

# Онлайновый алгоритм упаковки прямоугольников в несколько полос с гарантированными оценками точности<sup>1</sup>

С.Н. Жук

**Аннотация.** В работе описан онлайн-алгоритм упаковки произвольного множества прямоугольников в несколько полос и доказано, что он гарантирует нахождение решения, отличающегося от оптимального не более чем в константу раз.

## 1. Введение

Задача упаковки прямоугольников в несколько полос заключается в следующем: имеется несколько полубесконечных полос определённой ширины и набор прямоугольников. Требуется найти ортогональное размещение этих прямоугольников по полосам (без пересечений и вращений), имеющее минимальную высоту. Эта задача возникает, например, при распределении ресурсов в глобальных вычислительных сетях Grid, объединяющих различные вычислительные ресурсы (кластеры). Проблема оптимального размещения задач на группе кластеров имеет простую геометрическую интерпретацию [17]. Кластер можно представить как бесконечную в одну сторону полосу определённой 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 (bin-packing [1]). 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 существу-

<sup>1</sup>Работа выполнена при поддержке РФФИ, проект 05-01-00798.

ет эффективный алгоритм, позволяющий решить эту задачу точно. Поэтому рассматривают различные приближённые алгоритмы, которые позволяют достаточно быстро найти некоторое решение с определённой точностью.

Для оценки точности приближённых алгоритмов можно использовать следующие характеристики. Пусть  $T$  — некоторая последовательность прямоугольников,  $C$  — множество полос,  $H_O(T, C)$  и  $H_A(T, C)$  — высота оптимального размещения прямоугольников и высота размещения, получаемая с помощью алгоритма  $A$ , на множестве полос  $C$ . Тогда абсолютная точность алгоритма  $A$  определяется как

$$R_A = \sup_{T, C} \{H_A(T, C)/H_O(T, C)\},$$

а асимптотическая как

$$R_A^\infty = \lim_{k \rightarrow \infty} \sup_{T, C} \{H_A(T, C)/H_O(T, C) \mid H_O(T, C) \geq k\}.$$

Особый интерес представляют так называемые online алгоритмы. Эти алгоритмы отличаются от обычных тем, что они производят последовательное рассмотрение всех размещаемых объектов. В таких алгоритмах, каждый приходящий объект сразу же размещается, и его положение не изменяется с приходом следующих. Как правило такие алгоритмы более эффективны, кроме того их удобнее использовать на практике, так как не всегда с самого начала известны все размещаемые объекты.

Наиболее исследованы частные случаи рассматриваемой задачи. Один из них — уже упомянутая классическая задача об упаковке в контейнеры (bin-packing).

Другой частный случай, когда у нас имеется всего одна полоса, — это также известная и хорошо исследованная задача. Впервые её изучали Baker, Coffman и Rivest [2] в 1980 году. Ими была предложена простая эвристика Bottom-Left, позволяющая размещать прямоугольники внутри одной полосы с абсолютной точностью 3. Затем в 1981 году Baker, Brown и Katseff [4] представили offline алгоритм гарантирующий асимптотическую точность 5/4. И этот алгоритм оставался лучшим до тех пор как в 1996 году Kenyon и Remila [3] не построили алгоритм, позволяющий за полиномиальное время находить почти оптимальное решение (с асимптотической точностью  $1 + \varepsilon$  для любого  $\varepsilon > 0$ ).

Что касается online алгоритмов, то здесь тоже имеется значительное количество исследований. В 1983 году Baker и Shwartz [5] представили так называемые шельфовые алгоритмы. В данных алгоритмах полоса разбивается на гори-

зонтальные слои(шельфы) различной высоты, а для размещения прямоугольников внутри шельфов используется какая-нибудь эвристика для одномерной упаковки в контейнеры(bin-backing). Один из таких алгоритмов с очень простой эвристикой First Fit позволяет добиться асимптотической точности 1.7. Затем Csirik и Woeginger [8] показали, что никакой шельфовый алгоритм не может иметь асимптотическую точность меньше чем 1.69103 и построили алгоритм, позволяющий сколь угодно близко подойти к данному значению. Также Van Vliet [10] показал, что никакой online алгоритм упаковки прямоугольников в несколько полос не позволяет добиться асимптотической точности лучшей чем 1.5401.

Еще один важный частный случай — это задача о составлении расписания для  $m$  машин. Если предположить, что имеется  $m$  одинаковых полос ширины 1 и все прямоугольники также имеют ширину равную 1, то как раз получится данная задача. Впервые её исследовал Graham [11], он предложил очень простой online алгоритм, который размещает каждую новую задачу на наименее загруженную машину. Точность этого алгоритма равна  $2 - 1/m$ , для  $m$  машин. Несмотря на простоту данного алгоритма, долгое время не удавалось построить алгоритм, точность которого бы была меньше  $2 - \varepsilon$  для любого  $m$  (для фиксированного положительного  $\varepsilon$ ). Только в 1992 году Bartal et al.[13] предложили алгоритм с точностью не хуже 1.986 для всех  $m > 70$ . Это значение затем было улучшено до 1.945, а затем и до 1.923 в работах [15] и [12]. Также было получено несколько результатов о нижних оценках для таких алгоритмов. Bartal, Karloff и Rabani [16] показали, что ни один детерминированный алгоритм не может иметь точность лучше, чем 1.837 для достаточно большого  $m$ . Этот результат был улучшен Albers [12] до 1.852.

Что касается задачи об упаковке прямоугольников в несколько полос, то она до сих пор остается в значительной степени не исследованной. А. Пospelov [21] предложил offline алгоритм, основанный на эвристике Bottom-Left, позволяющий размещать прямоугольники с абсолютной точностью равной 3. В работе [18] был рассмотрен алгоритм, размещающий прямоугольники по полосам в режиме online, позволяющий добиться точности 10.

Во данной работе предложен алгоритм, работающий полностью в режиме online, размещающий прямоугольники в несколько полос с константной мультипликативной точностью 17 и асимптотической точностью сколь угодно близкой к 8. В разделе 2 дано более формальное описание задачи. В разделе 3 рассмотрены шельфовые алгоритмы, которые будут использоваться для упаковки прямоугольников внутри полосы. В разделах 4 и 5 описан сам алгоритм и доказаны соответствующие теоремы о верхних оценках для абсолютной и асимпто-

тической точности.

## 2. Постановка задачи

*Вход:*  $T = \{T_1, \dots, T_n\}$  — конечная последовательность прямоугольников,  $h(T_j)$  и  $w(T_j)$  — соответственно высота и ширина прямоугольника  $T_j$  (максимальная высота одного прямоугольника равна  $h_{max}$ ), и  $C = \{C_1, \dots, C_m\}$  множество полубесконечных полос,  $w_i$  — ширина  $i$ -й полосы.

*Задача:* Найти ортогональное размещение последовательности прямоугольников  $T$  по этим полосам, минимизирующее полную высоту этого размещения, то есть максимум по всем прямоугольникам и по всем полосам расстояния от дна полосы до верхней грани прямоугольника.

Пусть  $C$  и  $T$  — заданные множества полос и прямоугольников, и пусть  $A$  — некоторый алгоритм размещающий прямоугольники по полосам. Тогда будем обозначим  $H_O(T, C)$  — 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  $A$ .

Для оценки точности приближенных алгоритмов мы будем использовать следующие характеристики.

1. Абсолютная точность алгоритма  $A$  определяется как

$$R_A = \sup_{T, C} \{H_A(T, C)/H_O(T, C)\}.$$

2. Асимптотическая точность алгоритма  $A$  определяется как

$$R_A^\infty = \lim_{k \rightarrow \infty} \sup_{T, C} \{H_A(T, C)/H_O(T, C) \mid H_O(T, C) \geq k\}.$$

## 3. Шельфовый алгоритм

Для размещения прямоугольников внутри каждой из полос будет использоваться так называемый шельфовый алгоритм. Он был рассмотрен в [4] для случая одной полосы.

Пусть задана некоторая константа  $r \in (0, 1)$ . Шельф — это прямоугольная часть полосы, с шириной равной ширине полосы. Как только приходит новый прямоугольник  $T_i$ , шельфовый алгоритм определяет целое число  $k$ , такое что  $r^{k+1} < h(T_i) \leq r^k$  и после этого он размещает данный прямоугольник в шельфе высотой  $r^k$ . Прямоугольник может быть размещен в один из существующих

шelfов, или сверху может быть создан новый шelf высоты  $r^k$ . Прямоугольни- ки внутри каждого шelfа размещаются по горизонтали в ряд. Таким обра- зом, упаковка прямоугольников, попавших в один класс по высоте (то есть ко- торым соответствует одно значение  $k$ ), производится с помощью некоторой эври- стики, осуществляющей одномерную упаковку в контейнеры (Bin-Packing). Мы будем использовать эвристику First Fit (FF), которая каждый прямоуголь- ник помещается в первый шelf в который он вообще может быть размещен, и если таких нет, то создает новый шelf.

Описанный выше алгоритм будем обозначать  $Shelf(r)$ . Нам понадобится следующее свойство шelfовых алгоритмов.

**Теорема 1.** Пусть суммарная площадь прямоугольников упакованных на данную полосу равна  $S$ , ширина данной полосы равна  $w$  и максимальная высота прямоугольника равна  $h_{max}$ , тогда для высоты упаковки полу- чившейся с помощью алгоритма  $Shelf(r)$  справедливо

$$H_{Shelf(r)} \leq \frac{2}{r} \cdot \frac{S}{w} + \frac{h_{max}}{r(1-r)}.$$

*Доказательство.* Пусть множество шelfов данной полосы — это  $E = \{e_1, \dots, e_p\}$ .

Разобьем все шelfы на две группы:

$$G_1 = \{e \in E : e \text{ заполнен меньше чем на половину по ширине}\}$$

$$G_2 = \{e \in E : e \text{ заполнен не меньше чем на половину по ширине}\}$$

Обозначим также  $H_i = \sum_{e \in G_i} h(e)$ ,  $i = 1, 2$  ( $h(e)$  — высота шelfа  $e$ ).  $H_i$  — суммарная высота шelfов из группы  $G_i$ . Тогда для высоты размещения с помощью алгоритма  $A$  можно записать

$$H_{Shelf(r)} \leq H_1 + H_2. \quad (1)$$

Получим теперь оценки для  $H_1$  и  $H_2$ . Рассмотрим сначала группу  $G_1$ . Каж- дый шelf в этой группе заполнен меньше, чем на половину. Поэтому, так как мы используем эвристику First Fit, то каждой высоты у нас не больше одного шelfа.

Пусть  $h_1$  — высота самого большого шelfа в  $G_1$ , тогда

$$H_1 < h_1 + h_1 r + h_1 r^2 + h_1 r^3 + \dots = \frac{h_1}{1-r}.$$

Так как  $h_{max} > r \cdot h_1$ , то

$$H_1 \leq \frac{h_{max}}{r(1-r)}. \quad (2)$$

Оценим теперь  $H_2$ . Каждый из шelfов в  $G_2$  заполнен по ширине не меньше чем на половину и высота любого прямоугольника в нем составляет не менее  $r$  от высоты шelfа. Поэтому,

$$S \geq \frac{1}{2} r H_2 w \implies H_2 \leq \frac{2}{r} \frac{S}{w}. \quad (3)$$

Подставляя (2) и (3) в (1), получим

$$H_{Shelf(r)} \leq \frac{2}{r} \cdot \frac{S}{w} + \frac{h_{max}}{r(1-r)}.$$

#### 4. Описание алгоритма $A(r, \alpha)$

Будем считать, что все полосы упорядочены по ширине, т.е. для  $C_1, \dots, C_m$  выполняется  $w_1 \geq w_2 \geq \dots \geq w_m$ . И пусть задано некоторое число  $\alpha \in (0, 1)$ .

Будем использовать следующие обозначения:

$$last(T_j) = \max k : w_k \geq w(T_j)$$

$$first(T_j) = \max r : \sum_{k=r}^{last(T_j)} w_k \geq \alpha \sum_{k=1}^{last(T_j)} w_k$$

Полосы с номерами  $first(T_j)$ ,  $first(T_j) + 1, \dots, last(T_j)$  будем называть **допустимыми** для прямоугольника  $T_j$ .

##### Алгоритм для размещения прямоугольников.

1. Для каждого поступающего прямоугольника  $T_j$  производится рассмот- рение всех **допустимых** для него полос. И затем прямоугольник  $T_j$  от- правляется на одну из **допустимых** полос  $C_i$ , у которой отношение  $\frac{S_i}{w_i}$  минимально. Здесь  $S_i$  — суммарная площадь прямоугольников уже раз- мещённых на  $i$ -й полосе.
2. Для упаковки прямоугольников внутри каждой полосы используется шelfовый алгоритм  $Shelf(r)$ .

Описанный выше алгоритм зависит от двух параметров:  $r$  и  $\alpha$ . Будем его обозначать  $A(r, \alpha)$ .

## 5. Верхняя оценка для алгоритма $A(r, \alpha)$

**Теорема 2.** Для алгоритма  $A = A(r, \alpha)$  для любых множества полос  $C$  и последовательности прямоугольников  $T$  справедлива оценка:

$$H_A(T, C) \leq \frac{1}{\alpha(1-\alpha)} \cdot \frac{2}{r} \cdot H_O(T, C) + h_{max} \left( \frac{1}{r(1-r)} + 1 \right).$$

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

$$f = first(T_a), \quad l = last(T_a).$$

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  $W_i$ . Так как его добавили на  $k$ -ю полосу, то

$$\frac{S_k}{w_k} \leq \frac{S_i}{w_i}, \quad \forall i = f, \dots, l.$$

Поэтому,

$$\sum_{i=f}^l S_i = \sum_{i=f}^l \frac{S_i}{w_i} \cdot w_i \geq \sum_{i=f}^l \frac{S_k}{w_k} \cdot w_i = \frac{S_k}{w_k} \cdot \sum_{i=f}^l w_i. \quad (4)$$

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

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

$$first(T_b) \leq l \implies \sum_{i=l+1}^{\bar{l}} w_i < \alpha \sum_{i=1}^{\bar{l}} w_i \implies \sum_{i=1}^l w_i \geq (1-\alpha) \sum_{i=1}^{\bar{l}} w_i.$$

Так как  $f = first(T_a)$ , то

$$\sum_{i=f}^l w_i \geq \alpha \sum_{i=1}^l w_i \geq \alpha(1-\alpha) \sum_{i=1}^{\bar{l}} w_i.$$

Далее,

$$H_O \cdot \sum_{i=f_0}^m w_i \geq \sum_{i=f}^l S_i. \quad (5)$$

Подставляя сюда (4) и используя (5), получаем

$$H_O \cdot \sum_{i=1}^{\bar{l}} w_i \geq \frac{S_k}{w_k} \cdot \sum_{i=f}^l w_i \geq \alpha(1-\alpha) \cdot \frac{S_k}{w_k} \cdot \sum_{i=1}^{\bar{l}} w_i.$$

Или, что то же самое

$$\frac{S_k}{w_k} \leq \frac{H_O}{\alpha(1-\alpha)}. \quad (6)$$

Используя теорему 1 (свойство шельфовых алгоритмов) для полосы с номером  $k$  (прямоугольник  $T_a$  или размещается в один из существующих шельфов или размещается сверху образуя новый шельф)

$$H_A(T, C) \leq \frac{2}{r} \cdot \frac{S_k}{w_k} + h_{max} \left( \frac{1}{r(1-r)} \right) + h(T_a).$$

Используя (6) и то, что  $h_{max} \geq h(T_a)$ , получим

$$H_A(T, C) \leq \frac{1}{\alpha(1-\alpha)} \cdot \frac{2}{r} \cdot H_O(T, C) + h_{max} \left( \frac{1}{r(1-r)} + 1 \right).$$

Таким образом, теорема 2 доказана.

Используя, то что  $H_O(T, C) \geq h_{max}$ , получим:

**Следствие 1.** Для алгоритма  $A = A(r, \alpha)$  для любых множества полос  $C$  и последовательности прямоугольников  $T$  справедливо

$$\frac{H_A(T, C)}{H_O(T, C)} \leq \frac{1}{\alpha(1-\alpha)} \cdot \frac{2}{r} + \frac{1}{r(1-r)} + 1.$$

Минимум выражения в правой части достигается при  $r = \frac{3}{4}$  и  $\alpha = \frac{1}{2}$ .

**Следствие 2.** Для алгоритма  $A = A(r = \frac{3}{4}, \alpha = \frac{1}{2})$  для любых множества полос  $C$  и последовательности прямоугольников  $T$  справедливо

$$\frac{H_A(T, C)}{H_O(T, C)} \leq 17.$$

**Следствие 3.** Для алгоритма  $A = A(r, \alpha = \frac{1}{2})$

$$R_A^\infty s \leq \frac{8}{r}.$$

## Литература

- [1] M.R. Garey and D.S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979). 7
- [2] B.S. Baker, E.J. Coffman and R.L. Rivest, Orthogonal packings in two dimensions, *SIAM J. Computing*, 1980, v. 9, pp. 846-855. 8
- [3] C. Kenyon and E. Remila, A near optimal solution to a two-dimensional cutting stock problem, *Mathematics of Operations Research*, 2000, v. 25, pp. 645-656. 8
- [4] B.S. Baker, D.J. Brown and H.P. Katseff, A  $5/4$  algorithm for two-dimensional packing, *J. of Algorithms*, 1981, v. 2, pp. 348-368. 8
- [5] B.S. Baker and J.S. Schwarz, Shelf algorithms for two-dimensional packing problems, *SIAM J. Comput.*, 1983, v. 12, pp. 508-525. 8
- [6] E. G. Coffman, Jr., D. S. Johnson, P. W. Shor and G. S. Lueker. Probabilistic analysis of packing and related partitioning problems, *Statistical Science*, 1993, v. 8, pp. 40-47.
- [7] E. G. Coffman, Jr., P. W. Shor. Packings in two dimensions: Asymptotic average-case analysis of algorithms, *Algorithmica*, 1993, v. 9, pp. 253-277.
- [8] J. Csirik, G.J. Woeginger, Shelf Algorithms for On-Line Strip Packing. *Inf. Process. Lett.*, 1997, v. 63, N 4, pp. 171-175. 9
- [9] Ee-Chien Chang, W. Wang, M.S. Kankanhalli, Multidimensional on-line bin-packing: An algorithm and its average-case analysis, *Information Processing Letters*, 1993, v. 48, pp. 121-125.
- [10] A. Van Vliet, An improved lower bound for on-line bin packing algorithms, *Information Processing Letters*, 1992, v. 43, pp. 277-284. 9
- [11] R. L. Graham. Bounds for certain multiprocessing anomalies. *Bell System Technical Journal*, 1966, v. 45, pp. 1563-1581. 9
- [12] S. Albers. Better bounds for online scheduling. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, 1997, pp. 130-139. 9
- [13] Y. Bartal, A. Fiat, H. Karloff, and R. Vohra, New algorithms for an ancient scheduling problem, *J. Comput. Syst. Sci.*, 1995, v. 51, N 3, pp. 359-366. 9
- [14] U. Faigle, W. Kern, and G. Tur'an. On the performance of on-line algorithms for partition problems. In *Acta Cybernetica*, 1989, v. 9, pp. 107-119.

- [15] D. R. Karger, S. J. Phillips, and E. Torng. A better algorithm for an ancient scheduling problem. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1994, pp. 132-140. 9
- [16] Y. Bartal, H. Karloff, and Y. Rabani. A better lower bound for on-line scheduling. *Information Processing Letters*, 1994, v. 50, pp. 113-116. 9
- [17] M. Caramia, S. Giordani, A. Iovanella, Grid scheduling by on-line rectangle packing, *Networks*, 2004, V. 44, N 2, pp. 106-119. 7
- [18] С.Н. Жук, Анализ некоторых эвристик в задаче упаковки прямоугольников внесколько полос, *Труды Института системного программирования*, т. 6, Математические методы и алгоритмы, 2004, с . 13-26. 9
- [19] С.Н. Жук, Приближенные алгоритмы упаковки прямоугольников в несколько полос, *Дискретная математика*, 2006, т. 18, N 1 с. 91-105.
- [20] S. Zhuk, A. Chernykh, A. Avetisyan, S. Gaissaryan, D. Grushin, N. Kuzjurin, A. Pospelov, A. Shokurov, Comparison of Scheduling Heuristics for Grid Resource Broker., *Proc. Fifth Mexican Int. Conf. in Computer Science (ENC'04)*, IEEE Computer Society, 2004, pp. 388-392.
- [21] А.И. Пospelov, Анализ одного алгоритма упаковки прямоугольников, связанного с построением расписаний для кластеров, *Труды Института системного программирования*, т. 6, Математические методы и алгоритмы, 2004, с . 7-12. 9