, отличается от оптимального решения не более, чем в r раз (по значению целевой функции).

В данной работе рассмотрены два достаточно просто реализуемых алгоритма, в которых прямоугольники по полосам размещаются по мере поступления (то есть в режиме on-line). Для первого алгоритма показано, что его точность не аппроксимируется никакой абсолютной константой. А для второго (модификации первого) доказано, что он гарантирует константную мультипликативную точность. Это достигнуто за счет правильной формализации понятии допустимая полоса для прямоугольника. Показано также, что для второго алгоритма полученную оценку мультипликативной точности улучшить нельзя.

2. Постановка задачи

Сформулируем основную задачу. Входом для нее является конечная последовательность прямоугольников

$$T = \{T_1, \dots, T_n\},$$

высота и ширина прямоугольника T_j равны соответственно $h(T_j)$ и $w(T_j)$, $j = 1, \dots, n$, и множество полубесконечных полос

$$C = \{C_1, \dots, C_m\},$$

ширина i-й полосы равна w_i, $i = 1, \dots, m$. Требуется найти ортогональное размещение последовательности прямоугольников T по этим полосам (без пересечений и вращений, стороны прямоугольников параллельны сторонам полос), минимизирующее полную высоту этого размещения, то есть максимум по всем прямоугольникам и по всем полосам расстояния от дна полосы до верхней грани прямоугольника.

Пусть $H_O(T, C)$ — оптимальное значение высоты размещения последовательности прямоугольников T на полосах C и $H_A(T, C)$ — высота размещения этих прямоугольников на C, получающегося при использовании алгоритма A.

Предложение 1. *Если* $\mathrm{P} \neq \mathrm{NP}$, *то для данной задачи не существует полиномиального алгоритма, который позволяет получить приближенное решение с абсолютной мультипликативной точностью, меньшей* 2.

Доказательство. Рассмотрим следующую задачу о разбиении: имеется n неотрицательных чисел $a_1, \dots, a_n$. Существует ли разбиение этих чисел на два множества, имеющих одинаковую сумму элементов?

Известно, что эта задача является NP-полной. Рассмотрим теперь задачу о прямоугольниках и полосах для двух полос ширины $\frac{1}{2}\sum_{i=1}^{n} a_i$ и n прямоугольников таких, что

$$w(T_j) = a_j, \quad h(T_j) = 1, \quad j = 1, \dots, n.$$

Высота оптимального размещения в этом случае может быть или 1, или 2, в зависимости от того, существует ли разбиение чисел $a_1, \dots, a_n$ на два множества с равной суммой. Ясно, что если есть полиномиальный алгоритм, позволяющий с мультипликативной точностью, меньшей 2, построить размещение прямоугольников, то с его помощью мы можем однозначно ответить на вопрос о существовании искомого разбиения. Поэтому, если P $\neq$ NP, то такого алгоритма не существует.

3. Алгоритм 1

Мы будем рассматривать следующую двухэтапную модель работы алгоритмов.

На первом этапе прямоугольники распределяются по полосам. Это распределение производится по мере поступления прямоугольников (on-line) и, впоследствии, с приходом новых прямоугольников, номер этой полосы не изменяется.

На втором этапе производится размещение прямоугольников внутри каждой полосы. Это происходит после того, как все прямоугольники уже распределены по полосам.

3.1. Алгоритм Bottom-Left

Для размещения прямоугольников внутри одной полосы в данной работе используется достаточно простая эвристика, алгоритм Bottom-Left (BL), детально исследованный в [1]. Этот алгоритм применяется к последовательности прямоугольников и размещает их в порядке следования так, что каждый следующий прямоугольник размещается в самую левую из всевозможных самых нижних допустимых позиций.

Результаты работы алгоритма BL существенно зависят от порядка следования прямоугольников. Как показано в статье [1], для произвольного порядка отношение $H_{\mathrm{BL}}(T, C_i)/H_O(T, C_i)$ может получиться сколь угодно большим. Но, если перед размещением, список прямоугольников T отсортировать по убыванию ширины, то можно показать, что для полученного списка T_1

$$\frac{H_{\mathrm{BLD}}(T_1, C_i)}{H_O(C_i)} \leqslant 3.$$

Далее при использовании алгоритма мы везде будем полагать, что прямоугольники упорядочены по убыванию ширины и эвристику BL с предварительной сортировкой по ширине будем называть BLD (Bottom-Left Decreasing) алгоритмом.

Есть еще одно важное свойство алгоритма BLD, которое нам в дальнейшем понадобится [1].

Предложение 2. *Пусть при некотором размещении с помощью алгоритма BLD T_a — прямоугольник, верхняя граница которого расположена на наибольшей высоте, t_a — высота размещения основания этого прямоугольника, тогда любое сечение этой полосы, проходящее ниже t_a, заполнено по крайней мере наполовину.*

3.2. Описание Алгоритма 1

Каждый следующий прямоугольник T_j отправляется на одну из полос C_i, на которой он вообще может быть размещен (то есть $w(T_j) \leqslant w_i$) и у которой отношение S_i/w_i минимально. Здесь S_i — суммарная площадь прямоугольников, уже размещенных на i-й полосе.

После того как все прямоугольники T_j исчерпаны, в каждой полосе применяется алгоритм BLD.

3.3. Пример размещения со сколь угодно большим отношением

Теорема 1. *Для любого r существуют такое множество полос C и такая последовательность прямоугольников T, что*

$$\frac{H_{A_1}(T,C)}{H_O(T,C)} \geqslant r.$$

Доказательство. Достаточно предъявить пример таких C и T. Выберем некоторое целое $k > 0$.

(1) Пусть

$$C = \bigcup_{m=1}^{k} G_m$$

— множество полос, где

$$G_m = \{C_{m,1}, \dots, C_{m,2^{2m-3}} \mid C_{m,s} \text{ — полоса ширины } 2^{-m+1},\ s = 1, \dots, 2^{2^{m-3}}\},$$

$m = 1, 2, \dots, k$.

(2) Пусть

$$Y = \bigcup_{m=1}^{k} Y_m$$

— множество прямоугольников,

$$Y_m = \{Y_{m,1}, \dots, Y_{m,2^{2m-2}} \mid Y_{m,s} \text{ — прямоугольник ширины } 2^{-m+1} \text{ и высоты } 1\},$$

$m = 1, 2, \dots, k$.

(3) Пусть T — последовательность прямоугольников, полученная из Y сортировкой по возрастанию ширины.

Тогда

$$H_O(T,C) = 2$$

(все прямоугольники из Y_m размещаем на полосы из G_m в два слоя).

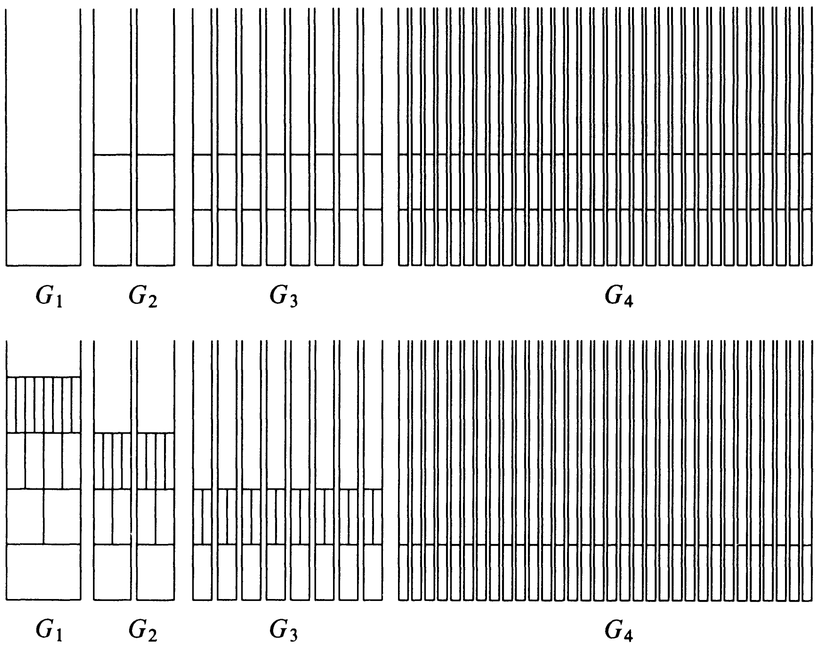

Рис. 1. Пример для Алгоритма 1, вверху — оптимальное решение

Применим алгоритм A_1 к T и C. Прямоугольники будут размещаться в следующем порядке: сначала все прямоугольники из множества Y_k, затем все из Y_{k-1} и так до Y_1. Так как

$$2^{2m-2}\frac{1}{2^{m-1}} = 2^{m-1} = 1 + \sum_{j=2}^{m} 2^{2j-3}\frac{1}{2^{j-1}},$$

суммарная ширина всех прямоугольников из Y_m равна суммарной ширине всех полос из $\bigcup_{i=1}^{m} G_i$, поэтому при размещении они равномерно распределятся по этим полосам и никак не будут влиять на дальнейшее размещение прямоугольников. Следовательно,

$$H_{A_1}(T, C) = k,$$

то есть

$$\frac{H_{A_1}(T, C)}{H_O(T, C)} = \frac{k}{2}.$$

Выбирая $k \geqslant 2r$, получим утверждение теоремы.

4. Алгоритм 2

4.1. Описание Алгоритма 2

Оказывается, что в вышеприведенный алгоритм можно внести небольшие изменения так, чтобы он стал алгоритмом с константной мультипликативной точностью.

Далее будем считать, что все полосы упорядочены по ширине, то есть для полос $C_1, \dots, C_m$ выполняются неравенства $w_1 \leqslant \dots \leqslant w_m$.

Введем обозначения

$$\text{first}(T_j) \text{ есть минимальное } i, \text{ для которого } w_i \geqslant w(T_j),$$

$$\text{last}(T_j) \text{ есть минимальное } r, \text{ для которого } \sum_{i=\text{first}(T_j)}^{r} w_i \geqslant \frac{1}{2} \sum_{i=\text{first}(T_j)}^{m} w_i.$$

Определение 1. Полосы с номерами $\text{first}(T_j), \text{first}(T_j)+1, \dots, \text{last}(T_j)$ назовем допустимыми для прямоугольника T_j.

Алгоритм 2 заключается в следующем.

Очередной прямоугольник T_j отправляется на одну из допустимых полос C_i, у которой отношение S_i/w_i минимально.

Прямоугольники в каждой полосе размещаются с помощью алгоритма BLD (так же, как и в Алгоритме 1).

Таким образом, Алгоритм 2 отличается от Алгоритма 1 только тем, что каждый прямоугольник может быть размещен не на любой полосе, на которой он может поместиться, а только на допустимой.

4.2. Доказательство верхней оценки

Докажем, что Алгоритм 2 позволяет решить основную задачу с константной мультипликативной точностью.

Теорема 2. *Для любого множества полос C и любой последовательности прямоугольников T справедливо неравенство*

$$\frac{H_{A_2}(T,C)}{H_O(T,C)} \leqslant 10.$$

Доказательство. Пусть в результате работы алгоритма самая большая высота размещения оказалась на k-й полосе. Пусть T_a — прямоугольник, который был добавлен последним на эту полосу, и

$$f = \text{first}(T_a), \qquad l = \text{last}(T_a).$$

Пусть $W_f, \dots, W_l$ — множества прямоугольников, которые уже были размещены на допустимых для T_a полосах $C_f, \dots, C_l$ непосредственно перед добавление прямоугольника T_a, и пусть $S_i = S(W_i)$ — площадь всех прямоугольников из W_i. Прямоугольник T_a добавили на k-ю полосу, поэтому

$$\frac{S_k}{w_k} \leqslant \frac{S_i}{w_i}, \qquad i = f, \dots, l.$$

Отсюда получаем, что

$$\sum_{i=f}^{l} S_i = \sum_{i=f}^{l} \frac{S_i}{w_i} w_i \geqslant \sum_{i=f}^{l} \frac{S_k}{w_k} w_i = \frac{S_k}{w_k} \sum_{i=f}^{l} w_i. \tag{1}$$

Пусть при размещении с помощью алгоритма BLD множества прямоугольников $W_k \cup \{T_a\}$ на k-й полосе T_{c_k} — прямоугольник, верхняя граница которого расположена на наибольшей высоте, t_k — высота размещения его нижней границы, и пусть

$$r_k = t_k - h(T_a).$$

Тогда

$$t_k + h(T_{c_k}) = H_{A_2}(T, C),$$
$$r_k + h(T_a) + h(T_{c_k}) = H_{A_2}(T, C). \tag{2}$$

По свойству алгоритма BLD (см. предложение 2)

$$S_k \geqslant \frac{1}{2} w_k r_k \quad \Longrightarrow \quad r_k \leqslant \frac{2S_k}{w_k}. \tag{3}$$

Пусть T_b — самый узкий из размещенных на полосах с номерами $f, \dots, l$ прямоугольников, и пусть $f_0 = \mathrm{first}(T_b)$. Тогда все прямоугольники, размещенные на $C_f, \dots, C_l$, не могут быть вообще размещены на полосах, имеющих номер, меньший f_0.

Так как T_b размещен на одной из полос $C_f, \dots, C_l$, справедливы соотношения

$$\mathrm{last}(T_b) \geqslant f \Longrightarrow \sum_{i=f_0}^{f-1} w_i \leqslant \frac{1}{2} \sum_{i=f_0}^{m} w_i$$
$$\Longrightarrow \sum_{i=f}^{m} w_i \geqslant \frac{1}{2} \sum_{i=f_0}^{m} w_i.$$

Так как $l = \mathrm{last}(T_a)$, справедливо неравенство

$$\sum_{i=f}^{l} w_i \geqslant \frac{1}{2} \sum_{i=f}^{m} w_i.$$

Поэтому

$$\sum_{i=f_0}^{m} w_i \leqslant 2 \sum_{i=f}^{m} w_i \leqslant 4 \sum_{i=f}^{l} w_i.$$

Далее,

$$H_O \sum_{i=f_0}^{m} w_i \geqslant \sum_{i=f}^{l} S_i.$$

Подставляя (1) в это неравенство, получаем, что

$$H_O \sum_{i=f_0}^{m} w_i \geqslant \frac{S_k}{w_k} \sum_{i=f}^{l} w_i \geqslant \frac{S_k}{4w_k} \sum_{i=f_0}^{m} w_i.$$

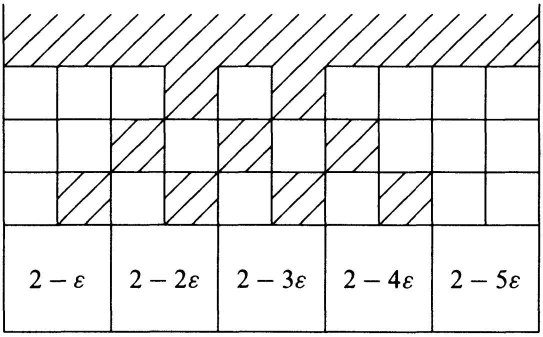

Рис. 2. Пример шахматной доски

Отсюда, используя (3), находим, что

$$r_k \leqslant \frac{2S_k}{w_k} \leqslant 8H_O. \tag{4}$$

Так как $H_O \geqslant h(T_j)$ для всех j, из (2) и (4) получаем, что

$$8H_O + H_O + H_O \geqslant H_{A_2}.$$

Следовательно,

$$\frac{H_{A_2}}{H_O} \leqslant 10.$$

5. Нижняя оценка для Алгоритма 2

Покажем теперь, что границу 10, полученную в теореме 2, нельзя улучшить.

Теорема 3. *Для любого $\varepsilon_0 > 0$ существуют такое множество полос C и последовательность прямоугольников T, для которых*

$$\frac{H_{A_2}(T, C)}{H_O(T, C)} \geqslant 10 - \varepsilon_0.$$

5.1. Пример шахматной доски

Построим вспомогательный пример размещения прямоугольников, аналогичный примеру шахматной доски, приведенному в [1].

Пусть имеется n прямоугольников $T_1, \ldots, T_n$, с

$$w(T_j) = h(T_j) = 2 - j\varepsilon;$$

N_s прямоугольников с шириной и высотой, равными 1, и одна полоса ширины $2n$.

При использовании алгоритма BLD эти прямоугольники будут размещены так, как показано на рис. 2.

Пусть k — число слоев в этом размещении, состоящих из маленьких прямоугольников. Будем считать, что n и N_s выбраны так, что последний слой заполнен полностью и выполняется соотношение

$$2n = k^2.$$

Тогда суммарное число всех маленьких прямоугольников удовлетворяет неравенству

$$N_s \leqslant \frac{k^3}{2} + O(k^2).$$

5.2. Основной пример

Вначале введем обозначения и выберем параметры.

(1) Выберем произвольное $\delta > 0$.

(2) Построим пример шахматной доски такой, что

$$k = 8q,$$

где q — целое число,

$$\frac{1}{k} < \delta, \quad N_s < k^3\left(\frac{1}{2} + \delta\right), \quad \varepsilon < \frac{1}{n}.$$

Положим

$$h_0 = \frac{k}{8}, \qquad w_0 = 2n = k^2.$$

(3) Выберем положительное целое число m такое, что

$$\frac{k}{m} < \delta.$$

(4) Выберем $0 < \theta < 1$ так, что

$$\theta^{2m} > 1 - \frac{\varepsilon}{2n},$$
$$\theta^{2m} > \left(1 + \frac{8}{k^3}\right)^{-1}.$$

(5) Введем обозначения

$$\begin{aligned} w_i^B &= \theta^{2m-i}, & i &= 1,\dots,m,\\ w_i^C &= \theta^{3m-i}, & i &= 1,\dots,m-1,\\ w_i^D &= \theta^{4m-i}, & i &= 1,\dots,m-1. \end{aligned}$$

Опишем полосы, используемые в данном примере.

(1) Имеются $2m$ полос шириной w_0, обозначим их $A_1,\dots,A_{2m}$;

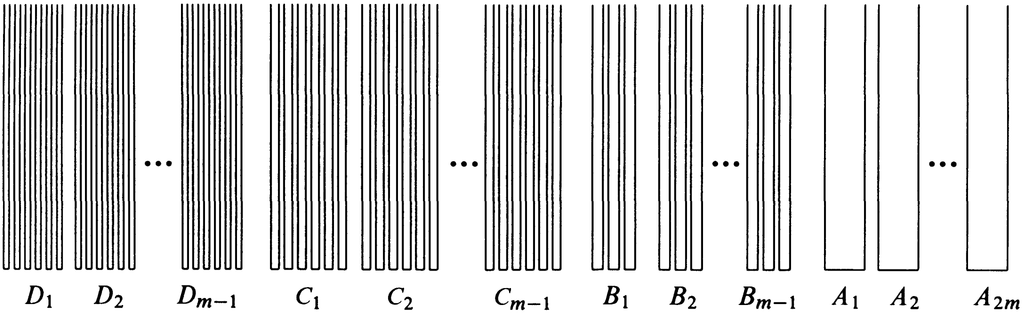

Рис. 3. Полосы основного примера

(2) $m-1$ множеств полос $B_1, \ldots, B_{m-1}$, B_i состоит из $n_B = 4n$ полос ширины w_i^B, $i = 1, \ldots, m-1$; пусть N_B — суммарное число полос во всех этих множествах;

(3) $m-1$ множеств $C_1, \ldots, C_{m-1}$, C_i состоит из $n_C = n_B$ полос ширины w_i^C, $i = 1, \ldots, m-1$;

(4) $m-1$ множеств $D_1, \ldots, D_{m-1}$, D_i состоит из $n_D = n_C$ полос ширины w_i^D, $i = 1, \ldots, m-1$.

Будем считать, что полосы упорядочены по возрастанию по ширине следующим образом: $\{D_1\}, \ldots, \{D_{m-1}\}, \{C_1\}, \ldots, \{C_{m-1}\}, \{B_1\}, \ldots \{B_{m-1}\}, A_1, \ldots, A_{2m}$.

Опишем прямоугольники, используемые в примере.

(1) Имеется $m-1$ множеств прямоугольников $R_1^D, \ldots, R_{m-1}^D$, R_i^D состоит из n_D прямоугольников ширины w_i^D и высот h_0, $i = 1, \ldots, m-1$;

(2) $m-1$ множеств прямоугольников $R_1^C, \ldots, R_{m-1}^C$, R_i^C состоит из n_C прямоугольников ширины w_i^C и высоты h_0, $i = 1, \ldots, m-1$;

(3) множество Q состоит из m прямоугольников, причем i-й имеет ширину $w_i^B \theta^{1/2}$ и высоту h_0, $i = 1, \ldots, m$;

(4) m множеств прямоугольников $X_1, \ldots, X_m$, X_i состоит из

$$N_X = 2h_0 w_0 = 2(k/8)2n = k^3/4$$

прямоугольников ширины w_i^B и высоты 1, $i = 1, \ldots, m$;

(5) m множеств прямоугольников $Y_1, \ldots, Y_m$, Y_i состоит из $N_Y = N_s - N_X$ прямоугольников ширины 1 и высоты 1, $i = 1, \ldots, m$;

(6) m множеств $U_1, \ldots, U_m$, U_i состоит из n прямоугольников, $i = 1, \ldots, m$, j-й прямоугольник из U_i имеет ширину $2 - j\varepsilon$ и высоту $2 - j\varepsilon$, так как $\varepsilon < 1/n$, высота и ширина каждого из этих прямоугольников больше 1.

(7) V — один прямоугольник ширины w_0 и высоты h_0.

Приведем теперь некоторые утверждения.

(1) Для прямоугольников $V, U_1 \dots U_m, Y_1 \dots Y_m, X_m$ допустимыми являются полосы $A_1, \dots, A_m$ и только они.

(2) Для прямоугольников из X_i полоса A_i является допустимой.

(3) Для i-го прямоугольника из Q полоса A_i является допустимой.

(4) Для прямоугольников из $X_1, \dots, X_{m-1}$ и из Q полосы из $C_1, \dots, C_{m-1}$ и из $D_1, \dots, D_{m-1}$ не являются допустимыми.

(5) Для прямоугольников из R_i^C все полосы из B_i являются допустимыми, а полосы из $D_1, \dots, D_{m-1}$ допустимыми не являются.

(6) Для прямоугольников из R_i^D все полосы из C_i являются допустимыми.

(7) Полосы $A_{m+1}, \dots, A_{2m}$ не являются допустимыми ни для одного из прямоугольников.

5.3. Размещение с помощью Алгоритма 2

Далее отношение S_a/w_a для некоторой полосы a будем называть заполненностью этой полосы.

Заметим, что размещение прямоугольников, получающееся при использовании алгоритма, существенно зависит от порядка, в котором они поступают. Выберем порядок, при котором отношение H_{A_2}/H_O получится наибольшим.

Опишем распределение прямоугольников по полосам.

(1) Сначала для каждого $i = 1, \dots, m-1$ приходят прямоугольники из R_i^D, они размещаются на полосы из C_i по одному прямоугольнику на каждую полосу. После этого заполненность любой полосы из C_i станет равной

$$\frac{h_0 w_i^D}{w_i^C} = h_0 \frac{\theta^{3m-i}}{\theta^{4m-i}} = h_0 \theta^{-m}.$$

(2) Далее для каждого $i = 1, \dots, m-1$ приходят прямоугольники из R_i^C, они размещаются на полосы из B_i также по одному прямоугольнику на каждую полосу. После этого заполненность любой полосы из B_i станет равной

$$\frac{h_0 w_i^C}{w_i^B} = h_0 \frac{\theta^{2m-i}}{\theta^{3m-i}} = h_0 \theta^{-m}.$$

(3) Далее приходят прямоугольники из $X_1, \dots, X_m$. Они приходят в таком порядке, чтобы прямоугольники из X_i заполняли полосу A_i. Так должно продолжаться до тех пор пока в каждой полосе $A_1, \dots, A_m$ не будет ровно по $N_X/2$ прямоугольников. При этом заполненность полосы A_i станет равной

$$\frac{w_i^B N_x/2}{w_{A_i}} = \frac{h_0 w_0 w_i^B}{w_0} = h_0 \theta^{2m-i} \leqslant h_0 \theta^{-m}.$$

(4) Затем для каждого $i = 1, \dots, m$ приходит еще один прямоугольник из X_i и размещается на полосе A_i. Так как

$$\begin{aligned}\frac{(1+N_x/2)w_i^B}{w_0} = \frac{(1+w_0h_0)w_i^B}{w_0} &= h_0(1+8/k^3)\theta^{2m-i}\\ &> h_0(1+8/k^3)\theta^{2m}\\ &> h_0 > h_0\theta^{-m},\end{aligned}$$

теперь для любых i и j заполненность полосы A_i больше заполненности любой из полос множества B_j.

(5) Далее для каждого $i = 1, \dots, m-1$ приходят оставшиеся прямоугольники из R_i^C. Они отправляются на полосы из B_i так, что в каждой полосе окажется два прямоугольника из R_i^C.

(6) Далее приходят оставшиеся прямоугольники из $X_1, \dots, X_m$. Они опять приходят в таком порядке, чтобы прямоугольники из X_i заполняли полосу A_i. Теперь в каждой из полос $A_1, \dots A_m$ находится

$$N_X = \frac{k^3}{4}$$

прямоугольников. Заполненность полосы A_i станет равной

$$\frac{N_x w_i^B}{w_0} = 2h_0\theta^{2m-i},$$

а заполненность полос из B_i станет равной

$$\frac{2h_0 w_i^C}{w_i^B} = 2h_0\theta^{-m}.$$

(7) Затем приходят прямоугольники из Q, i-й прямоугольник отправляется на полосу A_i. Для всех оставшихся прямоугольников допустимыми являются только полосы $A_1, \dots A_m$.

(8) Далее приходят все прямоугольники из $Y_1, \dots, Y_m$. Они равномерно распределятся по полосам $A_1, \dots, A_m$.

(9) Далее приходят прямоугольники из $U_1, \dots, U_m$. Для каждого $j = 1, \dots, n$ приходят j-е прямоугольники из всех U_i и j-й прямоугольник U_i отправляется на полосу A_i.

(10) Последним приходит прямоугольник V. Он отправляется на одну из полос $A_1, \dots, A_m$.

Опишем размещение внутри полос.

Далее внутри каждой полосы прямоугольники размещаются с помощью алгоритма BL. Так как ширина прямоугольников из X_i равна

$$w_i^B = \theta^{2m-i} > \theta^{2m} > 1 - \frac{\varepsilon}{2n},$$

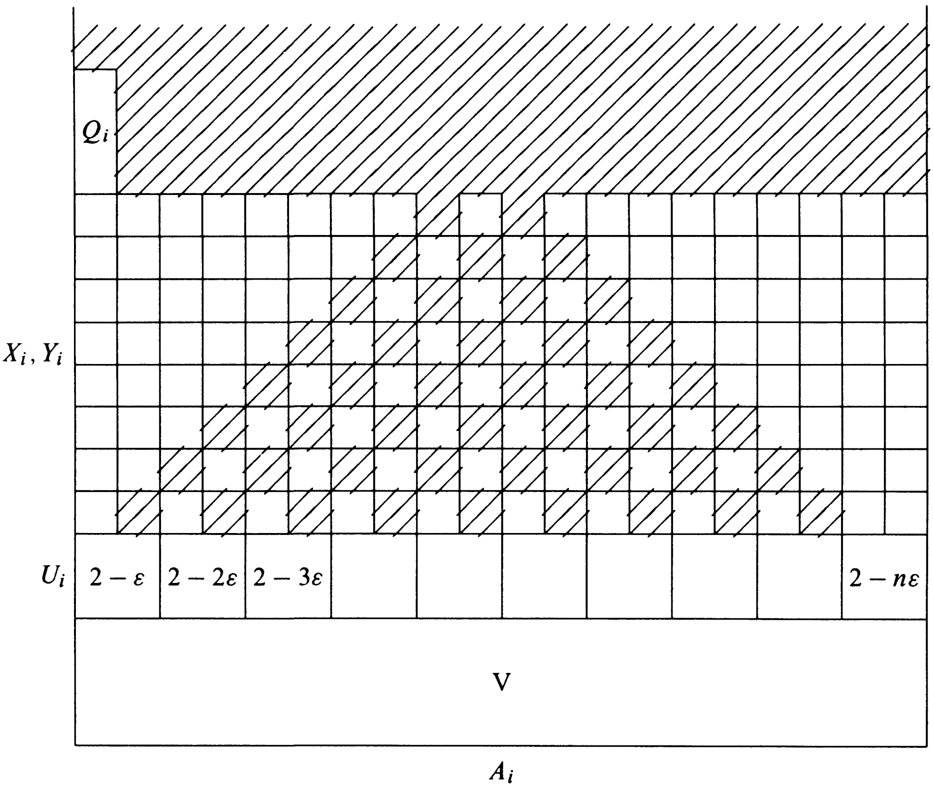

Рис. 4. Размещение с помощью Алгоритма 2

при их размещении в полосах $A_1, \ldots, A_m$ расположение прямоугольников практически не будет отличаться от расположения в примере шахматной доски.

После такого размещения прямоугольников, полоса, на которую попал прямоугольник V, будет выглядеть так, как показано на рис. 4.

Высота H_{A_2} полученного размещения удовлетворяет соотношениям

$$\begin{aligned} H_{A_2} &= h_0 + 2 - \varepsilon + k + h_Q \\ &> h_0 + 8h_0 + h_0 = 10h_0. \end{aligned}$$

5.4. Оптимальное размещение для основного примера

Для доказательства нижней оценки не обязательно размещать прямоугольники оптимальным образом, достаточно выбрать какое-нибудь размещение близкое к оптимальному, позволяющее доказать требуемую оценку.

Размещение будет следующим.

(1) Для каждого $i = 1, \ldots, m-1$ все прямоугольники из R_i^D размещаем на полосах из D_i.

(2) Для каждого $i = 1, \ldots, m-1$ все прямоугольники из R_i^C размещаем на полосах из C_i.

(3) Для каждого $i = 1, \ldots, m-1$ все прямоугольники из X_i размещаем на полосы из B_i. Высота этого размещения окажется равной

$$\frac{N_X}{n_b} = \frac{k^3/4}{4n} = \frac{k}{8} = h_0.$$

(4) Прямоугольник V размещаем на A_{2m}.

(5) Размещаем i-й прямоугольник из Q на полосу A_i так, чтобы он был расположен в левом нижнем углу полосы.

(6) В полосах $A_1, \ldots, A_{2m-1}$ размещаем прямоугольники из $Y_1, \ldots, Y_m$ так, чтобы они заполнили их до высоты h_0 (укладываем их в столбцы один на другой).

(7) Для каждого $i = 1, \ldots, m$ размещаем прямоугольники из U_i в один слой на полосе A_i поверх уже размещенных там прямоугольников.

(8) Далее все оставшиеся прямоугольники из $Y_1, \ldots, Y_m$ и прямоугольники из X_m размещаем равномерно сверху по полосам $A_1, \ldots, A_{2m}$.

Оценим высоту предложенного размещения. Число N_l размещенных на последнем шаге прямоугольников из $Y_1, \ldots, Y_m$ и X_mудовлетворяет соотношениям

$$\begin{aligned}
N_l &= N_X + mN_Y - \frac{k}{8}((m-1)2n + m(2n-1)) \\
&= \frac{k^3}{4} + mN_y - m\frac{k^3}{4} + \frac{k}{8}(2n+m), \\
N_Y &= N_S - N_X = N_s - \frac{k^3}{4} \\
&\leqslant k^3(1/4 + \delta), \\
N_l &\leqslant mk^3\delta + \frac{k^3}{4} + \frac{k}{8}(2n+m).
\end{aligned}$$

Высота, которую займут эти прямоугольники, не превзойдет

$$\begin{aligned}
H_l &\leqslant \frac{N_l}{2n2m} + 1 \\
&\leqslant \frac{mk^3\delta + k^3/4 + (k/8)(2n+m)}{2mk^2} + 1 \\
&\leqslant \frac{k\delta}{2} + \frac{3k}{16m} + \frac{1}{16k} + 1 \\
&\leqslant k\delta + 1.
\end{aligned}$$

Суммарная высота размещения не превзойдет

$$\begin{aligned} H_O &\leqslant h_0 + 2 + H_l \\ &\leqslant h_0 + 3 + k\delta \\ &\leqslant h_0(1 + 24/k + 8\delta) \\ &\leqslant h_0(1 + 32\delta). \end{aligned}$$

Итак,

$$\frac{H_{A_2}}{H_O} \geqslant \frac{10}{1 + 32\delta} \geqslant 10(1 - 32\delta).$$

При $\delta < \varepsilon_0/320$, получим утверждение теоремы 3

$$\frac{H_{A_2}}{H_O} \geqslant 10 - \varepsilon_0.$$

Замечание 1. Даже если в Алгоритме 2 прямоугольники внутри каждой полосы размещать оптимальным образом (а не с помощью алгоритма BLD), нельзя получить алгоритм с мультипликативной точностью, меньшей 5. Чтобы доказать это, достаточно рассмотреть пример из теоремы 3 и оценить высоту, которая получится при оптимальном размещении прямоугольников внутри полос и таком же, как и раньше, распределении по полосам.

Список литературы

1. Baker B. S., Coffman E. J., Rivest R. L., Orthogonal packings in two dimensions. *SIAM J. Comput.* (1980) **9**, 846–855.

2. Kenyon C., Remila E., A near optimal solution to a two-dimensional cutting stock problem. *Math. Oper. Research* (2000) **25**, 645–656.

3. Baker B. S., Brown D. J., Katseff H. P., A 5/4 algorithm for two-dimensional packing. *J. Algorithms* (1981) **2**, 348–368.

4. Brucker P., *Scheduling algorithms.* Springer, Berlin, 1998.

5. Coffman B. J., Garey M. R., Johnson D. S., Tarjan R. E., Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM J. Comput.* (1980) **9**, 808–826.

6. Drozdowski M., Scheduling multiprocessor tasks—an overview. *European J. Oper. Research* (1996) **94**, 215–230.

7. Foster, Kesselman C., *The Grid: Blueprint for a future computing infrastructure.* Morgan Kaufmann, San Francisco, 1999.

8. Coffman E. G., Jr., Garey M. R., Johnson D. S., Approximation algorithms for Bin-packing—An updated survey. In: *Algorithm design for computer system design.* Springer, Berlin, 1984, pp. 49–106.

Статья поступила 26.01.2005.