Анализ некоторых эвристик в задаче упаковки прямоугольников в несколько полос[1]

С. Н. Жук

Аннотация. Статья посвящена анализу некоторых эвристик в задаче упаковки прямоугольников в несколько полос. Предложен эффективный алгоритм, который размещает прямоугольники по полосам в онлайновом режиме и гарантирует константную мультипликативную точность. Это достигнуто за счёт правильной формализации понятии «допустимая полоса для прямоугольника». Показано, также, что полученная оценка точности достижима на некоторых исходных данных.

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, имеющее минимальную высоту.

Такая задача возникает, например, при распределении ресурсов в глобальных вычислительных сетях Grid, объединяющих разнородные вычислительные ресурсы (персональные компьютеры, рабочие станции, кластеры, суперкомпьютеры) и используемых для выполнения различных приложений (научных, инженерных и коммерческих). Задачу о распределении ресурсов в Grid можно представить как задачу о размещении приложений (задач) на группе кластеров. Такая модель имеет простую геометрическую интерпретацию. Кластер можно представить как бесконечную в одну строну полосу, определенной ширины (ширина характеризует число процессоров кластера), а приложение (задачу) — как прямоугольник с высотой и шириной равными времени выполнения и требуемому числу процессоров соответственно.

Известно, что даже для случая одной полосы, рассматриваемая задача является NP-трудной. Следовательно вряд ли можно ожидать, что существует эффективный алгоритм, позволяющий решить её точно. Поэтому рассматривают различные приближённые алгоритмы, которые позволяют достаточно быстро найти некоторое решение с определённой точностью.

Отметим, что в случае одной полосы — это известная и хорошо исследованная задача [1,2,3]. Для нее разработаны различные эвристики и 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 удобнее применять более простые эвристики.

[1]Работа выполнена при поддержке РФФИ, проект 02-01-00713.

В данной работе рассмотрены два достаточно просто реализуемых алгоритма, в которых прямоугольники по полосам размещаются по мере поступления. Для первого алгоритма показано, что его точность не аппроксимируется никакой константой. А для второго (модификации первого) доказано, что он гарантирует константную мультипликативную точность. Это достигнуто за счёт правильной формализации понятии «допустимая полоса для прямоугольника». Оказалось, что за счёт введения некоторых ограничений на множество полос, на которые можно размещать прямоугольники, можно существенно усовершенствовать алгоритм и достичь константной точности аппроксимации (мультипликативной). Для второго алгоритма показано, что полученную оценку мультипликативной точности нельзя улучшить.

2. Постановка задачи

Основная задача. Пусть $T = \{T_1, \ldots, T_n\}$ — конечная последовательность прямоугольников, $h(T_j)$ и $w(T_j)$ — соответственно высота и ширина прямоугольника T_j, и $C = \{C_1, \ldots, C_m\}$ — множество полубесконечных полос (кластеров), w_i — ширина i-й полосы.

Требуется 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 $H_O(T, C)$ — оптимальное значение высоты размещения, и $H_A(T, C)$ — высота размещения прямоугольников, получающегося при использовании алгоритма A.

Мы будем рассматривать следующую двухуровневую модель работы алгоритмов.

- На первом уровне для каждого прямоугольника определяется полоса, на которой он будет размещён. Этот этап размещения производится в режиме «online», то есть для каждого прямоугольника сразу определяется полоса, на которой он будет размещаться и впоследствии с приходом новых прямоугольников номер этой полосы не изменится.

- На втором уровне производится размещение прямоугольников внутри каждой полосы. Этот этап размещения производится уже не в режиме «online», то есть при добавлении нового прямоугольника внутри полосы может происходить переразмещение.

3. Алгоритм Bottom-Left

Для размещения последовательности прямоугольников внутри одной полосы в данной работе используется достаточно простая эвристика Bottom-Left(BL), детально исследованная в [1]. Алгоритм BL размещает все прямоугольники в порядке следования, так что каждый следующий прямоугольник размещается в самую левую из всевозможных самых нижних допустимых позиций. Рассмотрим некоторые его свойства.

Результаты работы этого алгоритма существенно зависят от порядка следования прямоугольников. Как показано в [1] для произвольного порядка отношение $\frac{H_A(T,C)}{H_O(T,C)}$ может получиться сколь угодно большим. Но если перед размещением список прямоугольников T отсортировать по убыванию ширины, то можно показать, что для полученного списка T_{sorted} выполняется

$$\frac{H_A(T_{sorted},C)}{H_O(T_{sorted},C)} \leq 3.$$

Далее при использовании алгоритма мы везде будем полагать, что прямоугольники упорядочены по убыванию ширины.

Обозначим также $BL_i(W)$ — высоту получающуюся при размещении с помощью алгоритма BL множества прямоугольников W в i-й полосе (с сортировкой по ширине перед размещением).

Есть ещё одно важное свойство алгоритма BL, которое нам в дальнейшем понадобиться [1].

Утверждение 1. Пусть при некотором размещении с помощью алгоритма BL упорядоченного по ширине списка прямоугольников T_a — последний размещенный прямоугольник, t_a — высота этого размещения основания прямоугольника. Тогда любое сечение этой полосы проходящее ниже t_a заполнено по крайней мере на половину.

4. Простой алгоритм

4.1. Описание *Алгоритма 1*

1. Для каждого следующего прямоугольника T_j, производится последовательное рассмотрение всех полос C_i, на которых данный прямоугольник может быть вообще размещён (т.е. $w(T_j) \leq w_i$). Для каждой полосы определяется высота размещения, которая получается при добавлении данного прямоугольника к имеющимся и применении алгоритма BL к новому отсортированному списку прямоугольников. После этого прямоугольник отправляется на одну из тех полос, у которых эта высота наименьшая.
2. После того как все прямоугольники T_j исчерпаны, в каждой полосе применяется алгоритм BL для соответствующего отсортированного списка.

4.2. Пример со сколь угодно большим отношением

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

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

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

1. Пусть $C = \bigcup_{m=1}^{k} G_m$, где
   $G_1 = \{\ C_{1,1} \mid C_{1,1}$ — полоса шириной $1\}$
   $G_m = \{\ C_{m,1}, \ldots, C_{m,2^{2m-3}} \mid C_{m,s}$ — полоса шириной $\frac{1}{2^{m-1}}\ \}$, где $m = 2, \ldots, k$.
2. Пусть $Y = \bigcup_{m=1}^{k} Y_m$
   $Y_m = \{\ Y_{m,1}, \ldots, Y_{m,2\cdot 2^{2m-3}} \mid Y_{m,s}$ — прямоугольник шириной $\frac{1}{2^{m-1}}$ и высотой $1\}$, $m = 1, \ldots, k$
3. Пусть T — последовательность прямоугольников, полученная из Y сортировкой по возрастанию ширины.

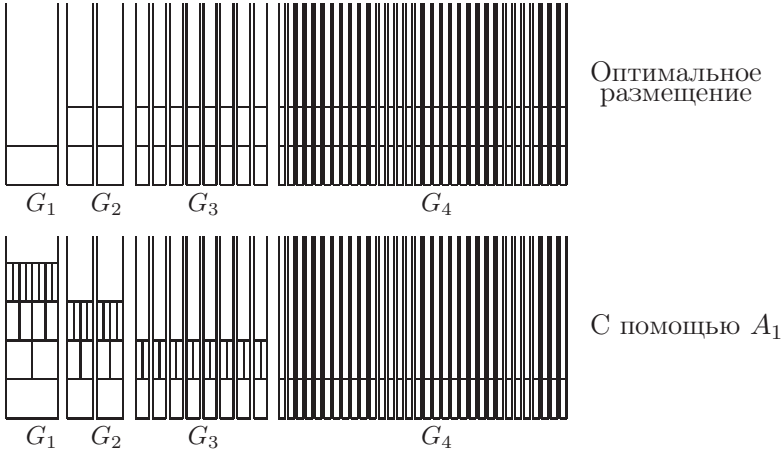

Рис. 1. Пример для *Алгоритма 1.*

Тогда, $H_O(T,C) = 2$. (Все прямоугольники из Y_m размещаем на полосы из G_m в два слоя).

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 т.д. до Y_1. Так как

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

то суммарная ширина всех прямоугольников из Y_m равна суммарной ширине всех полос из $\bigcup_{i=1}^{m} G_i$, поэтому при размещении они равномерно покроют слоем высотой 1 все эти полосы и никак не будут влиять на дальнейшее распределение прямоугольников. Поэтому $H_{A_1}(T, C) = k$, то есть $\frac{H_{A_1}(T,C)}{H_O(T,C)} = k/2$ для любого целого k. Выбирая $k \geq 2 \cdot r$ получим утверждение теоремы.

5. Усовершенствованный алгоритм

5.1. 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, т.е. для $C_1, \ldots, C_m$ выполняется $w_1 \leq w_2 \leq \cdots \leq w_m$.

Введём следующие обозначения:

$$\begin{aligned} first(T_j) &= \min i : \ w_i \geq w(T_j) \\ last(T_j) &= \min r : \sum_{i=first(T_j)}^{r} w_i \geq \frac{1}{2} \sum_{i=first(T_j)}^{m} w_i \end{aligned}$$

Определение 1. *Полосы с номерами $first(T_j)$, $first(T_j)+1$, …, $last(T_j)$ назовем* **допустимыми** *для прямоугольника T_j.*

Алгоритм 2.

1. Прямоугольники в каждой полосе размещаются с помощью алгоритма BL с предварительной сортировкой по ширине(также как и в *Алгоритме 1*).
2. Для каждого поступающего прямоугольника, производится последовательное рассмотрение только **допустимых** для него полос. Для каждой из них определяется высота размещения, которая получается при добавлении данного прямоугольника к имеющимся и размещении их в данной полосе с помощью алгоритма BL (с предварительной сортировкой). После этого прямоугольник отправляется на одну из тех полос, у которых получившаяся высота наименьшая.

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

5.2. Доказательство верхней оценки

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

Теорема 2. Для данного алгоритма $\frac{H_{A_2}(T,C)}{H_O(T,C)} \leq 10$.

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

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

Пусть $W_f, \ldots, W_l$ множества прямоугольников, которые уже были размещены на допустимых для T_a полосах $C_f, \ldots, C_l$ непосредственно перед добавление прямоугольника T_a. Так как его добавили на k-ю полосу, то

$$BL_i(W_i \cup \{T_a\}) \geq H_A(T, C) \quad i = f, \ldots, l.$$

Пусть при размещении с помощью алгоритма BL множества прямоугольников $W_i \cup \{T_a\}$ на i-ой полосе T_{c_i} — прямоугольник, имеющий максимальную верхнюю границу. И пусть $h_i = h(T_{c_i})$ и t_i — это его высота и высота его его нижней границы соответственно. Пусть $r_i = t_i - h(T_a)$, тогда

$$t_i + h_i \geq H_{A_2}(T, C) \quad i = f, \ldots, l,$$

$$r_i + h(T_a) + h_i \geq H_{A_2}(T, C) \quad i = f, \ldots, l,$$

$$\sum_{i=f}^{l} w_i r_i + h(T_a) \cdot \sum_{i=f}^{l} w_i + \sum_{i=f}^{l} w_i h_i \geq H_{A_2} \cdot \sum_{i=f}^{l} w_i. \tag{1}$$

Пусть $S(W_i)$ — суммарная площадь прямоугольников из множества W_i, тогда по свойству алгоритма BL (утверждение 1)

$$S(W_i) \geq \frac{1}{2} \cdot w_i \cdot r_i.$$

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

Т.к. T_b размещен на одной из полос $C_f, \ldots, C_l$, то

$$last(T_b) \geq f \implies \sum_{i=f_0}^{f-1} w_i \leq \tfrac{1}{2} \sum_{i=f_0}^{m} w_i \implies \sum_{i=f}^{m} w_i \geq \tfrac{1}{2} \sum_{i=f_0}^{m} w_i.$$

Т.к. $l = last(T_a)$, то

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

Поэтому получаем,

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

Далее,

$$H_O \cdot \sum_{i=f_0}^{m} w_i \geq \sum_{i=f}^{l} S(W_i) \geq \frac{1}{2} \sum_{i=f}^{l} w_i r_i,$$

$$\sum_{i=f}^{l} w_i r_i \leq 2 \cdot H_O \cdot \sum_{i=f_0}^{m} w_i \leq 8 \cdot H_O \cdot \sum_{i=f}^{l} w_i. \qquad (2)$$

Так как $H_O \geq h(T_j)$, $\forall j$, то из неравенств (1) и (2) получаем

$$8 \cdot H_O \cdot \sum_{i=f}^{l} w_i + h(T_a) \cdot \sum_{i=f}^{l} w_i + H_O \cdot \sum_{i=f}^{l} w_i \geq H_A \cdot \sum_{i=f}^{l} w_i.$$

Следовательно,

$$8 \cdot H_O + h(T_a) + H_O \geq H_{A_2},$$

$$8 \cdot H_O + H_O + H_O \geq H_{A_2},$$

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

6. Нижняя оценка для *Алгоритма 2*

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

Теорема 3. Для любого $\varepsilon_0 > 0$ существует такое множество полос и прямоугольников, и такой их порядок следования, для которого

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

6.1. Пример шахматной доски

Построим вспомогательный пример размещения прямоугольников, аналогичный «the checkerboard example» из [1].

Пусть у нас есть

1. n прямоугольников $T_1, \ldots, T_n$, с $w(T_j) = h(T_j) = 2 - j \cdot \varepsilon$;
2. N_s прямоугольников с шириной и высотой равными 1;
3. 1 полоса шириной 2n.

При использовании алгоритма BL эти прямоугольники будут размещены так, как показано на рисунке 2.

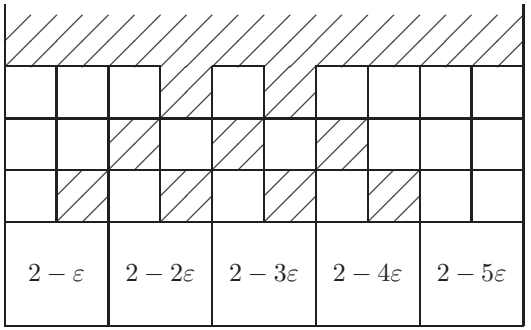

Рис. 2. Пример шахматной доски

Пусть k — количество слоев в этом размещении из маленьких прямоугольников. Будем считать n и N_s выбраны так, что последний слой заполнен полностью и выполняется соотношение $2 \cdot n = k^2$.

Суммарное количество всех маленьких прямоугольников не превосходит

$$N_s \leq k^3/2 + O(k^2).$$

6.2. Основной пример

- **Некоторые обозначения. Выбор параметров.**

  1. Выберем какое-нибудь $\delta > 0$ и построим вспомогательный пример «шахматной доски» такой, что
     $k = 8q$, где q — целое число.
     $1/k < \delta$,

$N_s < k^3(1/2+\delta)$,

$\varepsilon < \delta$.

Обозначим $h_0 = k/8 = q$ и $w_0 = 2 \cdot n = k^2$.

2. Выберем некоторое положительное целое число m, такое что $k/m < \delta$.
3. Выберем $\varepsilon_1 > 0$ так, что $2 \cdot n \cdot \varepsilon_1 < \varepsilon$ и $16n \cdot m^2 \cdot \varepsilon_1 < 1$.
   Обозначим $w_i = 1 - \varepsilon_1 \cdot (m - i)$, для $i = 1, \ldots, m$.
4. Выберем $\varepsilon_2 > 0$ таким что $m\varepsilon_2 < \varepsilon_1$.
   Обозначим $w'_i = w_i/2 + \varepsilon_2$.
5. Пусть $w_d = 1/4$.

- **Опишем полосы, используемые в данном примере.**

1. $2m$ полос шириной w_0. Обозначим их $A_1, \ldots, A_{2m}$.
2. $m-1$ множеств полос, $B_1, \ldots, B_{m-1}$.
   B_i — состоит из $n_b = 2 \cdot 2n = 4n$ полос шириной w_i.
   Пусть N_b — суммарное количество полос во всех этих множествах ($N_b = 4n \cdot (m-1)$).
3. $m-1$ множеств $C_1, \ldots, C_{m-1}$.
   C_i — состоит из $n_c = 2 \cdot n_b$ полос шириной $w'_i = w_i/2 + \varepsilon_2$.
   Суммарное количество полос в этих множествах $N_c = 2 \cdot N_b = 4n \cdot (m-1)$.
4. Одна полоса D шириной w_d.

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

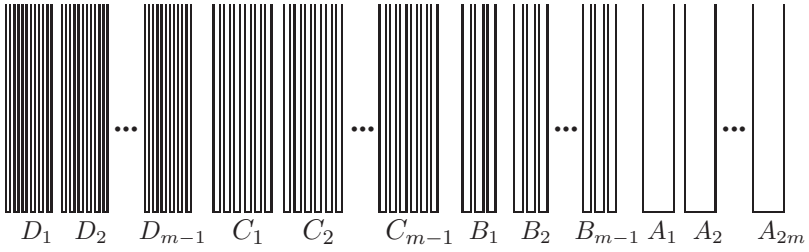

Рис. 3. Полосы основного примера.

- **Опишем прямоугольники, используемые в примере.**

1. Множество P состоит из N_c прямоугольников шириной $\frac{w_d}{N_C}$ и высотой h_0.
2. $m-1$ множеств прямоугольников $R_1, \ldots, R_{m-1}$.
   R_i состоит из $2 \cdot n_b$ прямоугольников шириной w'_i и высотой h_0.
3. Множество Q состоит из m прямоугольников, причём
   i-й имеет ширину $w_i - \varepsilon_1/2$ и высоту h_0.
4. m множеств прямоугольников $X_1 \ldots X_m$, где
   X_i состоит из $N_x = 2 \cdot (k/8) \cdot 2 \cdot n = k^3/4$ прямоугольников шириной $w_i = 1 - (m-i) \cdot \varepsilon_1$ и высотой 1.
5. m множеств прямоугольников $Y_1 \ldots Y_m$.
   Y_i состоит из $N_y = N_s - N_x$ прямоугольников шириной 1 и высотой 1.
6. m множеств $U_1, \ldots, U_m$.
   U_i состоит из n прямоугольников,
   j-й прямоугольник U_i имеет ширину $2 - j \cdot \varepsilon$ и высоту $2 - j \cdot \varepsilon$.
7. V — один прямоугольник шириной w_0 и высотой h_0.

- **Некоторые общие замечания.**

1. Для прямоугольников $V, U_1 \ldots U_m, Y_1 \ldots Y_m, X_m$ допустимыми являются полосы $A_1, \ldots, A_m$ 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 P полосы из $C_1, \ldots, C_{m-1}$ являются допустимыми.
5. Для прямоугольников из R_i все полосы из B_i являются допустимыми.
6. Полосы $A_{m+1}, \ldots, A_{2m}$ не являются допустимыми ни для одного из прямоугольников.
7. Так как

$$\frac{w_d}{2N_c} + w'_i = \frac{1}{4 \cdot 8n \cdot (m-1)} + \frac{1}{2} \cdot (1 - (m-i) \cdot \varepsilon_1) + \varepsilon_2 >$$

$$> \frac{1}{2} + \frac{1}{4 \cdot 8n \cdot m} - \frac{1}{2} \cdot m \cdot \varepsilon_1 \geq \frac{1}{2} + \frac{1 - 16nm^2 \cdot \varepsilon_1}{32nm} > \frac{1}{2},$$

то в полосы из $C_1, \ldots, C_m$ не влезут в один слой, прямоугольник из P и прямоугольник из одной из групп $R_1, \ldots, R_m$.

8. Так как ширина прямоугольников из X_i незначительно отличается от 1, то при их размещении в полосах $A_1, \ldots, A_m$ расположение прямоугольников практически не будет отличатся от расположения в примере «шахматной доски».

6.3. Размещение с помощью *Алгоритма 2*

Так как размещение прямоугольников, получающееся при использовании алгоритма, существенно зависит порядка в котором они поступают, то мы можем выбрать такой порядок, при котором рассматриваемое отношение получится наибольшим. При этом этом будем считать, что в случае равенства минимальных высот мы можем выбрать полосу, на которую будет отправлен прямоугольник.

1. Сначала приходят прямоугольники из P, они размещаются на полосы из $C_1, \ldots, C_{m-1}$, по одному прямоугольнику на каждую полосу.
2. Далее для каждого $i = 1, \ldots, m$ приходят N_b прямоугольников из R_i и они размещается на полосы из B_i по одной на полосу.
3. Далее приходят прямоугольники из $X_1, \ldots, X_m$. И приходят они в таком порядке, чтобы прямоугольники из X_i заполняли полосу A_i. Так должно продолжаться до тех пор пока полосы $A_1, \ldots, A_m$ не будут заполнены до высоты h_0.
4. Далее для каждого $i = 1, \ldots, m$ приходят оставшиеся N_b прямоугольников из R_i. Они размещаются на полосах из B_i вторым слоем.
5. Далее приходят оставшиеся прямоугольники из $X_1, \ldots, X_m$. Они опять приходят в таком порядке, чтобы прямоугольники из X_i заполняли полосу A_i. Теперь полосы $A_1, \ldots A_m$ заполнены до высоты $2 \cdot h_0$.
6. Затем приходят прямоугольники из Q, i-й прямоугольник размещается на полосу A_i.
7. Далее приходят все прямоугольники из $Y_1, \ldots, Y_m$. Они равномерно распределятся по полосам $A_1, \ldots, A_m$. Они все одинаковые, поэтому порядок их прихода не важен.
8. Далее приходят прямоугольники из $U_1, \ldots, U_m$. Для каждого $j = 1, \ldots, n_0$ приходят j-е прямоугольники для всех U_i и j-й прямоугольник U_i размещается на полосу A_i.
9. И последним приходит прямоугольник V. Он размещается на любую из полос $A_1, \ldots, A_m$.

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

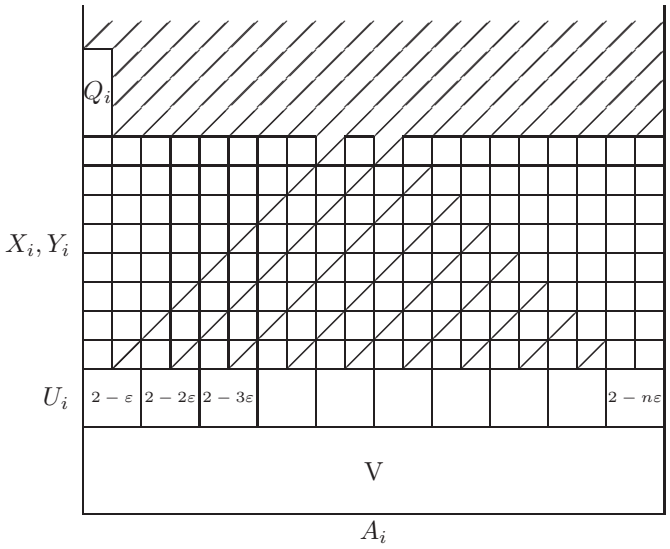

Рис. 4. Размещение с помощью *Алгоритма 2*.

Найдём высоту полученного размещения:

$$H_{A_2} = h_0 + 2 - \varepsilon + k + h_Q > h_0 + 8 \cdot h_0 + h_0 = 10 \cdot h_0.$$

6.4. Оптимальное размещение для основного примера

Для доказательство нижней оценки мы необязательно должны размещать прямоугольники оптимальным образом, достаточно получить какое-нибудь размещение близкое к оптимальному, позволяющее доказать требуемую оценку.

Размещение будет следующим:

1. Все прямоугольники из P размещаем на полосу D.
2. Для каждого $i = 1, \ldots, m-1$ все прямоугольники из R_i 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

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

4. i-й прямоугольник из Q размещаем на полосу A_i, так чтобы он был расположен в левом нижнем углу полосы.

5. Прямоугольник V размещаем на A_{2m}.

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 (просто 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 сверху уже размещенных там прямоугольников.

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

Оценим высоту получившегося размещения. Количество размещенных на последнем шаге прямоугольников из $Y_1, \ldots, Y_m$

$$N_l = N_x + m \cdot N_y - (k/8) \cdot ((m-1) \cdot 2n + m \cdot (2n-1)) =$$

$$= m \cdot N_y - m \cdot k^3/4 + k^3/4 + (k/8) \cdot (2n+m)$$

$$N_y = N_s - N_x = N_s - k^3/4 \le k^3(1/4 + \delta)$$

$$N_l \le mk^3\delta + k^3/4 + (k/8) \cdot (2n+m).$$

Высота, которую займут эти прямоугольники не превзойдет

$$H_l \le \frac{N_1}{2 \cdot n \cdot 2 \cdot m} + 1 \le \frac{m \cdot k^3 \cdot \delta + k^3/4 + (k/8) \cdot (2n+m)}{2 \cdot m \cdot k^2} + 1 \le$$

$$\le \frac{k\delta}{2} + \frac{3k}{16m} + \frac{1}{16k} + 1 \le k\delta + 1.$$

Суммарная высота размещения не превзойдет

$$H_O \le h_0 + 2 + H_l \le h_0 + 3 + k\delta \le h_0 \cdot (1 + 24/k + 8\delta) \le h_0 \cdot (1 + 32\delta).$$

Итак,

$$\frac{H_{A2}}{H_O} \ge \frac{10}{1 + 32\delta} \ge 10 \cdot (1 - 32\delta).$$

То есть, если положить $\delta < \varepsilon_0/320$, получим утверждение теоремы 3

$$\frac{H_{A2}}{H_O} \ge 10 - \varepsilon_0.$$

7. Заключение

В данной работе получены некоторые результаты для задачи об упаковке прямоугольников в несколько полос. Рассмотрено два похожих алгоритма и показано, что точность первого не аппроксимируется никакой константой, а второй алгоритм – гарантирует константную мультипликативную точность. Это достигнуто за счёт правильной формализации понятия «допустимый кластер для задачи». Поскольку задача достаточно новая, то здесь остаётся ещё много неисследованных вопросов. Например, было бы интересно получше рассмотреть offline алгоритмы. Хотя в этом случае уже имеются некоторые результаты, но пока неизвестно, можно ли данную задачу, как в случае одной полосы, решить с точностью $1 + \varepsilon$ для любого $\varepsilon > 0$.

Также следует заметить, что оценки, полученные в данных алгоритмах, достаточно велики, поэтому можно попытаться разработать какие-нибудь более эффективные алгоритмы. И кроме того, хотя оценка 10 и достижима (как показано в теореме 3), она требует очень специального набора входных данных, который вряд ли может встретиться в реальности. Поэтому было бы интересно исследовать, как данные алгоритмы будут вести себя в среднем.

Литература

[1] B.S. Baker, E.J. Coffman and R.L. Rivest, Orthogonal packings in two dimensions, SIAM J. Computing, 1980, v. 9, pp. 846-855.

[2] C. Kenyon and E. Remila, A near optimal solution to a two-dimensional cutting stock problem, Mathematics of Operations Research, 25 (2000), 645-656.

[3] 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.

[4] K. Jansen, Scheduling malleable parallel tasks: an asymptotic fully polynomial-time approximation scheme, Proc. European Symposium on Algorithms, ESA, 2002.

[5] P. Brucker, Scheduling Algorithms, Springer Ferlag (1998), 217-218.

[6] L. Hluchy, V.D. Tran, D. Froehlich, and W. Castaings, Methods and Experiences of Parallelizing Flood Models, The 10th EuroPVM/MPI conference. LNCS 2840. Sept. 2003, Venice. pp. 677-681.

[7] E.J. Coffman, M.R. Garey, D.S. Johnson and R.E. Tarjan, Performance bounds for level-oriented two-dimensional packing algorithms, SIAM J. Computing, 1980, v. 9, pp. 808-826.

[8] M. Drozdowski, Scheduling multiprocessor tasks - an overview, European J. of Oper. Research, 1996, v. 94, pp. 215-230.