Об онлайновых алгоритмах для задач упаковки в контейнеры и полосы, их анализе в худшем случае и в среднем

[1,2] *Д.О. Лазарев* <dennis810@mail.ru>
[1,2] *Н.Н. Кузюрин* <nnkuz@ispras.ru>
[1] *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25*
[2] *Московский физико-технический институт, 141700, Московская область, г. Долгопрудный, Институтский пер., 9*

Аннотация. В работе рассмоторены онлайновые алгоритмы для классических задач упаковки Bin Packing и Strip Packing и их обобщений: задач Multidimensional Bin Packing, Multiple Strip Packing и задаче об упаковке в полосы различной ширины. Для последней задачи описан анализ в худшем случае; для остальных задач приведен как анализ в худшем случае, так и анализ в среднем (вероятностный анализ). Рассмотрены лучшие известные нижние и верхние оценки. Приведены основные алгоритмы и описаны методы их анализа.

Ключевые слова: Bin Packing; Multidimensional Bin Packing; Strip Packing; Multiple Strip Packing; задача об упаковке в полосы различной ширины; вероятностный анализ; анализ в худшем случае

DOI: 10.15514/ISPRAS-2018-30(4)-14

Для цитирования: Лазарев Д.О., Кузюрин Н.Н. Об онлайновых алгоритмах для задач упаковки в контейнеры и полосы, их анализе в худшем случае и в среднем. Труды ИСП РАН, том 30, вып. 4, 2018 г., стр. 209-230. DOI: 10.15514/ISPRAS-2018-30(4)-14

1. Введение

В последние годы заметно повысился интерес к задачам оптимизации в различных производственных и логистических процессах [1,2,3,11]. Для популярных в настоящее время задач анализа больших данных часто используются облачные вычисления, которые также требуют решения задач оптимизации. Важную роль в такого рода задачах оптимизации играет теория расписаний и, в частности, различные классы задач упаковки [4,5].

Интерес к задачам упаковки всегда стимулировался их многочисленными практическими приложениями. Так, одномерная задача упаковки в контейнеры

(bin packing) возникла в силу потребности решения задач раскроя и перевозки материалов. Двумерная задача упаковки в контейнеры хорошо моделировала задачу оптимизации размещения объектов, например, автомобилей, в вагонах, паромах и т.п., а также размещения микросхем на платах в задачах построения СБИС. Трехмерная задача упаковки учитывала еще одно измерение и хорошо описывала оптимизацию размещения трехмерных объектов на складах и т.п.

В настоящее время в силу быстрого роста популярности распределенных вычислений, широкого распространения вычислительных кластеров, грид-технологий а также облачных вычислений этот интерес к задачам упаковки возрастает в связи с новыми приложениями: задачами управления ресурсами распределенных вычислительных систем, развития техники облачных вычислений [6,7,8,12,35,41]. При этом возникают новые классы задач упаковки, в частности, задачи упаковки прямоугольников в полосу и несколько полос, задачи упаковки приложений в виртуальные машины и контейнеры и т.п. В данной работе мы ставим целью описать классические результаты, касающиеся различных задач упаковки, а также привести ряд новых результатов, полученных в самое последнее время.

2. Постановка задачи

2.1 Bin Packing

Определим задачу Multidimensional Bin Packing Problem (Multidimensional BPP), или задачу об упаковке в контейнеры размерности n в постановке упаковки в ящик (Box Packing): дан набор d-мерных открытых прямоугольных параллелипипедов, длины сторон которых не превосходят 1. Требуется упаковать параллелипипеды без вращений и пересечений в как можно меньшее число d-мерных кубов с единичной стороной.

Например, в случае одномерной задачи Bin Packing (BP) требуется распределить объекты в минимальное количество контейнеров так, чтобы суммарный вес объектов в каждом контейнере не превышал 1.

Задача является NP-трудной, поэтому будем рассматривать приближённое решение. Будем рассматривать случай онлайновой упаковки, когда алгоритм получает параллелипипеды поочередно, и размещение каждого следующего объекта не влияет на положение предыдущих. Существуют два способа анализа таких алгоритмов.

- Анализ в худшем случае, или *Worst Case* анализ.

  Здесь эффективность алгоритма A на наборе параллелипипедов σ оценивается через асимптотическую мультипликативную точность

  $$R_A^\infty = \limsup_{n\to\infty}\ \sup_{\sigma}\left\{\frac{cost_A(\sigma)}{cost(\sigma)}\middle|cost(\sigma)\ge n\right\},$$

  где $cost_A(\sigma)$ – число кубов, занятых параллелипипедами из σ, при их упаковке алгоритмом A; $cost(\sigma)$- минимальное возможное число занятых кубов при упаковке набора параллелепипедов.

- Анализ в среднем, или *Average Case* анализ.

  В предположении, что длины сторон параллелепипедов имеют распределение F, чаще всего – равномерное распределение на некотором отрезке $[0, u]$, $u \leq 1$, нужно оценить математическое ожидание суммарного объема площади контейнеров, не заполненной параллелепипедами, $W_A^n = E_{\mathrm{F}} V_A^n$, после выпадения (не более чем) n параллелепипедов

При анализе в худшем случае исследуются лишь алгоритмы, которым неизвестно количество объектов до выпадения последнего из них. При анализе в среднем можно анализировать эффективность алгоритмов, когда число параллелепипедов известно до начала работы (*closed-end*) и когда число параллелепипедов становится известным лишь тогда, когда все они выпали (*open-end*). В случае *open-end* W_A^n вычисляется после выпадения ровно n параллелепипедов.

В [24] Шором было ослаблено условие *open-end* и было дано следующее описание работы алгоритмов *open-end по Шору*:

- натуральное число n известно алгоритму до начала работы;
- выбирается случайно натуральное $k, 1 \leq k \leq n$;
- на вход подается k объектов со случайными длинами сторон и после них – символ останова.

Оценивается математическое ожидание W_A^n объёма контейнеров, не заполненного после упаковки всех k объектов.

2.2 Strip Packing и Multiple Strip Packing

В зависимости от количества полос задачи разделяются на следующие:

- Strip Packing(SP): дана полубесконечная полоса единичной ширины;
- Multiple Strip Packing(MSP): дан набор полубесконечных полос $C = \{C_1, \dots, C_m\}$ единичной ширины.

В эти полосы требуется упаковать без вращений и пересечений n открытых прямоугольников $T = \{T_1, \dots, T_n\}$, минимизируя при этом высоту упаковки. Высота упаковки – координата самой высокой верхней стороны одного из прямоугольников. Исследуются онлайновые алгоритмы упаковки, то есть алгоритмы, получающие прямоугольники из T последовательно, и размещающие T_i до получения $T_{i+1}, \dots, T_n$ для любого $i \leq n - 1$.

Как и в случае задачи BP, существует два способа анализа алгоритмов упаковки:

- Анализ в худшем случае, или *Worst Case* анализ.

  Здесь эффективность алгоритма A на наборе прямоугольников T оценивается через его асимптотическую мультипликативную точность

$$R_A^\infty = \limsup_{n \to \infty} \sup_{T} \left\{ \frac{H_A(T)}{H_O(T)} \middle| H_O(T) \geq n * h_{max} \right\},$$

  где $H_A(T)$ – высота упаковки прямоугольников из T при их упаковке алгоритмом A; $H_O(T)$- минимальная возможная высота упаковки прямоугольников из T, h_{max}-максимальная высота прямоугольника. Анализ проводится в предположении, что h_{max} известно алгоритму до начала работы

- Анализ в среднем, или *Average Case* анализ.

  В предположении, что длины и высоты прямоугольников имеют распределения F_i и F_w, чаще всего – равномерные распределения на некоторых отрезках $[0, u]$ и $[o, v]$ соответсвенно, нужно оценить математическое ожидание суммарной площади части полос от основания до высоты упаковки, не заполненной прямоугольниками $W_A^n = E_{\mathrm{F}_l \mathrm{F}_w} S_A^n$, после выпадения n прямоугольников.

При оценке в среднем эффективность алгоритмов анализируется, как и для задачи BP, в *open-end* и *close-end* случаях.

3 Задача Bin Packing

Задача Bin Packing или задача упаковки в контейнеры – одна из первых известных NP-трудных в сильном смысле задач [9]. Для нее было составлено множество приближённых алгоритмов для анализа в худшем и в среднем случаях [10].

Опишем важные методы онлайновой упаковки.

First Fit. Каждый следующий поступивший объект попадает в последний созданный контейнер, в который он помещается. Если же он не помещается ни в один контейнер, то создаётся новый контейнер, в котором объект и оказывается.

Best Fit. Каждый следующий объект попадает в наиболее плотно заполненный контейнер, в который он может поместиться. Если объект никуда не помещается, то для его упаковки создаётся новый контейнер.

3.1 Анализ в худшем случае

Следуя классической работе [11], покажем, что асимптотическая точность алгоритма First Fit $R_{FF}^\infty = \frac{17}{10}$.

Пусть на вход алгоритму First Fit поступило в порядке возрастания индексов k объектов $a_1, \dots, a_k$ размерами $s(a_1), \dots, s(a_k)$, образующих набор σ. Алгоритм First Fit их размещает в контейнеры $B_1, \dots, B_n$; номера контейнеров возрастают в порядке их создания.

Определение: грубостью α_i i-го контейнера B_i назовём максимальную незаполненную длину контейнера с номером, меньшим i.

Ниже $cost_A(\omega)$ обозначает число контейнеров, в которые алгоритм A размещает объекты из набора ω, а OPT – оптимальный алгоритм.

Теорема 1 (Johnson, Demers, Ullman, Garey, Graham, 1974) [11]. Для любого набора σ, верно, что $cost_{FF}(\sigma) \le \frac{17}{10} cost_{OPT}(\sigma) + 2$.

В доказательстве используется весовая функция $W(a)$. Для доказательства теоремы достаточно статической весовой функции, т.е. вес объекта зависит лишь от его размера. Функция из работы [11] проиллюстрирована ниже:

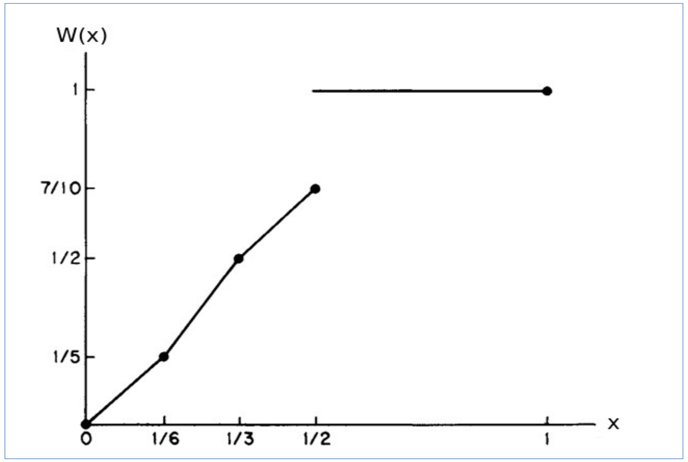

Рис. 1. Весовая функция
Fig. 1. Weighting function

Доказательство теоремы опирается на две Леммы:

Лемма 1: $\sum_{i=1}^{n} W(s(a_i)) \le \frac{17}{10} cost_{OPT}(\sigma)$.

Лемма 2: $\sum_{i=1}^{n} W(s(a_i)) \ge cost_{FF}(\sigma) - 2$.

Для доказательства Леммы 2 нужны два утверждения.

Утверждение 1: Если контейнер с грубостью $\alpha \le \frac{1}{2}$ заполнен объектами с весами $b_1, .., b_l$, и если $\sum_{i=1}^{l} b_i > 1 - \alpha$, то $\sum_{i=1}^{l} W(b_i) \ge 1$.

Утверждение 2: Если контейнер с грубостью $\alpha \le \frac{1}{2}$ заполнен объектами с весами $b_1 \ge .. \ge b_l$ и $\sum_{i=1}^{l} W(b_i) = 1 - \beta$, то либо $m = 1$ и $b_1 < \frac{1}{2}$, либо $\sum_{i=1}^{l} b_i \le 1 - \alpha - \frac{5}{9}\beta$.

Далее, пусть для контейнеров $B_1', \dots, B_m'$ и только для них $\sum_{a_j \in B_i} W(s(a_j)) = 1 - \beta_i, \beta_i > 0$. Для любых $1 \le i < j \le m$ контейнер B_j' был создан позже B_i'. Так как в B_m' нет элемента размером $s(a_i) \ge \frac{1}{2}$, то грубость m-го контейнера $\alpha_m \le \frac{1}{2}$. По утверждению 2, $\alpha_i \ge \alpha_{i-1} + \frac{5}{9}\beta_{i-1}$, следовательно,

$\sum_{i=1}^{m-1} \beta_i \ge \frac{9}{5} \sum_{i=1}^{m-1} (\alpha_{i+1} - \alpha_i) = \frac{9}{5}(\alpha_m - \alpha_1) < \frac{9}{5} * \frac{1}{2} < 1.$

Из предыдущего неравенства и оценки $\beta_m \le 1$, имеем требуемое утверждение:

$$\sum_{i=1}^{n} W(s(a_i)) \ge cost_{FF}(\sigma) - 2$$

Отсюда получаем, что $R_{FF}^{\infty} \le 1.7$.

Для получения равенства достаточно привести пример сколь угодно большого набора σ с $cost_{FF}(\sigma) = \frac{17}{10} cost_{OPT}(\sigma)$ (рис. 2).

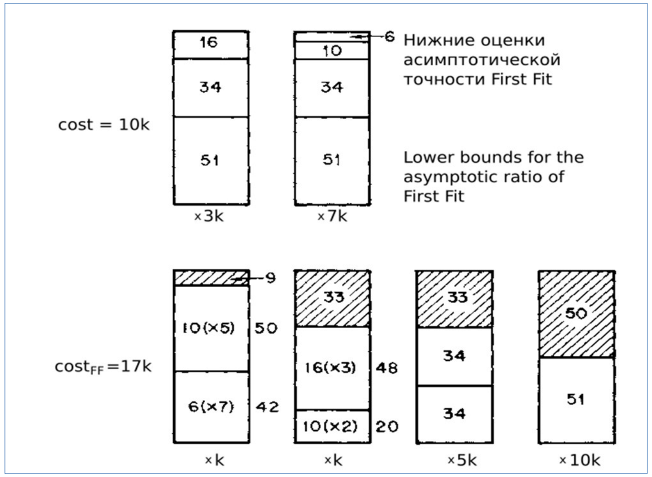

Рис. 2. Точная нижняя оценка First Fit
Fig. 2. Strict lower bounds for First Fit

Отметим, что первоначально в работе [12] в правой части леммы 2 вместо «-2» было «-3». Позднее в работе [13] «-2» было улучшено до «-1».

Заметим, что алгоритм First Fit создаёт новый контейнер тогда и только тогда, когда ни в одном из уже созданных контейнеров не достаточно места для размещения вновь прибывшего объекта. Скажем, что алгоритм принадлежит семейству **Any Fit**, если он удовлетворяет этому условию. В работе [10] доказано, что любой алгоритм AF из Any Fit имеет асимптотическую точность не лучше, чем First Fit.

Теорема 2 (Johnson, 1973) [10]. $R_{AF}^{\infty} \ge R_{FF}^{\infty}$ для любого Any Fit алгоритма AF.

Таким образом, при дальнейшем анализе можно либо ослабить требование онлайновости алгоритма, как, например, было сделано в работах [15] и [16], либо исследовать алгоритмы не из Any Fit, как сделал Яо в своей работе [14].

Скажем, что объект – типа A, если длина его лежит в интервале $(\frac{1}{2}, 1]$; типа B_1, если в $(\frac{2}{5}, \frac{1}{2}]$; типа B_2, если в $(\frac{1}{3}, \frac{2}{5}]$; типа X, если в $\left(0, \frac{1}{3}\right]$.

Алгоритм 1: Refined First Fit (RFF, Yao, 1981) [14]**.**

Пакуем объект каждого типа, кроме каждого шестого объекта типа B_2, в отдельный класс по алгоритму First Fit. Каждый шестой объект типа B_2 пакуем также по алгоритму First Fit в класс, где лежат все объекты типа A.

В [14] было также показано, что $R_{RFF}^{\infty} = \frac{5}{3}$. В дальнейшем был предложен целый ряд алгоритмов с уменьшенной асимптотической точностью; самый лучший из известных – алгоритм Harmonic++ из работы [17] с $R^{\infty} = 1.58889$. Однако,в среднем такие алгоритмы имеют асимптотику $W_A^n = \theta(n)$ и на практике редко используются.

В работе [14] было также показано, что для любого онлайнового алгоритма $R^{\infty} \geq \frac{3}{2}$. Было выбрано $0 < \varepsilon < 0.01$, и на вход алгоритму сначало поступало k объектов размером $\frac{1}{6} - 2\varepsilon$, затем – k объектов размером $\frac{1}{3} + \varepsilon$ и k объектов размером $\frac{1}{2} + \varepsilon$. Было доказано, что либо после упаковки k объектов, либо после упаковки $2k$ объектов, либо после упаковки всех $3k$ объектов число занятых контейнеров будет в $\frac{3}{2}$ раза больше числа контейнеров, занятых оптимальной упаковкой тех же объектов.

В работе [18] приведена лучшая оценка, чем в [14]: показано, что для любого онлайнового алгоритма $R^{\infty} \geq 1.536$... Лучшая из известных оценок $R^{\infty} \geq 1.540$.. была получена в работе [19] фон Влиетом.

3.2 Анализ в среднем

Часто нас интересует не эффективность работы алгоритма в худшем случае, а его среднее поведение. Известны алгоритмы упаковки при различных распределениях размеров упаковываемых объектов [20], однако в этом обзоре мы ограничимся равномерным распределением на отрезке $[0, v], v \leq 1$.

3.2.1 Размеры объектов имеют равномерное распределение на $[0, 1]$

Рассмотрим open-end случай, то есть случай, когда алгоритму неизвестно число объектов, которые он должен упаковать, до выпадения последнего из них.

В работе [23] была улучшена оценка модификации [22] теоремы из [21]:

Теорема 3 (Coffman, Shor, 1991) [23]. Если в квадрате случайно выбраны $2n$ точек, обозначим все точки на “+” или “-“ так, что у любой точки вероятности быть обозначенной плюсом и минусом равны. Пусть выбрано максимальное число пар “+ -“ так, что плюс в каждой паре правее минуса, и суммарная длина отрезков T, соединяющих плюс с минусом в каждой паре, минимальна. Тогда математическое ожидание $E(T) = O(\sqrt{n}\log^{\frac{3}{4}} n)$.

В работах [24] и [25] с помощью модификации этой теоремы была получена асимптотика для задачи Best Fit.

Теорема 4 (Shor, Leighton, 1989) [24, 25]. Математическое ожидание незаполненного пространства контейнеров при использовании алгоритма Best Fit $W_{BF}^n = \theta(\sqrt{n}\log^{\frac{3}{4}} n)$.

Приведём конструкцию для доказательства данной оценки.

Сначала каждому выпавшему объекту a_i, где $i = 1, \ldots, N$, ставится в соответствие точка в квадрате $[0,1] * [0,1]$: координата $x = s(a_i)$, координата $y = \frac{i}{N}$.

Плюсами помечаются объекты размером $s(a_i) \geq \frac{1}{2}$, а все остальные – минусами.

Затем весь квадрат отражается относительно прямой $x = \frac{1}{2}$ так, что плюсы и минусы попадают в образовавшийся прямоугольник $\left[0, \frac{1}{2}\right] * [0,1]$. Далее применяется теорема 3 для алгоритма MBF, или изменённого алгоритма Best Fit. Данный алгоритм отличается от Best Fit лишь тем, что контейнер закрывается после того, как в него попадает объект размером меньше $\frac{1}{2}$.

В работах [24] и [26] была оценена асимптотика алгоритма First Fit:

Теорема 5 (Coffman et al., 1991) [24, 26]. Математическое ожидание незаполненного пространства контейнеров при использовании алгоритма First Fit $W_{BF}^n = \theta(n^{\frac{2}{3}})$.

В работе [24] также была получена нижняя оценка W_A^n для любого онлайнового алгоритма:

Теорема 6 (Shor, 1986) [24]. Для любого онлайнового open-end алгоритма A математическое ожидание незаполненного пространства $W_A^n = \Omega(\sqrt{n\log n})$.

В работе [27] Шором был получен онлайновый open-end по Шору алгоритм с $W_A^n = O(\sqrt{n\log n})$.

Далее рассмотрим closed-end случай, когда алгоритм знает количество объектов, которые нужно упаковать до выпадения первого объекта a_1.

В работе [24] был предложен следующий алгоритм для упаковки n прямоугольников в режиме closed-end с $W_A^n = \theta(\sqrt{n})$:

1) для каждого из первых $\left\lceil \frac{n}{2} \right\rceil$ объектов создаём новый контейнер;
2) оставшиеся объекты пакуем алгоритмом Best Fit.

Таким образом, в closed-end случае нижняя оценка $W_{OPT}^n = \Omega(\sqrt{n})$ оказывается точной.

3.2.2 Размеры объектов имеют равномерное распределение на $[0, u], u < 1$

В [26] была получена нижняя оценка математического ожидания незаполненного пространства контейнеров:

Теорема 7 (Coffman et al, 1991) [26]. Пусть $u < 1$, размеры объектов выбираются попарно независимо согласно равномерному распределению на $[0, u]$ и A – онлайновый open-end алгоритм упаковки. Тогда математическое ожидание свободного пространства контейнеров $W_A^n = \Omega(\sqrt{n})$.

На данный момент неизвестны алгоритмы, на которых эта оценка достигается.

4. Многомерная задача Bin Packing

Многомерная задача Bin Packing в постановке упаковки в контейнеры Box Packing является естественным обобщением одномерной задачи Bin Packing (BPP). Как и BPP, она NP-трудна в сильном смысле, и для неё рассматриваются приближенные онлайновые алгоритмы.

4.1 Анализ в худшем случае

В работе [28] были получены нижние оценки для асимптотической точности онлайновых алгоритмов для многомерной задачи Bin Packing: для двухмерного случая $R_A^\infty \geq 1.802$, для трёхмерного случая $R_B^\infty \geq 1.974$ для любых онлайновых алгоритмов A и B.

Лучшая верхняя оценка для двумерной задачи была получена в [29], где предложен алгоритм с $R_B^\infty \leq 2.554$

Ксириком и фон Влиетом в [30] для d-мерной задачи был построен онлайновый алгоритм с асимптотической точностью $R_A^\infty = (\Pi_\infty)^d$, где $\Pi_\infty \approx 1.691$.

Важной для приложений характеристикой онлайнового алгоритма является открытое количество ящиков в каждый момент времени. Если в какой-то момент времени ящик закрыт, то после этого в него нельзя упаковать объект. Если в каждый момент времени число открытых ящиков не превышает некоторого наперед заданного числа k, то говорим, что алгоритм использует ограниченное пространство (bounded space).

В [31] Эпштейном и фон Сти был придуман алгоритм, использующий ограниченное пространство с такой же точностью $R_A^\infty = (\Pi_\infty)^d$. Кроме того, было доказано, что не существует алгоритма, использующего ограниченное пространство и имеющего лучшую асимптотическую точность.

4.2 Анализ в среднем

Ещё меньше результатов было получено при анализе в среднем. В работе [32] был предложен алгоритм Hash-Packing для d-мерной упаковки с точностью $W_A^n = O(n^{\frac{d+1}{d+2}})$ для длин сторон параллелипипедов, равномерно распределённых на [0,1]. Идея алгоритма заключается в нахождении среди поступающих d-мерных параллелипипедов групп параллелипипедов, достаточно плотно дополняющих друг друга до контейнера, и в отдельной упаковке групп параллелипипедов.

Алгоритм 2: Hash Packing (Chang, Wang, Kankanhalli, 1993) [32].

1. Выберем натуральное $m = m(n)$: $\exists N, \varepsilon, \delta > 0\ \forall n > N: n^\varepsilon < m < n^{\frac{1}{d}-\delta}$.
2. Скажем, что d-мерные прямоугольные параллелипипеды x_1 и x_2 лежат в одной группе, если для любых их соответствующих сторон $h_i(x_1)$ и $h_i(x_2)$ верно одно из условий:
   - на $[h_i(x_{1(\text{или } 2)}), h_i(x_{2(\text{или } 1)}))$ нет точек вида $\frac{j}{2m}, j = 1, \dots, 2m-1$;
   - на $[h_i(x_{1(\text{или } 2)}), 1 - h_i(x_{2(\text{ или } 1)}))$ имеется ровно одна точка вида $\frac{j}{2m}$, $j = 1, \dots, 2m-1$.

3. У каждого прямоугольника в контейнере любой группы, в зависимости от длин его сторон, есть его место и, если в одном из контейнеров его группы его место свободно, то он на это место помещается. Иначе создаётся новый контейнер, куда и помещается параллелипипед на свое место.

Теорема 8 (Chang, Wang, Kankanhalli, 1993) [32]. Для предложенного алгоритма $W_A^n = \theta\left(\frac{n}{m}\right) + \theta(\sqrt{nm^d})$.

Первое слагаемое учитывает объём свободного пространства в контейнерах, заполненных всеми возможными (2^d) параллелипипедами, а второе – число не полностью заполненных контейнеров. Выбирая $m = n^{\frac{1}{d+2}}$, получаем $W_A^n = O(n^{\frac{d+1}{d+2}})$.

Алгоритм удовлетворяет условию open-end по Шору, для получения более сильного условия open-end с помощью алгоритма нужно упаковывать последовательно 2, $2^2, \dots,\ 2^k, \dots$ параллелипипедов, пока не получим символ останова.

5. Задача Strip Packing упаковки прямоугольников в полосу

Естественным обобщением задачи BP является задача Strip Packing упаковки прямоугольников в полубесконечную полосу. Задача впервые была исследована в [33], а в [34] Бэкером и Шварцем для применения алгоритмов из BP для задачи SP был предложен класс шельфовых алгоритмов, где шельф – такая часть полосы единичной ширины и ограниченной высоты, что прямоугольник не пересекается с границей шельфа, а лежит либо внутри, либо снаружи шельфа, и любая вертикальная прямая, пересекающая шельф, пересекает не более одного прямоугольника, лежащего внутри шельфа.

Алгоритм 3: Shelf(A,r). Пусть высота прямоугольника R, $h(R) \in (r^{k+1}, r^k]$, $r < 1$. Прямоугольник размещается в одном из шельфов высоты r^k. Для размещения прямоугольника в шельфах высоты r^k используем некоторую эвристику A для задачи BP в предположении, что множество шельфов данной высоты – это контейнеры, а ширина прямоугольника – вес упаковываемого объекта. При надобности алгоритм A создает новый шельф высоты r^k наверху текущей упаковки.

5.1 Анализ в худшем случае

Задача Strip Packing в случае, когда высоты всех прямоугольников одинаковы, эквивалентна задаче BP, стало быть, для онлайновой постановки задачи SP верна нижняя оценка из [19]: $R_A^\infty > 1.540$.

В [35] были предложены шельфрвые алгоритмы с асимптотичесской точностью R^∞, сколь угодно близкой к $\Pi_\infty (\Pi_\infty \approx 1.691)$, а также было показано, что для любого онлайнового шельфового алгоритма $R_A^\infty \geq \Pi_\infty$.

В [36] Ханом, Ивамом, Йе и Жангом был предложен алгоритм для задачи SP, имеющий асимптотическую точность $R_A^\infty \leq 1.588\ldots$ Для задач BP и SP неизвестен алгоритм с лучшей асимптотической точностью. Алгоритм разделяет прямоугольники на группы по ширине. Прямоугольники с шириной $w \leq \varepsilon_0$, упаковываются шельфовым алгоритмом $Shelf(Harmonic++, r)$. Пусть $\varepsilon_0 < t_1 <..< t_k = 1$, $c > h_{max}$, где h_{max}- максимальная высота прямоугольника. Прямоугольники, ширина которых $w \in (t_i, t_{i+1}]$, упаковывались в полоски высоты c и ширины t_{i+1}. Полоски одинаковой высоты, так как в данном случае задача SP эквивалентна задаче BP, размещались аналогом алгоритма Harmonic++ из [17] для BP.

5.2 Анализ в среднем

Задача исследовалась в предположении, что длины и высоты сторон-независимые в совокупности случайные велчины, равномерно распределённые на отрезке [0,1].

В 1993 году задача Strip Packing была исследована в среднем случае в [29] Коффманом и Шором. Было показано, что для любого шельфового алгоритма $W^n_{Shelf(A,r)} = \Omega(N^{\frac{2}{3}})$. Позднее в [38] были получены оценки точности онлайных шельфовых алгоритмов (в closed-end случае при наперед заданных значениях r), использующих алгоритмы First Fit и Best Fit для размещения прямоугольников внутри шельфов:

$$W^n_{Shelf(FF,r_{FF})} = O\left(N^{\frac{3}{4}}\right), W^n_{Shelf(BF,r_{BF})} = O\left(N^{\frac{2}{3}} \log^{\frac{1}{2}} N\right)$$

В работе [37] был предложен closed-end алгоритм с точностью $O\left(N^{\frac{2}{3}}\right)$, а 2011 году в [39] Кузюриным и Поспеловым был предложен open-end алгоритм с точностью $\theta\left(N^{\frac{2}{3}}\right)$. В работе [40] Трушниковым был предложен новый closed-end алгоритм, точность которого была исследована в работе [41], где было показано, что $W^n = O(N^{\frac{1}{2}} \log^{\frac{3}{2}} N)$. В [42] оценка была улучшена до

$$W^n = O\left(N^{\frac{1}{2}} \log N\right).$$

Значительное улучшение точности по сравнению с ранее известными алгоритмами обусловлено ограничением общей высоты разбиения на

контейнеры и усовершенствованным способом упаковки прямоугольники в контейнеры.

Алгоритм 4 (Трушников, 2012) [40]**.**

Пусть N – заранее известное число прямоугольников. Обозначим $d = \left\lfloor \frac{\sqrt{N}}{4} \right\rfloor, U = \frac{N}{4d}$. В основании полосы по левому краю выделяются прямоугольные области (контейнеры) высоты U, ширина i-ого контейнера равна $\frac{i}{d}$. Эти области образуют пирамиду. Рассмотрим вторую пирамиду, центрально симметричную данной относительно центра области, которая начинается от основания пирамиды, имеет высоту $(d+1)U$, а ширина равна ширине полосы (см. рис. 3)

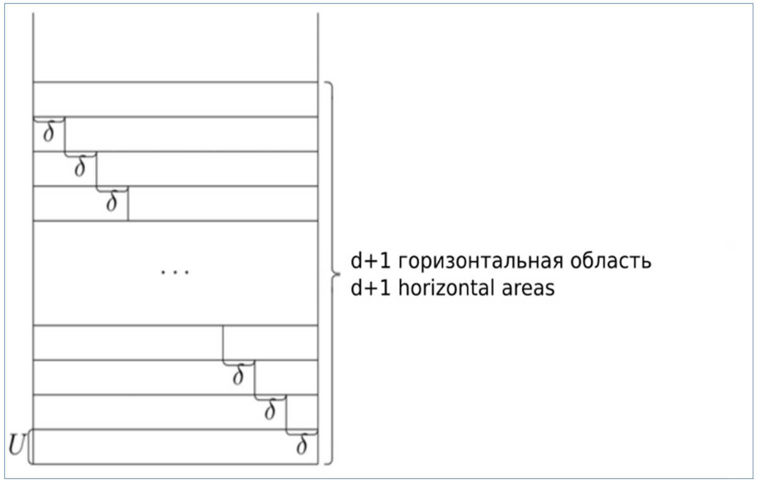

Рис. 3. Разбиение полосы на контейнеры
Fig. 3. Division of the strip into containers

Каждый четный прямоугольник размещается в одну пирамиду, а каждый нечетный – в другую. При упаковке каждого следующего прямоугольника шириной w:

- находим i: $\frac{i-1}{d} < w \leq \frac{i}{d}$;
- ищем $\min j : i \leq j \leq d$ такое, что прямоугольник помещается в контейнер шириной $\frac{j}{d}$;
- если такое j существует, кладём прямоугольник на верхнюю грань верхнего прямоугольника в контейнере шириной $\frac{j}{d}$;
- иначе называем прямоугольник выпавшим и кладём его наверх текущей упаковки.

Принципиальным отличием данного алгоритма от его предшественников является то, что число контейнеров $(2d)$ – зависит от числа прямоугольников, но не зависит от самих прямоугольников.

Для задачи Strip Packing известна лишь очевидная нижняя оценка, верная как для online, так и для offline алгоритмов: $W^n = \Omega(N^{\frac{1}{2}})$.

6. Задача Multiple Strip Packing упаковки прямоугольников в несколько полос единичной ширины

Для практических приложений, например, для задач оптимизации в распределённых вычислительных системах [43] полезно, перейти от рассмотрения задачи об упаковке прямоугольников в одну полосу к рассмотрению задачи об упаковке прямоугольников в несколько полос (MSP). В данной задаче продуктивна идея об упаковке некоторого объекта в наименее заполненную полосу, где объектами могут быть шельфы для шельфовых алгоритмов или выпавшие прямоугольники для алгоритмов из [40].

6.1 Анализ в худшем случае

Рассмотрим шельфовый алгоритм упаковки прямоугольников в несколько полос, предложенный в работе [44].

Алгоритм 5 (Йе, Хан, Жанг, 2011) [44]:

- выбираем шельф, в который кладётся прямоугольник согласно эвристике $Shelf(A, r)$;
- если такого шельфа не существует, создаем новый шельф наверху текущей упаковки в полосе, заполненной не выше любой другой полосы.

Аналогично [35], можно получить семейство шельфовых алгоритмов с асимптотической точностью R^∞, сколь угодно близкой к $\Pi_\infty (\Pi_\infty \approx 1.691)$.

Размещая же каждый контейнер согласно алгоритму из работы [36] в наименее заполненную по высоте полосу, можно получить для случая MSP оценку из [36]: $R_A^\infty \leq 1.588$...

6.2 А*нализ в среднем*

Алгоритм 4 может быть обобщён на случай нескольких полос упаковкой выпавших прямоугольников в наименее высоко заполненный контейнер.

В работе [42] алгоритм был обобщен на случай, когда число полос для упаковки зависит от числа прямоугольников. Был предложен алгоритм упаковки N прямоугольников в k полос, $k \leq \sqrt{N}$. Для него была доказана следующая теорема:

Теорема 9 (Лазарев, Кузюрин, 2017) [42]. При упаковке N прямоугольников в k полос, $k \leq \sqrt{N}$, алгоритм обеспечивает площадь незаполненного пространства $W^n = O(N^{\frac{1}{2}} \log N)$.

Для задачи Multiple Strip Packing, как и для задачи SP, известна лишь очевидная нижняя оценка, верная как для online, так и для offline алгоритмов: $W^n = \Omega(N^{\frac{1}{2}})$.

7. Задача об упаковке прямоугольников в полосы различной ширины. Анализ в худшем случае

Обобщением задачи MSP является задача об упаковке прямоугольников в полосы различной ширины.

Дан набор полубесконечных полос $C = \{C_1, \dots, C_m\}$, w_i – ширина i-ой полосы. В эти полосы требуется упаковать без вращений и пересечений n открытых прямоугольников $T = \{T_1, \dots, T_n\}$, минимизируя при этом высоту упаковки.

Отличие от задачи MSP заключается лишь в том, что ширина у полос различная.

В [45, 46] задача впервые была рассмотрена Жуком, и был предложен алгоритм, распределяющий прямоугольники в полосы в режиме онлайн, а в полосах использующий оффлайновую эвристику для упаковки с асимптотической точностью $R_A^\infty < 10$. В [47] для $\forall r \in (0,1)$ был предложен полностью онлайновый алгоритм A_r с асимптотической точностью $R_{A(r)}^\infty \leq \frac{8}{r}$. В [48] Жук доказал, что для любого онлайнового алгоритма $R_A^\infty \geq e$, $e = 2{,}7182\dots$. В работе [49] был предложен онлайновый алгоритм с асимптотической точностью $R_A^\infty < \frac{2e}{r}, \forall r \in (0,1)$.

Предположим, что $w_1 \geq \dots \geq w_m$. Пусть $R -$ прямоугольник с шириной $w(R)$. Скажем, что $last(R) = max\{k: w_k \geq w(R)\}$. Разобьем все выпавшие прямоугольники $\{T\}$ на m множеств $M_1, \dots, M_m: R \in M_i \Leftrightarrow last(R) = i$.

Если Q – множество прямоугольников, то за $S(Q)$ обозначим суммарную площадь прямоугольников из множества Q.

Определим $h(T) = \sup_k \frac{\sum_{i=1}^k S(M_i)}{\sum_{i=1}^k w_i}$. Нетрудно показать, что высота оптимального распределения $H_o \geq h$.

За $y_i(T)$ обозначим суммарную площадь прямоугольников, попавших в i-ую полосу после выпадения множества T прямоугольников.

Алгоритм 6: A_r (Жук, 2012) [49].

При добавлении очередного прямоугольника R:

- вычисляем $h = h(T + \{R\})$;
- определяем номер полосы k: $k = \max\left\{i: w(R) \leq w_i, \frac{y_i}{w_i} \leq eh\right\}$ (было доказано, что такая полоса всегда существует);

- для размещения внутри полосы используется шельфовый алгоритм $Shelf(r), r < 1$, разбивающий полосу на слои высотой $r^z, z \in \mathbb{Z}$; алгоритм кладёт прямоугольник R высотой $h(R)$ в слой высотой r^x ($r^{x-1} < h(R) \leq r^x$), а для упаковки в слой используется эвристика First Fit.

Для получения асимптотической точности R_A^∞ используется следующая лемма.

Лемма. При упаковке прямоугольников высотой не больше h_{max} суммарной площадью S алгоритмом $Shelf(r)$ в полосу ширины w высота упаковки

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

Для доказазательства леммы нужно рассмотреть слои, заполненные менее чем на $\frac{w}{2}$ по ширине, слои, заполненные более чем наполовину по ширине, и последний слой.

Из леммы получаем, что для алгоритма A_r мультипликативная асимптотическая точность $R_A^\infty \leq 2e$.

8. Заключение

Классическая задача Bin Packing очень хорошо изучена: при анализе в худшем случае показано, что для любого онлайнового алгоритма $R^\infty \geq 1.540$, и был предложен алгоритм с $R^\infty \leq 1.589$. При анализе в среднем как в случае open-end, так и в случае closed-end были предложены алгоритмы, у которых математическое ожидание площади незаполненного пространства контейнеров W^n имеет неулучшаемую асимптотику.

Однако многие обобщения этой задачи активно изучаются в настоящее время, и многие результаты еще предстоит получить. В частности, в многомерном обобщении Multidimensional Bin Packing задачи Bin Packing при анализе в среднем неизвестны нижние оценки, кроме очевидных $W^n = \Omega(\sqrt{n})$.

В задачах Strip Packing и Multiple Strip Packing при анализе в худшем случае был предложен алгоритм с $R^\infty \leq 1.589$, а при анализе в среднем в closed-end случае, был предложен алгоритм с $W^n = O(\sqrt{n}\ln n)$, однако при анализе в постановке open-end неизвестна нижняя оценка W^n, лучшая $W^n = \Omega\left(\sqrt{n}\right)$, и неизвестен алгоритм с $W^n = o(n^{2/3})$.

Задача упаковки прямоугольником в полосы различной ширины активно исследовалась в худшем случае, однако в среднем задача еще не была проанализирована.

Список литературы

[1]. Массобрио Р., Несмачнов С., Черных А., Аветисян А., Радченко Г. Применение облачных вычислений для анализа данных большого объёма в умных городах. Труды ИСП РАН, том 28, вып. 6, 2016 г., стр. 121-140, DOI: 10.15514/ISPRAS-2016-28(6)-9

[2]. Аничкин А.С., Семенов В.А. Математическая формализация задач проектного планирования в расширенной постановке. Труды ИСП РАН,том 29, вып. 2, 2017 г., стр. 231-256. DOI: 10.15514/ISPRAS-2017-29(2)-9

[3]. Зеленова С.А., Зеленов С.В. Критерий существования бесконфликтного расписания для системы строго периодических задач. Труды ИСП РАН, том 29, вып. 6, 2017 г., стр. 183-202. DOI: 10.15514/ISPRAS-2017-29(6)-10

[4]. Ghalam L., Grosu D. A Parallel Approximation Algorithm for Scheduling Identical Machines. In IEEE International Parallel and Distributed Processing Symposium Workshops, 2017, pp. 619-628

[5]. Sheikhalishahi M., Wallace R. M., Grandinetti L., Vazquez-Poletti J. L., Guerriero F. A multi-dimensional job scheduling. Future Generation Computer Systems, vol. 54, 2016, pp. 123-131

[6]. Tchernykh A., Schwiegelshohn U., Yahyapour R., Kuzjurin N. On-line hierarchical job scheduling on grids with admissible allocation. Journal of Scheduling, 2010, vol. 13, issue 5, pp. 545-552

[7]. Tshernykh A., Ramirez J.M., Avetisyan A., Kuzjurin N., Grushin D., Zhuk S. Two-Level Job-Scheduling strategies for a Computational Grid. Lecture Notes in Computer Science book series, vol. 3911, pp. 774-781

[8]. Cohil B., Shah S., Goleshha Y., Patel D. A Comparative Analysis of Virtual Machine Placement Techniques in the Cloud Environment. International Journal of Computer Applications, vol. 156, no. 14, 2016, pp. 12-18

[9]. Garey M.R., Johnson D.S. Computers and intractability: a guide to the theory of NP-completeness. Freeman: San Francisco, 1979, 338 p.

[10]. Johnson D.S. Near-optimal Bin Packing Algorithms. PhD Thesis, Massachusetts Institute of Technology, Department of Mathematics, Cambridge, 1973. 401 p.

[11]. Johnson D.S., Demers A., Ullman J.D., Garey M.R., Graham R.L. Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms. SIAM Journal on computing, vol. 3, issue 4, 1974, pp. 299- 325

[12]. Garey M.R. Graham R.L., Ullman J.D., Worst-case analysis of memory allocation algorithms. Proceedings of the fourth annual ACM symposium on theory of computing. 1972, pp. 143-150

[13]. Garey M.R., Graham R.L., Johnson D.S., Yao A.C. Resource constrained scheduling as generalized bin packing. Journal of Combinatorial Theory, Series A, vol. 21, issue 3, 1976, pp. 257-298

[14]. Yao A.C. New Algorithms for Bin Packing. Journal of the ACM, vol. 27, issue 2, 1981, pp. 207-227

[15]. Gambosi G., Postiglione a., Talamo M.M. New algorithms for online Bin Packing. In Proceedings of the First Italian Conference on Algorithms and Complexity, 1990, pp. 44-59

[16]. Ivcović Z. and Lloyd E. Fully dynamic algorithms for Bin Packing: Being (mostly) myopic helps. Lecture Notes in Computer Science, vol. 726, pp. 224-235

[17]. Seiden S.S. On the Online Bin Packing Problem. Lecture Notes in computer science, vol. 2076, 2002, pp. 207-227

[18]. Brown J.D. A lower Bound for On-Line One Dimensional Bin Packing Algorithms. Technical Report R-864, coordinated Science laboratory, University of Illinois, Urbana, IL, 1979.

[19]. Vliet A. An improved lower bound for on-line bin packing algorithms. Information Processing Letters, vol. 43, issue 5, 1992, pp. 277-284

[20]. Breitgand D., Epstein A. Improving consolidations of virtual machines with risk-aware Bandwidth oversubscription in compute clouds. In Proceedings of the IEEE INFCOM, 2012, pp. 2861-2865
[21]. Ajtai M., Komlós J., Tusnadi G. On Optimal Matchings. Combinatorica, vol. 4, issue 4, 1984, pp. 259-264
[22]. Karp R.M., Luby M., Marchetti-Spaccamela A. A probabilistic analysis of multidimensional bin packing problem. In Proceedings of the sixteen annual ACM symposium on theory of computing, 1984, pp.289-298
[23]. Coffman E.G., Shor P.W. A Simple Proof of the sqrt(n log3/4 n) Upright Matching Bound. SIAM Journal on Discrete Mathematics, vol. 4, issue 1, 1991, pp. 48-57
[24]. Shor P.W. The average-case analysis of some on-line algorithms for bin packing. Combinatorica, vol. 6, issue 4, 1986, pp. 179-200
[25]. Leighton F.T., Shor P. Tight bonds for minimax grid matching, with application to the average-case analysis of algorithms. In Proceedings of the eighteenth Annual ACM symposium on theory of computing, 1986 , pp. 91-103
[26]. Coffman E.G., Courcoubetis C., Garey M.R., Johnson D.S., McGeoch L.A., Shor P.W., Weber R. and Yannakakis M. Fundamental discrepancies between average-case analysis under discrete and continuous distributions: a bin packing study. In Proceedings of the Twenty-first Annual ACM symposium on theory of computing, 1991, pp. 230-240
[27]. Shor P.W. How to pack better than Best Fit: tight bounds for average-case online Bin Packing. In Proceedings 32nd of the Annual Symposium of foundations of Computer Science, 1991, pp. 752-759
[28]. Galambos G., A. van Vliet. Lower bounds for 1-, 2-, and 3- dimensional on-line bin packing algorithms. Computing, vol. 52, issue 3, 1994, pp. 281-297
[29]. Han X., Chin F.Y.L., Ting H.-F., Zhang G., Zhang Y. A new upper bound 2.5545 on 2D Online Bin Packing. ACM Transactions on algorithms, vol. 7, issue 4, 2011, article No. 50
[30]. Csirik J., A. van Vliet. An on-line algorithm for multidimensional bin packing. Operation Research Letters, vol. 13, issue 3, 1993, pp. 149-158
[31]. Epstein l., R. van Stee. Optimal Online Algorithms for Multidimensional Packing Problems. In Proceedings of the Fifteenth Annual ACM-CIAM Symposium on Discrete algorithms, 2004, pp. 214-223
[32]. Chang E-C, Wang W., Kankanhalli M.S. Multidimensional on-line bin-packing: An algorithm and it's average-case analysis. Information Processing Letters, vol. 48, issue 3, 1993, pp. 121-125
[33]. Baker B.S., Coffman E.G., Rivest R.L. Orthogonal Packings in Two Dimensions. SIAM Journal on Computing, vol. 9, issue 4, 1980, pp. 846-855
[34]. Baker B.S., Schwarz J.S. Shelf algorithms for two-dimensional packing problems. SIAM Journal on Computing, vol. 12, issue 3, 1983, pp. 508-525
[35]. Csirik J., Woeginger G.J. Shelf algorithms for on-line strip packing. Information Processing Letters, vol. 63, issue 4, 1997, pp. 171-175
[36]. Han X., Iwama K., Ye d., Zhang G. Strip Packing vs Bin Packing. Lecture Notes in Computer Science, vol. 4508, 2007, pp. 358-367
[37]. Coffman E.G., Shor P.W. Packing in two dimensions: Asymptotic average-case analysis of algorithms. Algorithmica, vol. 9, issue 3, 1993, pp. 253-277
[38]. Кузюрин Н.Н., Поспелов А.И. Вероятностный анализ различых шельфовых алгоритмов упаковки прямоугольников в полосу. Труды ИСП РАН, том 12, 2007 г., стр. 17-26

[39]. Кузюрин Н.Н., Поспелов А.И. Вероятностный анализ нового класса алгоритмов упаковки прямоугольников в полосу. Ж. вычисл. матем. и матем. физ., том 51, no. 10, 2011, стр. 1931-1936
[40]. Трушников М.А. Об одной задаче Коффмана-Шора, связанной с упаковкой прямоугольников в полосу. Труды ИСП РАН, том 22, 2012 г., стр. 456-462, DOI: 10.15514/ISPRAS-2012-22-24
[41]. Трушников М.А. Вероятностный анализ нового алгоритма упаковки прямоугольников в полосу. Труды ИСП РАН, том 24, 2013 г., стр. 457-468, DOI: 10.15514/ISPRAS-2013-24-21
[42]. Лазарев Д.О., Кузюрин Н.Н. Алгоритм упаковки прямоугольников в несколько полос и анализ его точности в среднем. Труды ИСП РАН, том 29, вып. 6, 2017 г., стр. 221-228, DOI: 10.15514/ISPRAS-2017-29(6)-13
[43]. Кузюрин Н.Н., Грушин Д.А., Фомин А. Проблемы двумерной упаковки и задачи оптимизации в распределённых вычислительных системах. Труды ИСП РАН, том 26, вып. 1, 2014 г., стр. 483-502, DOI: 10.15514/ISPRAS-2014-26(1)-21
[44]. Ye D., Han X., Zhang G. Online multiple-strip packing. Theoretical Computer Science, vol. 412, issue 3, 2011, pp. 233-239
[45]. Zhuk S.N. Approximation algorithms for packing rectangles into several strips. Discrete Mathematics and Applications, vol. 16, issue 1, 2006, pp. 73-85
[46]. Жук С.Н. Анализ некоторых эвристик упаковки прямоугольников в несколько полос. Труды ИСП РАН, том 6, 2004 г., стр. 13-26
[47]. Жук С.Н. Онлайновый алгоритм упаковки прямоугольников в несколько полос с гарантированными оценками точности. Труды ИСП РАН, том 12, 2007 г., стр. 7-16
[48]. Zhuk S.N. On-line algorithms for packing rectangles into several strips. Discrete Mathematics and Applications, vol. 17, issue 5, 2007, pp. 517-531
[49]. Жук С.Н. О построении расписаний выполнения параллельных задач на группах кластеров с различной производительностью. Труды ИСП РАН, том 23, 2012 г., стр. 447-454, DOI: 10.15514/ISPRAS-2012-23-27

On on-line algorithms for Bin, Strip and Box Packing, and their worst- and average-case analysis

[1,2] *D.O. Lazarev* <dennis810@mail.ru>
[1,2] *N.N. Kuzjurin* <nnkuz@ispras.ru>
[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*
[2] *Moscow Institute of Physics and Technology, Dolgoprudnyj, Institutskij alley, Moscow region, 141700, Russia*

Abstract. In this survey, online algorithms for such packing problems as Bin Packing, Strip Packing and their generalizations, such as Multidimensional Bin Packing, Multiple Strip Packing and packing into strips of different width were considered. For the latter problem only worst-case analysis was described, for all other problems, both worst-case and average case (probabilistic analysis) asymptotical ratios were considered. Both lower and upper bonds were described. Basic algorithms and methods for their analysis were considered. For worst-case analysis of the Bin Packing Problem it was shown that for any online algorithm $\boldsymbol{R}^{\infty} \geq \mathbf{1.540}$, and an algorithm with $\boldsymbol{R}^{\infty} \leq \mathbf{1.589}$ was obtained. Both approaches for analyzing the algorithm and obtaining the lower bonds were discussed. Also it was shown that First Fit algorithm for Bin Packing has asymptotical competitive ratio of $\frac{\mathbf{17}}{\mathbf{10}}$. For average case analysis in the case when object's sizes have a uniform distribution on $[\mathbf{0},\mathbf{1}]$ in open-end analysis a construction for obtaining both lower bound of $\boldsymbol{W}^{\boldsymbol{n}} = \boldsymbol{\Omega}(\sqrt{\boldsymbol{n}\,\boldsymbol{ln}\,\boldsymbol{n}})$ and algorithm with $\boldsymbol{W}^{\boldsymbol{n}} = \boldsymbol{\theta}(\sqrt{\boldsymbol{n}\,\boldsymbol{ln}\,\boldsymbol{n}})$ was shown. In the case of closed-end analysis an algorithm with $\boldsymbol{W}^{\boldsymbol{n}} = \boldsymbol{\theta}(\sqrt{\boldsymbol{n}})$ was described. For Multidimensional Bin Packing with $\boldsymbol{d}$ dimensions an algorithm with $\boldsymbol{R}^{\infty} = (\Pi_{\infty})^{\boldsymbol{d}}$, where $\Pi_{\infty} \approx \mathbf{1.691}$, was obtained. For average case analysis an algorithm with $\boldsymbol{W}_{\boldsymbol{A}}^{\boldsymbol{n}} = \boldsymbol{O}(\boldsymbol{n}^{\frac{\boldsymbol{d}+\mathbf{1}}{\boldsymbol{d}+\mathbf{2}}})$ was shown. For worst-case analysis of Strip Packing Problem and Multiple Strip Packing Problem algorithms with $\boldsymbol{R}^{\infty} \leq \mathbf{1.589}$ were also obtained. For the closed-end average case analysis algorithms with $\boldsymbol{W}^{\boldsymbol{n}} = \boldsymbol{\theta}(\sqrt{\boldsymbol{n}}\,\boldsymbol{ln}\,\boldsymbol{n})$ were described. For the open-end average case analysis of Strip Packing Problem an algorithm with $\boldsymbol{W}^{\boldsymbol{n}} = \boldsymbol{O}\left(\boldsymbol{n}^{\frac{\mathbf{2}}{\mathbf{3}}}\right)$ was considered. For generalization of Multiple Strip Packing Problem, where strips have different widths, an online algorithm with $\boldsymbol{R}^{\infty} \leq \frac{\mathbf{2}\boldsymbol{e}}{\boldsymbol{r}}$ for any $\boldsymbol{r} < \mathbf{1}$, where $\boldsymbol{e} = \mathbf{2.718}$ ..., was described.

Keywords: Bin Packing; Multidimensional Bin Packing; Strip Packing; Multiple Strip Packing; Packing in Strips of different width; probabilistic analysis; worst-case analysis.

DOI: 10.15514/ISPRAS-2018-30(4)-14

For citation: Lazarev D.O., Kuzjurin N.N. On on-line algorithms for Bin, Strip and Box Packing, and their worst- and average-case analysis. Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue 4, 2018. pp. 209-230 (in Russian). DOI: 10.15514/ISPRAS-2018-30(4)-14

References

[1]. Massobrio R., Nesmachnow S., Tchernykh A., Avetisyan A., Radchenko G. Towards a Cloud Computing Paradigm for Big Data Analysis in Smart Cities. Trudy ISP RAN/Proc.ISP RAS, vol. 28, issue 6, 2016. pp. 121-140 (in Russian). DOI: 10.15514/ISPRAS-2016-28(6)-9
[2]. Anichkin A.S., Semenov V.A. Mathematical formalization of project scheduling problems. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 2, 2017. pp. 231-256 (in Russian). DOI: 10.15514/ISPRAS-2017-29(2)-9
[3]. Zelenova S.A., Zelenov S.V. Non-conflict scheduling criterion for strict periodic tasks. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 6, 2017. pp. 183-202 (in Russian). DOI: 10.15514/ISPRAS-2017-29(6)-10
[4]. Ghalam L., Grosu D. A Parallel Approximation Algorithm for Scheduling Identical Machines. In IEEE International Parallel and Distributed Processing Symposium Workshops, 2017, pp. 619-628
[5]. Sheikhalishahi M., Wallace R. M., Grandinetti L., Vazquez-Poletti J. L., Guerriero F. A multi-dimensional job scheduling. Future Generation Computer Systems, vol. 54, 2016, pp. 123-131
[6]. Tchernykh A., Schwiegelshohn U., Yahyapour R., Kuzjurin N. On-line hierarchical job scheduling on grids with admissible allocation. Journal of Scheduling, 2010, vol. 13, issue 5, pp. 545-552
[7]. Tshernykh A., Ramirez J.M., Avetisyan A., Kuzjurin N., Grushin D., Zhuk S. Two-Level Job-Scheduling strategies for a Computational Grid. Lecture Notes in Computer Science book series, vol. 3911, pp. 774-781
[8]. Cohil B., Shah S., Goleshha Y., Patel D. A Comparative Analysis of Virtual Machine Placement Techniques in the Cloud Environment. International Journal of Computer Applications, vol. 156, no. 14, 2016, pp. 12-18
[9]. Garey M.R., Johnson D.S. Computers and intractability: a guide to the theory of NP-completeness. Freeman: San Francisco, 1979, 338 p.
[10]. Johnson D.S. Near-optimal Bin Packing Algorithms. PhD Thesis, Massachusetts Institute of Technology, Department of Mathematics, Cambridge, 1973. 401 p.
[11]. Johnson D.S., Demers A., Ullman J.D., Garey M.R., Graham R.L. Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms. SIAM Journal on computing, vol. 3, issue 4, 1974, pp. 299- 325
[12]. Garey M.R. Graham R.L., Ullman J.D., Worst-case analysis of memory allocation algorithms. Proceedings of the fourth annual ACM symposium on theory of computing. 1972, pp. 143-150
[13]. Garey M.R., Graham R.L., Johnson D.S., Yao A.C. Resource constrained scheduling as generalized bin packing. Journal of Combinatorial Theory, Series A, vol. 21, issue 3, 1976, pp. 257-298
[14]. Yao A.C. New Algorithms for Bin Packing. Journal of the ACM, vol. 27, issue 2, 1981, pp. 207-227
[15]. Gambosi G., Postiglione a., Talamo M.M. New algorithms for online Bin Packing. In Proceedings of the First Italian Conference on Algorithms and Complexity, 1990, pp. 44-59
[16]. Ivcović Z. and Lloyd E. Fully dynamic algorithms for Bin Packing: Being (mostly) myopic helps. Lecture Notes in Computer Science, vol. 726, pp. 224-235
[17]. Seiden S.S. On the Online Bin Packing Problem. Lecture Notes in computer science, vol. 2076, 2002, pp. 207-227

[18]. Brown J.D. A lower Bound for On-Line One Dimensional Bin Packing Algorithms. Technical Report R-864, coordinated Science laboratory, University of Illinois, Urbana, IL, 1979.
[19]. Vliet A. An improved lower bound for on-line bin packing algorithms. Information Processing Letters, vol. 43, issue 5, 1992, pp. 277-284
[20]. Breitgand D., Epstein A. Improving consolidations of virtual machines with risk-aware Bandwidth oversubscription in compute clouds. In Proceedings of the IEEE INFCOM, 2012, pp. 2861-2865
[21]. Ajtai M., Komlós J., Tusnadi G. On Optimal Matchings. Combinatorica, vol. 4, issue 4, 1984, pp. 259-264
[22]. Karp R.M., Luby M., Marchetti-Spaccamela A. A probabilistic analysis of multidimensional bin packing problem. In Proceedings of the sixteen annual ACM symposium on theory of computing, 1984, pp.289-298
[23]. Coffman E.G., Shor P.W. A Simple Proof of the sqrt(n log3/4 n) Upright Matching Bound. SIAM Journal on Discrete Mathematics, vol. 4, issue 1, 1991, pp. 48-57
[24]. Shor P.W. The average-case analysis of some on-line algorithms for bin packing. Combinatorica, vol. 6, issue 4, 1986, pp. 179-200
[25]. Leighton F.T., Shor P. Tight bonds for minimax grid matching, with application to the average-case analysis of algorithms. In Proceedings of the eighteenth Annual ACM symposium on theory of computing, 1986 , pp. 91-103
[26]. Coffman E.G., Courcoubetis C., Garey M.R., Johnson D.S., McGeoch L.A., Shor P.W., Weber R. and Yannakakis M. Fundamental discrepancies between average-case analysis under discrete and continuous distributions: a bin packing study. In Proceedings of the Twenty-first Annual ACM symposium on theory of computing, 1991, pp. 230-240
[27]. Shor P.W. How to pack better than Best Fit: tight bounds for average-case online Bin Packing. In Proceedings 32nd of the Annual Symposium of foundations of Computer Science, 1991, pp. 752-759
[28]. Galambos G., A. van Vliet. Lower bounds for 1-, 2-, and 3- dimensional on-line bin packing algorithms. Computing, vol. 52, issue 3, 1994, pp. 281-297
[29]. Han X., Chin F.Y.L., Ting H.-F., Zhang G., Zhang Y. A new upper bound 2.5545 on 2D Online Bin Packing. ACM Transactions on algorithms, vol. 7, issue 4, 2011, article No. 50
[30]. Csirik J., A. van Vliet. An on-line algorithm for multidimensional bin packing. Operation Research Letters, vol. 13, issue 3, 1993, pp. 149-158
[31]. Epstein l., R. van Stee. Optimal Online Algorithms for Multidimensional Packing Problems. In Proceedings of the Fifteenth Annual ACM-CIAM Symposium on Discrete algorithms, 2004, pp. 214-223
[32]. Chang E-C, Wang W., Kankanhalli M.S. Multidimensional on-line bin-packing: An algorithm and it's average-case analysis. Information Processing Letters, vol. 48, issue 3, 1993, pp. 121-125
[33]. Baker B.S., Coffman E.G., Rivest R.L. Orthogonal Packings in Two Dimensions. SIAM Journal on Computing, vol. 9, issue 4, 1980, pp. 846-855
[34]. Baker B.S., Schwarz J.S. Shelf algorithms for two-dimensional packing problems. SIAM Journal on Computing, vol. 12, issue 3, 1983, pp. 508-525
[35]. Csirik J., Woeginger G.J. Shelf algorithms for on-line strip packing. Information Processing Letters, vol. 63, issue 4, 1997, pp. 171-175
[36]. Han X., Iwama K., Ye d., Zhang G. Strip Packing vs Bin Packing. Lecture Notes in Computer Science, vol. 4508, 2007, pp. 358-367

[37]. Coffman E.G., Shor P.W. Packing in two dimensions: Asymptotic average-case analysis of algorithms. Algorithmica, vol. 9, issue 3, 1993, pp. 253-277
[38]. Kuzjurin N.N., Pospelov A.I. Probabilistic analysis of different shelf algorithms for packing rectangles into a strip. Trudy ISP RAN/Proc. ISP RAS, vol. 12, 2007, pp. 17-26 (in Russian)
[39]. Kuzyurin N.N., Pospelov A.I. Probabilistic analysis of a new class of strip packing algorithms. Comput. Math. and Math. Phys., vol. 51, issue 10, 2011, article no. 1817
[40]. Trushnikov M.A. On one problem of Koffman-Shor connected to strip packing problem. Trudy ISP RAN/Proc. ISP RAS, vol. 22, 2012, pp. 456-462 (in Russian). DOI: 10.15514/ISPRAS-2012-22-24
[41]. Trushnikov M.A. Probabilistic analysis of a new strip packing algorithm. Trudy ISP RAN/Proc. ISP RAS, vol. 24, 2013, str. 457-468 (in Russian). DOI: 10.15514/ISPRAS-2013-24-21
[42]. Lazarev D.O., Kuzyrin N.N. An algorithm for Multiple Strip Package and its average case evaluation. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 6, 2017. pp. 221-228 (in Russian). DOI: 10.15514/ISPRAS-2017-29(6)-13
[43]. Kuzjurin N.N., Grushin D.A., Fomin A. Two-dimensional packing problems and optimization in distributed computing systems. Trudy ISP RAN/Proc. ISP RAS, vol. 26, issue 1, 2014, pp. 483-502 (in Russian). DOI: 10.15514/ISPRAS-2014-26(1)-21
[44]. Ye D., Han X., Zhang G. Online multiple-strip packing. Theoretical Computer Science, vol. 412, issue 3, 2011, pp. 233-239
[45]. Zhuk S.N. Approximation algorithms for packing rectangles into several strips. Discrete Mathematics and Applications, vol. 16, issue 1, 2006, pp. 73-85
[46]. Zhuk S.N. Analysis of some heuristics of packing rectangles into several strips. Trudy ISP RAN/Proc. ISP RAS, vol. 6, 2004, pp. 13-26 (in Russian)
[47]. Zhuk S.N. Online algorithm for packing rectangles into several strips with guaranteed accuracy estimates. Trudy ISP RAN/Proc. ISP RAS, vol. 12, 2007, pp. 7-16 (in Russian)
[48]. Zhuk S.N. On-line algorithms for packing rectangles into several strips. Discrete Mathematics and Applications, vol. 17, issue 5, 2007, pp. 517-531
[49]. Zhuk S.N. On-line algorithm for scheduling parallel tasks on a group of related clusters. Trudy ISP RAN/Proc. ISP RAS, vol. 23, 2012, pp. 447-454 (in Russian). DOI: 10.15514/ISPRAS-2012-23-27